Legal Knowledge and Information Systems

JURIX 2022: The Thirty-fifth Annual Conference, Saarbrücken, Germany, 14-16 December 2022

Editors:
Enrico Francesconi, Georg Borges and Christoph Sorge

JURIX 2022

Legal Knowledge and Information Systems

JURIX 2022:
The Thirty-fifth Annual Conference

Editors:
Enrico Francesconi, Georg Borges and Christoph Sorge

In recent years, interest within the research community and the legal industry regarding technological advances in legal knowledge representation and processing has been growing. This relates to areas such as computational models of legal reasoning, cybersecurity, privacy, trust and blockchain methods, among other things.

This book presents the proceedings of JURIX 2022, the 35th International Conference on Legal Knowledge and Information Systems, held from 14 –16 December in Saarbrücken, Germany, under the auspices of the Dutch Foundation for Legal Knowledge Based Systems and hosted by Saarland University. The annual JURIX conference has become an international forum for academics and professionals to exchange knowledge and experiences at the intersection of law and artificial intelligence (AI). For this edition, 62 submissions were received from 163 authors in 24 countries. Following a rigorous review process, carried out by a programme committee of 72 experts recognised in the field, 14 submissions were selected for publication as long papers, 22 as short papers and 5 as demo papers, making a total of 41 papers altogether and representing a 22.5% acceptance rate for long papers (66.1% overall). The broad array of topics covered includes argumentation and legal reasoning, legal ontologies and the semantic web, machine and deep learning and natural language processing for legal knowledge extraction, as well as argument mining, translation of legal texts, defeasible logic, legal compliance, explainable AI, alternative dispute resolution, legal drafting and smart contracts.

Providing an overview of recent advances, the book will be of interest to all those working at the interface between the law and AI.

JURIX 2022

ISBN 978-1-64368-364-5

ISBN 978-1-64368-364-5 (print)
ISBN 978-1-64368-365-2 (online)
ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

LEGAL KNOWLEDGE AND INFORMATION SYSTEMS

Frontiers in Artificial Intelligence and Applications

The book series Frontiers in Artificial Intelligence and Applications (FAIA) covers all aspects of theoretical and applied Artificial Intelligence research in the form of monographs, selected doctoral dissertations, handbooks and proceedings volumes. The FAIA series contains several sub-series, including 'Information Modelling and Knowledge Bases' and 'Knowledge-Based Intelligent Engineering Systems'. It also includes the biennial European Conference on Artificial Intelligence (ECAI) proceedings volumes, and other EurAI (European Association for Artificial Intelligence, formerly ECCAI) sponsored publications. The series has become a highly visible platform for the publication and dissemination of original research in this field. Volumes are selected for inclusion by an international editorial board of well-known scholars in the field of AI. All contributions to the volumes in the series have been peer reviewed.
The FAIA series is indexed in ACM Digital Library; DBLP; EI Compendex; Google Scholar; Scopus; Web of Science: Conference Proceedings Citation Index – Science (CPCI-S) and Book Citation Index – Science (BKCI-S); Zentralblatt MATH.

Volume 362

Recently published in this series

Vol. 361. A.M. Metelli, Exploiting Environment Configurability in Reinforcement Learning
Vol. 360. J.-L. Kim (Ed.), Machine Learning and Artificial Intelligence – Proceedings of MLIS 2022
Vol. 359. M. Hecher, Advanced Tools and Methods for Treewidth-Based Problem Solving
Vol. 358. A.J. Tallón-Ballesteros (Ed.), Fuzzy Systems and Data Mining VIII – Proceedings of FSDM 2022
Vol. 357. G. Šír, Deep Learning with Relational Logic Representations
Vol. 356. A. Cortés, F. Grimaldo and T. Flaminio (Eds.), Artificial Intelligence Research and Development – Proceedings of the 24th International Conference of the Catalan Association for Artificial Intelligence
Vol. 355. H. Fujita, Y. Watanobe and T. Azumi (Eds.), New Trends in Intelligent Software Methodologies, Tools and Techniques – Proceedings of the 21st International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques (SoMeT_22)

ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

Legal Knowledge and Information Systems

JURIX 2022: The Thirty-fifth Annual Conference, Saarbrücken, Germany, 14–16 December 2022

Edited by

Enrico Francesconi

Institute of Legal Informatics and Judicial Systems, National Research Council of Italy (IGSG-CNR)

Georg Borges

Saarland University, Saarbrücken, Germany

and

Christoph Sorge

Saarland University, Saarbrücken, Germany

IOS Press

Amsterdam • Berlin • Washington, DC

ISBN 978-1-64368-364-5 (print)
ISBN 978-1-64368-365-2 (online)
Library of Congress Control Number: 2022949887
doi: 10.3233/FAIA362

Publisher
IOS Press BV
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

For book sales in the USA and Canada:
IOS Press, Inc.
6751 Tepper Drive
Clifton, VA 20124
USA
Tel.: +1 703 830 6300
Fax: +1 703 830 2300
sales@iospress.com

Preface

We are pleased to present to you the proceedings of the 35th International Conference on Legal Knowledge and Information Systems – JURIX 2022. For more than three decades, the JURIX conferences have been held under the auspices of the Dutch Foundation for Legal Knowledge Based Systems (www.jurix.nl). Traditionally based in Europe, in the time JURIX has become an international forum for academics and professionals to exchange knowledge and experiences at the intersection of Law and Artificial Intelligence. Over the years, JURIX has witnessed the growing interest of the research community and the industry in technological advances on legal knowledge representation, computational models of legal reasoning, evidential reasoning, argumentation, case-based and rule-based reasoning, machine learning and natural language processing for legal knowledge acquisition, big data and data analysis, open data and the semantic web in the legal domain, online dispute resolution, legal document management and information retrieval, knowledge discovery, data mining, as well as cybersecurity, privacy, trust and blockchain methods.

The 2022 edition of JURIX, which runs from 14–16 December, is hosted by the Saarland University in Saarbrücken, Germany. This edition marks an important milestone, because it coincides with a relative return to normal after the challenges of the Covid-19 crisis: in fact it will be the first conference of the series of the AI&Law community events expected to be held fully in presence again.

For this edition, we have received 62 submissions from 163 authors of 24 countries. 14 of these submissions were selected for publication as long papers (10 pages), 22 as short papers (6 pages) and 5 as demo papers (4 pages) for a total of 41 presentations. This is the result of a balance between inclusiveness and a very competitive and rigorous review process, which was carried out by a Program Committee composed by 72 recognised experts in the field. The result was a total acceptance rate (long, short and demo papers) of 66.1%, which testifies the overall good quality of the submissions, with 22.5% acceptance rate for long papers.

The accepted papers cover a broad array of topics, from argumentation and legal reasoning, to legal ontologies and semantic web; from machine and deep learning to natural language processing for legal knowledge extraction, as well as argument mining and translation of legal texts; from defeasible logic, legal compliance and explainable AI, to alternative dispute resolution, legal drafting and smart contracts.

Three invited speakers, from different and complementary areas (industry, European institutions and academia), have honored JURIX 2022, by kindly accepting to deliver their keynote lectures: Martin Rollinger, Paul Nemitz and Mireille Hildebrandt.

Martin Rollinger is the CEO of Sinc, a leading company in Germany in the implementation of AI and legal informatics in the judiciary. In the past 20 years, he's been involved with numerous e-Government projects in domains such as education, law enforcement, intelligence and environmental management. He has been responsible for one of the largest projects in the digitization of German courts and public prosecutors offices and has been actively involved in building tools and products in the AI and law space for the past 10+ years.

Paul Nemitz is the Principal Advisor in the Directorate General for Justice and Consumers of the European Commission. He was appointed in April 2017, following a 6-year appointment as Director for Fundamental Rights and Citizen's Rights in the same Directorate General. As Director, Paul Nemitz led the reform of Data Protection legislation in the EU, the negotiations of the EU – US Privacy Shield and the negotiations with major US Internet Companies of the EU Code of Conduct against incitement to violence and hate speech on the Internet. He is a Member of Commission for Media and Internet Policy of the Social Democratic Party of Germany (SPD), Berlin and a visiting Professor of Law at the College of Europe in Bruges. Paul is also a Member of the Board of the Verein Gegen Vergessen – Für Demokratie e.V., Berlin and a Trustee of the Leo Baeck Institute, New York. He chairs the Board of Trustees of the Arthur Langerman Foundation, Berlin.

Mireille Hildebrandt is Research Professor on 'Interfacing Law and Technology' at Vrije Universiteit Brussels (VUB), appointed by the VUB Research Council, and co-Director of the Research Group on Law Science Technology and Society studies (LSTS) at the Faculty of Law and Criminology. Mireille also holds the part-time Chair of Smart Environments, Data Protection and the Rule of Law at the Science Faculty, at the Institute for Computing and Information Sciences (iCIS) at Radboud University Nijmegen. Her research interests concern the implications of automated decisions, machine learning and mindless artificial agency for law and the rule of law in constitutional democracies.

As tradition, also this year JURIX has been accompanied by satellite co-located events including workshops, tutorials and a Doctoral Consortium. We thank the workshops and tutorial organizers for their excellent proposals and for the effort involved in organizing the events. This year's edition comprises 1 tutorial "AI and Machine Learning – their benefits and drawbacks for use in ODR" and 6 workshops: AICOL 2022 – "AI Approaches to the Complexity of Legal Systems"; SORO – "Interdisciplinary Workshop on the Governance for Social Robots"; WAICOM – "Workshop on AI Compliance Mechanism"; LDA 2022 – "CEILI Workshop on Legal Data Analysis"; LN2FR – "Methodologies for Translating Legal Norms into Formal Representations"; AMPM 2022 "2nd Workshop in Agent-based Modeling & Policy-Making". AICOL is at its 14th edition and, traditionally, it represents a space to discuss models of legal knowledge more suitable to the complexity of contemporary legal systems. SORO is a new workshop which stems from a Japanese project about the governance for social robots through interdisciplinary approaches, including ethical considerations like deception, privacy, safety, etc. from their close interactions with humans. WAICOM represents a somehow related workshop which discusses legal and technical solutions about the adherence of AI's behaviour to legal and ethical principles. LDA, at its 8th edition, intends to focus on representation, analysis and reasoning with legal data. LN2FR explores the various challenges connected with the task of using formal languages and models to represent legal norms in a machine-readable manner. Finally, AMPM, at its 2nd edition, aims to create space for agent-based modeling, exploiting computation to investigate factual underpinnings of the legal phenomenon, like the intricate networks of cognitive, social, technological, and legal mechanisms through which law emerges, is applied, and exerts its effects.

Moreover, since 2013 and also this year, JURIX has offered young researchers, entering the AI&Law field as Ph.D. students, the opportunity to present their work during the Doctoral Consortium session, which represents an effective environment of growth and tutoring. For the coordination of this event, our special thanks go to Monica Palmi-

rani, Doctoral Consortium Chair and untiring coordinator of the Joint International Doctoral Degree in Law, Science and Technology (LAST-JD).

Organizing this edition of the conference would not have been possible without the support of many people and institutions. Special thanks are due to the local organizing team of the Institute of Legal Informatics at Saarland University, chaired by Georg Borges and Christoph Sorge, and to the Faculty of Law at Saarland University for sponsoring the event.

Moreover, we are particularly grateful to the 72 members of the Program Committee for their commitment and excellent work in a rigorous review process, including their active participation in the discussions concerning borderline papers. Finally, we would like to thank the former and current JURIX executive and steering committee members for their continuous support and advice, as well as for sharing experiences and suggestions about JURIX organization challenges.

Enrico Francesconi, JURIX 2022 Program Chair
Georg Borges, JURIX 2022 Conference Co-Chair
Christoph Sorge, JURIX 2022 Conference Co-Chair

This page intentionally left blank

About the Conference

Program Chair
Enrico Francesconi, IGSG-CNR, Italy

Conference Chairs
Georg Borges, Saarland University, Germany
Christoph Sorge, Saarland University, Germany

Doctoral Consortium Chair
Monica Palmirani, University of Bologna

Local Organizing Committee
Andreas Rebmann
Andreas Sesing-Wagenpfeil
Christina Anna Digeser
Diogo Sasdelli
Leslie Dennert
Magdalena Elisabeth Friedel
Marc Alexander Ostoja-Starzewski
Puria Sheikhipour Jouneghani

Program Committee
Tommaso Agnoloni, IGSG-CNR
Wolfgang Alschner, University of Ottawa
Francisco Andrade, University of Minho
Grigoris Antoniou, University of Huddersfield
Michał Araszkiewicz, Jagiellonian University
Kevin Ashley, University of Pittsburgh
Katie Atkinson, University of Liverpool
Trevor Bench-Capon, University of Liverpool
Floris Bex, Utrecht University
Michael Bommarito, Bommarito Consulting, LLC
Daniele Bourcier, CNRS
Karl Branting, The MITRE Corporation
Pompeu Casanovas, University Autonomous of Barcelona
Marcello Ceci, University of Luxembourg
Jack G. Conrad, Thomson Reuters
Giuseppe Contissa, University of Bologna
Claudia d'Amato, University of Bari
Luigi Di Caro, University of Torino
Rossana Ducato, University of Aberdeen and UCLouvain
Massimo Durante, University of Turin

Jenny Eriksson Lundström, Uppsala University
Aldo Gangemi, Università di Bologna & CNR-ISTC
Marco Giacalone, Vrije Universiteit Brussel
Tom Gordon, University of Postdam
Guido Governatori, Independent researcher
Jakub Harašta, Czechia Masaryk University
Mustafa Hashmi, Data 61, CSIRO
Jeff Horty, University of Maryland
John Joergensen, Rutgers University
Daniel Martin Katz, Chicago-Kent College of Law
Jeroen Keppens, King's College London
Tomer Libal, American University of Paris
Emiliano Lorini, IRIT
Réka Markovich, University of Luxembourg
Thorne Mccarty, Rutgers University
Elena Montiel-Ponsoda, Universidad Politécnica de Madrid
Paulo Novais, University of Minho
Gordon Pace, University of Malta
Ugo Pagallo, University of Turin
Monica Palmirani, University of Bologna
Ginevra Peruginelli, IGSG-CNR
Wim Peters, University of Aberdeen
Marta Poblet, RMIT University
Radim Polčák, Czechia Masaryk University
Henry Prakken, Utrecht University
Paulo Quaresma, Universidade de Evora
Vctor Rodrguez Doncel, Universidad Politécnica de Madrid
Antoni Roig, Autonomous University of Barcelona
Antonino Rotolo, University of Bologna
Giovanni Sartor, EUI/CIRSFID
Ken Satoh, National Institute of Informatics and Sokendai
Jaromir Savelka, Carnegie Mellon University
Burkhard Schafer, The University of Edinburgh
Erich Schweighofer, University of Vienna
Giovanni Sileno, University of Amsterdam
Barry Smith, SUNY Buffalo
Clara Smith, UNLP and UCALP
Sarah Sutherland, Canadian Legal Information Institute
Leon van der Torre, University of Luxembourg
Tom Van Engers, University of Amsterdam
Marc van Opijnen, KOOP
Bart Verheij, University of Groningen
Serena Villata, CNRS – Sophia-Antipolis
Fabio Vitali, University of Bologna
Vern Walker, Maurice A. Deane School of Law at Hofstra University
Bernhard Waltl, BMW Group
Yueh-Hsuan Weng, Tohoku University

Radboud Winkels, University of Amsterdam
Adam Wyner, Swansea University
Minghui Xiong, Zhejiang University
John Zeleznikow, Victoria University
Tomasz Zurek, T.M.C. Asser Institute, University of Amsterdam

JURIX Executive Committee

Tom van Engers, University of Amsterdam, president
Bart Verheij, University of Groningen, vice-president/secretary
Floris Bex, Utrecht University/Tilburg University, treasurer

JURIX Steering Committee

Michał Araszkiewicz, Jagiellonian University, Kraków
Katie Atkinson, University of Liverpool
Réka Markovich, Université du Luxembourg
Víctor Rodríguez-Doncel, Universidad Politécnica de Madrid

Sponsors

This page intentionally left blank

Contents

This page intentionally left blank

Full Papers

This page intentionally left blank

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220443

A Hybrid Model of Argument Concerning Preferences Between Statutory Interpretation Canons

Michał ARASZKIEWICZ[a,1]
[a]*Department of Legal Theory, Jagiellonian University in Kraków, Poland*
ORCiD ID: 0000-0003-2524-3976

Abstract. This paper extends the existing account of statutory interpretation based on argument schemes theory. It points out that the preference relations among statutory canons are not always determined by some predefined rules, but in certain systems of law or legal domains, it is necessary to argue these preference relations on the basis of case law. A set of factors favouring linguistic arguments and teleological arguments is presented, and a case-based argument scheme for the assignment of preference relations is reconstructed.

Keywords. Argumentation schemes, case-based reasoning, factors, hybrid systems, statutory interpretation

1. Introduction

Statutory interpretation is one of the most theoretically engaging and practically significant topics in law [1]. In order for statutory norms to be applied to specific cases or to be elaborated in legal scholarship, their meaning needs to be grasped by relevant actors. On a day-to-day basis, lawyers exchange arguments on how statutory law should be understood. The interpretive statements, i.e. the statements on what meaning should be assigned to statutory expressions, are supported or attacked through arguments based on the so-called interpretive canons or rules of statutory interpretation/construction [2,3,4].

The topic of statutory interpretation caught the attention of the AI and Law community during the times of elaboration of the first hybrid systems, joining the elements of rule-based and case-based reasoning [5] and has become one of the mainstream research topics therein in the 2010s [6,7,8,9]. The currently dominant approach to this subject is to represent reasoning with the interpretation of statutes using argumentation schemes theory [10], which enables its formalization in computational models of argument for defeasible reasoning [11,12,13]. These formalisms allow the representation of interpretive canons as defeasible rules and reasoning with preference relations defined on the set of arguments (hereafter, the Basic Setting) [9].

This paper, using the same general formal framework, extends the Basic Setting by introducing a layer of reasoning not broadly researched so far, namely case-based

[1] Corresponding Author: michal.araszkiewicz@uj.edu.pl. The article was financed by the National Centre for Sciences as part of research project agreement UMO-2018/29/B/HS5/01433.

reasoning with and about default preference relations between (classes of) interpretive canons. The model we develop is therefore a hybrid one as it joins rule-based reasoning (encompassed by the Basic Setting) and case-based reasoning to show how arguments based on past cases may be used to justify conclusions about the relative preference for canons. In the classical work on hybrid systems [5], the cases were used first and foremost as sources of information on the meaning of statutory terms; we extend this approach to show what other information, important for interpretive discourse, may be derived from past cases.

2. Interpretive Arguments: An Informal Exposition and Standard Formalization

2.1. Informal Exposition

Reasoning based on interpretive canons may be conveniently modelled using argumentation schemes theory [10]. The general scheme for arguments based on interpretive canons was formulated by Walton et al. [9]. In the generalized form, the capital letters E, D, M and C represent certain expression, document, meaning and canon, respectively, while the same small letters indicate a specific expression, document, meaning and canon. The relation "interpreted as" joining elements of E and M is a conceptual, extensional relation [6], for instance, an equivalence, difference, inclusion or exclusion relation.

Major premise: If the interpretation of E in D as M satisfies C's condition, then E should (not) be interpreted as M in D.

Minor premise: The interpretation of e in d as m satisfies c's condition.

Conclusion (interpretive statement): e should (not) be interpreted as m in d.

This interpretive argument scheme encompasses both positive and negative conclusions, that is, conclusions that favour either acceptance or rejection of a particular interpretation of the expression in question.

2.2. Example

Let us assume that in the village of Wooferton exists a rule in a local regulation, prohibiting anyone from entering the public library with dogs. In natural language, the rule reads as follows: *It is prohibited to enter public buildings with dogs.*

It may be subject to doubt whether actually all dogs are encompassed by the said prohibition. Let us present an argument based on a plain meaning canon (the major premise is left implicit; we also do not investigate whether this argument is actually convincing).

Minor premise: The interpretation of *dogs* in *Wooferton regulations* as *all dogs* satisfies the *plain meaning* condition.

Conclusion: *Dogs* should be interpreted as *all dogs* in *Wooferton regulations*.

However, this interpretation may be contested because, on its account, an assistance dog would not be allowed to the public building in Wooferton. This leads to doubts because then the rights of incapacitated persons would be unduly limited. We may therefore formulate the following teleological argument:

Minor premise: The interpretation of *dogs* in *Wooferton regulations* as *dogs kept for company or entertainment only* satisfies the *teleological interpretation* condition.

Conclusion: *Dogs* should be interpreted as *dogs kept for company and entertainment only* in *Wooferton regulations*.

Because assistance dogs are not kept for company and entertainment only, they are excluded from the prohibition.

Walton et al. [9] also formulated three general critical questions to arguments based on interpretive canons. It is also possible to determine specific sets of critical questions assigned to argument schemes based on particular canons [14]. Such specific critical questions may provide more definitive answers as to why a particular interpretive statement could be rejected.

If, in connection with a given problem, at least two interpretive statements concerning the same statutory expression are generated, it is necessary to determine which one should prevail. The Basic Setting enables all types of attacks on the arguments supporting or demoting particular interpretive statements. In particular, it is possible to express a preference relation between arguments supporting incompatible conclusions. The conclusion supported by the preferred argument should be accepted rather than another one.

2.3. A Formalization of the Basic Setting

Interpretive canons may be represented as defeasible rules in the general form [10]:

$$r: \varphi_1, \varphi_2, \ldots \varphi_n \Rightarrow \psi_1, \psi_2, \ldots \psi_n,$$

where φ and ψ are formulas in a logical language. The representation of interpretive canons as defeasible rules enables the application of the Defeasible Modus Ponendo Ponens Rule as the leading inference mechanism [15].

The authors [9] also introduce the function operator BestInt(E, D), which reads as the "best interpretation of E in D" and assumes the assignment of meaning to the expression as a result. The assignment of meaning is expressed by set-theoretical relations such as equivalence ($\equiv$), difference ($\not\equiv$) and inclusion ($\sqsubseteq$).

Oftentimes, it will be possible to generate at least two different interpretations of the same expression of the same document, and it will not be possible to accept both of them. In such a situation, it may be necessary to compare the strength of arguments pleading for incompatible conclusions. The Basic Setting [9] introduces a priority relation between both classes of and instances of canons, expressed in the following general pattern:

$$C_1(E, D, M_i) \succ C_2(E, D, M_j),$$

which reads that canon C_1 assigning meaning M_i to expression E in document D has priority over canon C_2 assigning meaning M_j to expression E in document D. Such a statement provides a defeasible basis for accepting the conclusion of arguments based on C_1 rather than on C_2.

These basic considerations enable the formalization of reasoning with arguments based on interpretive canons, including all types of attacks between arguments (undermining, undercutting, and rebuttal) and with preferences between arguments, with the use of computational models of arguments enabling the generation of extension on the basis of Dung-style semantics [12,13,16].

3. Case-Based Reasoning about Preferences for Interpretive Canons

3.1. Informal Exposition

How does one justify a preference relation between (instances of) interpretive canons? Of course, there exist different possibilities, and the actual persuasive force of particular arguments may vary across jurisdictions. One of the most important insights following from the comparative research is that it is possible to reconstruct general preference relations between classes of interpretive canons: canons based on language have some default priority over other types of arguments. However, canons based on systemic considerations and, eventually, teleology may rebut the former if there exists some additional reason to depart from the result following from the argument that has default priority [2]. It is also possible to reconstruct more specific rules: Walton et al. [9] refer to the work of Alexy and Dreier [17] who have reconstructed a set of such rules; for instance, in the field of criminal law, the ordinary meaning canon has a priority over arguments based on technical meaning.

However, the reconstruction of such rules is not always possible, and even if they are reconstructed, such rules are subject to doctrinal criticism and to multidirectional evolution in case law. For instance, the general default rule model [2] is contemporarily viewed as obsolete in many jurisdictions and in the context of some domains of law, especially where the regulation serves as the implementation of EU law, which is one of the factors leading to a relatively stronger position of canons based on the purpose of regulation. In actual legal practice, except for some specific contexts, establishing a preference relation among canons of interpretation is a matter of balance of reasons rather than the application of generic rules assigning default priorities. These findings suggest that the assignment of preference relations between the canons for a given class of cases may be modelled by means of case-based reasoning (CBR) patterns.

The standard paradigm for modelling CBR in AI and Law is a factor-based approach initiated by HYPO [18] and developed in multiple directions [19]. A factor is typically defined as a "stereotypical fact pattern", i.e. a generalization from the description of facts of the case, which serves as a reason for a certain outcome of the case (e.g. to decide for the plaintiff or the defendant). Recently, it has been rightly pointed out that factors should be assigned to particular issues rather than to final outcomes directly [20]. This approach helps, in particular, to represent the multi-layered structure of legal reasoning. Accordingly, interpretive dispute also involves deciding certain issues, including whether a specific canon or a class of canons should be preferred to another one.

Classical factors are generalizations from the facts of cases in particular domains. Such factors may also play a role in the solving of interpretive problems, but as far as the disputes about the relative preference for canons are concerned, typically different categories of factors are considered: factors that represent features of the types of statutory regulation, the types of legal norms etc. Below, we present an example list of factors that tend to support either preferring linguistic canons over teleological ones or vice versa. We represent these factors as binary, following the approach initiated in the CATO system [21] and adopted, for instance, in [11]. It should be noted that some of these factors may be interpreted as gradual, enabling representation with magnitudes and dimensions [18,22]; we leave this possibility, however, for future work. Factors that support preference for linguistic arguments use the symbol PrefLing, while those supporting preference for teleological arguments use the symbol PrefTel.

Factors supporting preference for linguistic canons over teleological ones

F1(PrefLing). The interpreted expression is a part of a prohibition directed to an individual.

F2(PrefLing). The interpreted expression imposes sanctions.

F3(PrefLing). The interpreted expression restricts individual liberty or rights.

F4(PrefLing). The interpreted expression is a part of the norm, which is an exception to a more general rule.

F5(PrefLing). The interpreted expression is defined legally.

F6(PrefLing). The interpreted expression imposes a duty on an individual towards a public authority.

F7(PrefLing). The interpreted expression represents the power of a public authority.

F8(PrefLing). The values realized by the regulation are recognized to the greatest extent by literal interpretation.

Factors supporting preference for teleological canons over linguistic ones

F1(PrefTel). The interpreted expression grants rights or liberty to an individual.

F2(PrefTel). The interpreted expression is a part of a general principle.

F3(PrefTel). The interpreted expression is a general clause.

F4(PrefTel). The interpreted regulation implements a European Union law.

F5(PrefTel). The interpreted regulation concerns a constitutional value.

F6(PrefTel). The interpreted regulation concerns human rights.

F7(PrefTel). The interpreted expression concerns the mutual rights and duties of equal persons.

F8(PrefTel). The values realized by the regulation are recognized to the greatest extent by restrictive interpretation.

F9(PrefTel). The values realized by the regulation are recognized to the greatest extent by extensive interpretation.

The above list of factors enables the construction and evaluation of arguments based on CATO-style reasoning, including citing on-point cases (i.e. cases sharing at least one factor with the problem) and distinguishing arguments as well as different types of counterexamples [18,21]. The presented list also enables the development of a factor hierarchy in the style of CATO.

3.2. Example Continued

For the sake of application of this framework, let a case from the case base be characterized by two finite, possibly empty, sets of factors relevant to the issue of relative preference among interpretive canons (F(PrefLing) and F(PrefTel)), the adopted preference relation among canons ($SetOfCanons_a \succ SetOfCanons_b$) and the specific preference relation between specific canons and the scope of their application in a given problem: $C_1(E, D, M_i) \succ C_2(E, D, M_j)$. Let us note that the "case" in this sense does not have to concern any set of events (real or hypothetical); it may only concern an interpretive problem concerning an existing or hypothetical regulation.[2]

Case = $\{F_{1, \ldots, n}(PrefLing); F_{1, \ldots, n}(PrefTel); SetOfCanons_a \succ SetOfCanons_b; C_1(E, D, M_i) \succ C_2(E, D, M_j)\}$

For instance, let us assume that the Wooferton regulations, discussed in the previous section, have been the subject of a litigation process and that the court found that the teleological canons should have priority over the linguistic ones and thus ruled that "dogs" should be interpreted as "dogs kept for company and entertainment only". Let us further assume that the court found that the interpreted expression is a part of prohibition directed to an individual (F_1(PrefLing)) and the interpreted expression restricts individual liberty or rights (F_3(PrefLing)), but at the same time the interpreted regulation concerns constitutional values and human rights (protection of health, equal access to culture): F_5(PrefTel) and F_6(PrefTel). These values are best realized through restrictive interpretation: F_8(PrefTel). Therefore, the representation of the Wooferton case could be as follows, where "purposive" and "plain language" are names of specific canons (instantiations of "C_1" and "C_2"):

Case 1: Wooferton = $\{F_1$(PrefLing), F_3(PrefLing); F_5(PrefTel), F_6(PrefTel), F_9(PrefTel); Tel $\succ$ Ling; purposive(dogs, Wooferton regulations, [kept_for_company_or_entertainment_only]dogs $\succ$ plain language(dogs, Wooferton regulations, dogs}

Let us now consider a problem case: in the neighbouring town of Cat Hill, there exists a regulation that states the following: *It is prohibited to enter the public playground with any animals dangerous to children, including dogs and cats.*

An interpretive doubt emerges: should the prohibition encompass all dogs and cats *a limine*, or does it concern only such dogs and cats that are dangerous as individual animals? Let us assume that two alternative interpretations exist, one suggested by the plain meaning canon:

Any animals dangerous to children, including dogs and cats = any animals dangerous to children, including dangerous dogs and dangerous cats.

[2] For the sake of simplicity, we assume that only one preference relation between sets of canons was expressed, and that only the preference between the set of linguistic canons and the teleological canons was considered. A generalization is possible.

The other one, extensive with regard to the former one, enhancing the protection of the safety of the children:

Any animals dangerous to children, including dogs and cats = animals at least potentially dangerous to children, including all dogs and all cats

Taking into account the factor overlap between Wooferton and the problem, the former case may be persuasively cited to bring the following solution to the Cat Hill case:

Case 2: Cat Hill = {F_1(PrefLing), F_3(PrefLing); F_5(PrefTel), F_6(PrefTel), F_9(PrefTel); Tel $\succ$ Ling; purposive(any animals dangerous to children, including dogs and cats, CatHill regulations, [at_least_potentially_dangerous_to_children]animals $\supseteq$ [dogs $\wedge$ cats]) $\succ$ plain language(any animals dangerous to children, including dogs and cats, CatHill regulations, [dangerous]animals $\supseteq$ [dangerous]dogs $\wedge$ [dangerous]cats)}.

The proposed approach enables representation of not only preference relations between (the instances of) interpretive canons based on predefined, e.g. doctrinal, rules but also the preference relations extracted from the (evolving) case law. The approach proposed by Prakken and Sartor [11] enables the extraction of factor-based rules and preferences between them from case bases. On the layer considered here, such rules will assume collections of factors relevant for the establishment of preference relations among canons as their antecedents and the default assignment of such a preference relation as their consequents.

3.3. Argument Scheme

It is possible to reconstruct an argument scheme based on the overlap of relevant factors between the problems and lead to a conclusion concerning the default preference relation between the classes of factors.

Argument Scheme for Case-Based Default Preference for Interpretive Canons

Major Premise: There exists case c in a case base such that
c = {$F_{1, ..., n}$(PrefLing); $F_{2, ..., n}$(PrefTel); $SetOfCanons_a \succ SetOfCanons_b$; C_1(E, D, M_i) $\succ$ C_2(E, D, M_j)}

Minor Premise: The problem situation is characterized by the set of factors {$F_{1, ..., n}$(PrefLing); $F_{2, ..., n}$(PrefTel)} and F(c) $\cap$ F(p) $\neq$ Ø (on-pointness).

Conclusion: In case p, $SetOfCanons_a \succ SetOfCanons_b$ and eventually C_1(E, D, M_i) $\succ$ C_2(E, D, M_j) should be accepted.

As this argument scheme is a subtype of factor-based argument based on an earlier case, it is subject to characteristic attacks based on distinguishing (pointing out the differences between the cited case and the problem) or on counterexamples (indicating that on-point cases other than the cited one suggest a different conclusion). In the theory of argumentation schemes, such types of attacks may be codified in the list of critical questions; a list of critical questions assigned to an interpretive argument based on

precedent was presented in [14], and they may be used accordingly with regard to a case-based argument concerning the preference relation between interpretive arguments.

4. Discussion and Related Work

Case-based reasoning patterns used for statutory interpretation have been discussed extensively since the early work on hybrid systems [5]. An argument based on precedent is one of the widely recognized canons of statutory interpretation [2]. The argumentation scheme approach, as advocated by [9], is able to express interpretive arguments based on earlier cases. However, these cases may serve as the source of information not only about how statutory expressions should be interpreted but also about the (default) preference relations between the interpretive canons. In certain contexts, these preference relations may be expressed in the form of relatively definitive rules, formulated a priori, for instance, by legal doctrine [17], but there exist jurisdictions and domains of law where such preference relations must be reconstructed from the case law. This paper present a list of factors that are relevant for the purpose of establishing such preference relations, which may be understood as one of the issues [20] important for solving an interpretive problem.

For simplicity, we assumed that these factors are binary, as in CATO [21] or Prakken and Sartor's model [11]. However, it should be noted that at least some are, in principle, fit for representation through factors with magnitudes; for instance, one might consider the *degree* of restriction of an individual's liberty or rights (F3(PrefLing)). This opens a possibility for integrating the existing proposal with some recent approaches concerning reasoning with precedent [20,22,23,24,25].

There also emerges the question of whether judicial reasoning about the preference relations between interpretive canons should be modelled with the *result model* or the *reason model* [22]. If descriptive adequacy is adopted as the methodological goal, the result model seems fit for the modelling of rationales where the court uses a more magisterial style, leaving the reasons for adopting a particular preference relation implicit, while the latter seems more appropriate for the modelling of argumentative rationales.

A system of statements that play a role in case-based reasoning, focused on the context of common law tradition, has been developed in the AI and Law community [26]. Such a system of statements also needs to be developed in the context of statutory interpretation for both Anglo-American and continental law traditions.

5. Conclusions and Future Work

This paper extends the currently dominating approach to the modelling of statutory interpretation using argumentation scheme theory by introducing case-based reasoning patterns for arguments concerning the preference relations between the interpretive canons. The resulting model is a hybrid one, as it encompasses the rule-based approach following the formalization of argumentation scheme theory and the model of reasoning with precedent introduced by Prakken and Sartor [11] with the classical case-based reasoning patterns discussed in the CATO system. Importantly, the reconstructed set of factors is related to the specific type of issue, namely, to the assignment of priority between the (categories of) canons; hence, it influences the final outcome of any particular case only indirectly. The methodological aim of this work is to bridge the gap

between logical, normative models of legal argument and a more descriptive stance, which aims to represent the diversity of reasoning patterns as they are expressed in the rationales of judicial decisions and analysed in legal doctrine and theory. For the domains where courts apply more argumentative style of rationale drafting and this expresses the rules on which it bases the issue resolution, the reason model [22, 23, 24] may be found more adequate.

Further extensions of the developed framework encompass the inclusion of different layers of interpretive reasoning, such as the interpretation of conditions of interpretive canons themselves (the concepts such as "literal meaning", or "legislative intent" are subject to numerous controversies) and the representation of controversial distinction between interpretive reasoning and reasoning concerning legal classification or subsumption [27]. The extended model should also support teleological considerations, as discussed in the field of computational models of legal argument [28,29,30].

The presented model offers a version of a case-based argument scheme for ascription of preference between interpretive canons. Arguments schemes for case-based reasoning were discussed in [31] and in [32] which calls for a comparative evaluation.

Another important aspect of the extension of the model is the inclusion of the evolution of interpretive considerations over time, which may be documented in the case law. Such a model would represent not only how the understanding of statutory concepts changes over time [33,34,35] but also how the opinions concerning the relative strength of the interpretive canons and the understanding of the canons themselves change. Referring to past cases may not only stabilize the meaning of statutory terms [36] but also the preference relations among the canons in given domains of law.

Finally, the conceptual framework provided by the multi-layered model of legal reasoning should serve as the basis of an extensive annotation system enabling legal prediction with the use of Natural Language Processing tools based on machine learning (ML)[37]. The problems of legal interpretation have become the subject of interest for ML researchers interested in information retrieval and predictive analytics [38,39].

References

[1] Dickson J. Interpretation and coherence in legal reasoning. In: Zalta EN, editor. The Stanford encyclopedia of philosophy. Winter 2016 ed. Center for the Study of Language and Information (CSLI), Stanford University; 2001. Available from: https://plato.stanford.edu/archives/win2016/entries/legal-reas-interpret/.

[2] MacCormick N, Summers R, editors. Interpreting statutes. Routledge; 1991.

[3] Jellum L, Hricik D. Modern statutory interpretation. Problems, theories and lawyering strategies. 2nd ed. Carolina Academic Press; 2009.

[4] Wróblewski J. The judicial application of law. Springer; 1992.

[5] Skalak D., Rissland E. Arguments and cases: an inevitable intertwining. Artif Intell Law. 1992;1(1):3-44.

[6] Araszkiewicz M. Towards systematic research on statutory interpretation in AI and law. In: Ashley K, editor. JURIX 2013, vol. 235 of Frontiers in Artificial Intelligence and Applications.IOS Press; 2013. p. 15 -24.

[7] Araszkiewicz M, Zurek T. Comprehensive framework embracing the complexity of statutory interpretation. In: Rotolo A, editor. Legal knowledge and information systems – JURIX 2015, vol. 279 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2015. p. 145-8.

[8] Rotolo A, Governatori G, Sartor G. Deontic defeasible reasoning in legal interpretation: two options for modelling interpretive arguments. In: ICAIL '15: Proceedings of the 15th International Conference on Artificial Intelligence and Law. ACM; 2015.p. 99-108.

[9] Walton D, Sartor G, Macagno F. An argumentation framework for contested cases of statutory interpretation. Artif Intell Law. 2016;24:51-91.

[10] Walton D, Reed C, Macagno F. Argumentation schemes. Cambridge University Press; 2008.

[11] Prakken H, Sartor G. Modelling reasoning with precedents in a formal dialogue game. Artif Intell Law. 1998;6:231-87.
[12] Gordon TF, Prakken P, Walton D. The Carneades model of argument and burden of proof. Artif Intell. 2007;171(10-11):875-96.
[13] Prakken H. An abstract framework for argumentation with structured arguments. Argument Comput. 2010;1(2):93-124.
[14] Araszkiewicz M. Critical questions to argumentation schemes in statutory interpretation. J Appl Log. 2021;8:291-320.
[15] Verheij B. Dialectical argumentation with argumentation schemes: an approach to legal logic. Artif Intell Law. 2003;11(1-2):167-195.
[16] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif Intell. 1995;77(2):321-57.
[17] Alexy R, Dreier R. Statutory interpretation in the Federal Republic of Germany. In: MacCormick N, Summers R, editors. Interpreting statutes. Routledge; 1991.
[18] Ashley K. Modeling legal arguments: reasoning with cases and hypotheticals. MIT Press; 1991.
[19] Bench-Capon T. HYPO'S legacy: introduction to the virtual special issue. Artif Intell Law. 2017;25:205-50.
[20] Bench-Capon T, Atkinson K. Precedential constraint: the role of issues. In: ICAIL '21: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. ACM; 2021. p. 12-21.
[21] Aleven V. Teaching case-based argumentation through a model and examples. PhD thesis, University of Pittsburgh; 1997.
[22] Horty J. Reasoning with dimensions and magnitudes. Artif Intell Law. 2019;27:309-45.
[23] Horty J. Modifying the reason model. Artif Intell Law. 2021;29:271-85.
[24] Prakken H. A formal analysis of some factor- and precedent-based accounts of precedential constraint. Artif Intell Law. 2021;29:559-85.
[25] Bench-Capon T., Atkinson K. Argument Schemes for Factor Ascription. In: Proceedings of COMMA 2022, IOS Press, p. 68-79.
[26] Al-Abdulkarim L, Atkinson K, Bench-Capon T. Statement types in legal argument. In: Bex F, editor. Legal knowledge and information systems – JURIX 2016: The Twenty Ninth Annual Conference. IOS Press; 2016. p. 3-12.
[27] Wank R. Juristische Metodenlehre. Eine Einleitung für Wissenschaft un Praxis. Vahlen; 2020.
[28] Berman DH, Hafner CD.Representing teleological structure in case-based legal reasoning: the missing link. In: Proceedings of the Fourth International Conference on Artificial Intelligence and Law. ACM; 1993. p. 50-59.
[29] Bench-Capon T, Sartor G.A model of legal reasoning with cases incorporating theories and values. Artif Intell. 2003;150(1):97-143.
[30] Maranhão J, de Souza EG, Sartor G. A dynamic model for balancing values. In: ICAIL '21: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. ACM; 2021. p. 89-98.
[31] Wyner A., Bench-Capon T. Argument Schemes for Legal Case-based Reasoning. In: Lodder A., Mommers L. editors. Legal Knowledge and Information Systems – JURIX 2007: The Twentieth Annual Conference. IOS Press; 2007. p. 139-149.
[32] Prakken H., Wyner A., Bench-Capon T., Atkinson K. A formalisation of argumentation schemes for legal case-based reasoning in ASPIC+. Journal of Logic and Computation 2015;25(5): 1141-1166
[33] Levi EH. An introduction to legal reasoning. Univ Chic Law Rev. 1948;15(3):501-74.
[34] Henderson J., Bench-Capon T. Describing the Development of Case Law. In: ICAIL'19: Proceedings of the Seventeenth International Conference on AI and Law. ACM; 2019. p. 32-41.
[35] Bench-Capon T., Henderson J. A Dialogical Model of Case Law Dynamics. In: Araszkiewicz M., Rodríguez-Doncel V. editors. Legal Knowledge and Information Systems - JURIX 2019: The Thirty-second Annual Conference. IOS Press; 2019. p. 163-168.
[36] Savelka J., Ashley K. On the Role of Past Treatment of Terms from Written Laws in Legal Reasoning. In:Rahman S., Matthias Armgardt M., Nordtveit Kvernenes H. Ch. Editors. New Developments in Legal Reasoning and Logic: From Ancient Law to Modern Legal System, Springer 2022, p. 379-95.
[37] Ashley K. Artificial intelligence and legal analytics, new tools for law practice in the digital age. Cambridge University Press; 2017.
[38] Savelka J, Ashley K. Legal information retrieval for understanding statutory terms. Artif Intell Law. 2022;30(2):245-89.
[39] Habernal I et al. Mining legal arguments in court decisions. Available from: https://arxiv.org/abs/2208.06178.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220444

Implementing a Theory of a Legal Domain

Trevor BENCH-CAPON [a] and Thomas F GORDON [b]
[a] *University of Liverpool, UK*
[b] *Berlin, Germany*

Abstract. We describe a system for constructing, evaluating and visualising arguments based on a theory of a legal domain, developed using the Angelic methodology and the Carneades argumentation system. The visualisation can be used to explain particular cases and to refine and maintain the theory. A full implementation of the well known US Trade Secrets Domain is used to illustrate the process.

Keywords. legal argumentation, Angelic methodology, Carneades

1. Introduction

Modelling reasoning with legal cases has been a central concern of AI and Law from the very beginning. A good deal of the research has built on the pioneering work of HYPO [1] and CATO [2]. Both of these addressed cases in US Trade Secrets law, which is the domain which we will model in this paper. Over the years a series of stages in reasoning with legal cases have been identified [3]. The outcome is decided on the basis of the resolution of a number of issues which set out what must be shown to establish a claim. The relationship between the issues and the outcome can be expressed in a set of rules [4]. The issues are resolved by weighing the reasons, generally called *factors*, to resolve that issue for the plaintiff against the reasons to resolve that issue for the defendant (e.g. [2]). Preferences between these reasons are (in common law domains) derived from the decisions in previous, precedent, cases [5]. Often the set of factors describing the cases is taken as given, but in some cases the ascription of factors is itself a matter of controversy which must be resolved using precedents [6]. This structure lends itself to a hierarchical representation, as in [2], with the outcome as the root, issues at the upper layers, abstract factors in the middle layers and the base level factors as the leaves. This was used in [7] and extended in [8] to allow the base level factors to be ascribed on the basis of a series of questions answered by the user.

We will describe how to realise this approach to produce a system which will model the domain theory and, when given a particular case, will produce an argument map showing what was accepted and what was rejected in the case, and the arguments which justified these positions. The argument map provides a visual explanation of the reasoning in the case, and, where the result is unexpected, provides a means to identify how to perform the required corrective or adaptive maintenance of the theory.

Section 2 describes the methodology used to specify the domain theory and the system used to make the theory executable. The transition from theory to implementation is

Table 1. Sample fragment of ADP for Trade Secret Misappropriation from [9]

Node	Children	Acceptance Conditions	Justification
InfoValuable	F6p F8p F11d F15p InfoObtainable	REJECT IF F11d ACCEPT IF F8p ACCEPT IF F6p ACCEPT IF F15p REJECT IF InfoObtainable ACCEPT	Silfen Lewis Mason College Restatement
InfoObtainable	F15p F16d F20d F24d	REJECT IF F15p ACCEPT IF F16d OR F24d OR F20d REJECT	College Ferranti

described in Section 3, while Section 4 discusses how the argument map can be used for explanation and maintenance. Finally, Section 5 offers some discussion and concluding remarks.

2. Background

2.1. The Angelic Methodology

The Angelic methodology, which is intended to produce a theory of a legal domain, was introduced in [7] and has subsequently been refined in a series of projects, including several with the law firm Weightmans, one of which is described in [8]. The methodology has two outputs: the Angelic Design Proforma (ADP), originally called ADF, and a set of questions required to instantiate the ADP for a particular case.

The ADP comprises a table with four columns, in which the rows describe nodes in a hierarchy. Example rows, taken from the full example given in [9], are given in Table 1. The first column gives the ID of the node, which is intended to be an informative label for the node. The second column gives the children of the node. The third column gives a list of prioritised reasons to accept or reject the node. These reasons use only the children of the node, and conclude with a default in case none of the reasons are satisfied. The final column gives the source of these reasons and priorities between them which may be a statute, a commentary, a precedent case or any other authoritative source.

The hierarchy is essentially the factor hierarchy of CATO [2], with the outcome as the root, issues at the upper layers, abstract factors in the middle layers, and base level factors as the leaves, with the addition of acceptance conditions, inspired by Abstract Dialectical Frameworks [10]. The questions are intended to be posed to the user to instantiate the leaf nodes for a particular case. A leaf node may correspond to a single answer, or be derived from one or more answers. An example question is discussed in Section 3.2.

2.2. Carneades

Carneades is both a formal model of structured argument and a software implementation of this formal model. The original version of Carneades [11] provided a recursive procedure to evaluate argument graphs, given a set of assumptions, to label the statement nodes acceptable (In) or not acceptable (Out), and a way to visualise the output in "argument maps" [12].

Here we are using the latest version of Carneades (4.3). The formal model of this version was first presented in [13]. While it supports all the features of the earlier version, the formal model is quite different. These differences were motivated by the desire to handle not just attack relations among arguments, but also the *balancing* of competing arguments, so as to be better able to support practical reasoning, where the pros and cons of alternative options are weighed, including support for multi-criteria decision analysis (MCDA). This also enables support for cumulative arguments, where the failure of a premise can weaken an argument without defeating it completely. To achieve these goals, the structure of argument graphs is now tripartite, with issue, statement and argument nodes, and arguments have been extended with assignable and customisable weighing functions [14]. When evaluating argument graphs, statements are now labelled either In, Out or Undecided. The formalisation of issues assures that at most one option (position) is In, serving as a constraint. The formal model now uses fixed-point semantics when evaluating argument graphs, so as to be able to handle cycles in argument graphs. Just as in abstract argumentation, various semantics can be applied, such as grounded, preferred or complete semantics. The implementation, however, only supports grounded semantics, which we have found to be sufficient for our legal application scenarios, including the trees produced by Angelic. As before, the implementation provides a way to visualise argument graphs in argument maps.

Another new feature of this version of Carneades is its provision of a language and inference engine for argumentation schemes. The language is based on Constraint Handling Rules [15], a forwards-chaining rule system with a declarative, logical semantics. In our implementation of Constraint Handling Rules [16], every time a rule is applied (fired), an argument is generated as a side-effect. Given a set of argumentation schemes (rules) and a set of assumptions, the rule engine is first applied to generate an argument graph and then the argument graph is evaluated to label the statements in the graph. Finally, as before, the resulting argument graph can be visualised in an argument map.

3. Realising an Angelic Theory in Carneades

To make the theory executable it is necessary to represent the acceptance conditions and to get the assumptions from the user to instantiate a particular case. Because the underlying conceptions of Angelic and Carneades are very similar, both conceptualising the reasoning as forming an argument graph, each acceptance condition can be represented as a Carneades scheme. For moving from question responses to factors we distinguish three kinds of factors, as discussed in Section 3.2.

3.1. Representing the acceptance conditions

Let us use the acceptance conditions for the InfoObtainable node of the trade secrets ADP, shown in Table 2 below, as an example. Each node in the ADP has four properties (columns in the table): a node name; the children of the node, used in the acceptance conditions; a set of prioritised acceptance conditions; and the source of each condition, the case or text on which it is based. As we will see, all of this information in an ADP maps directly into Carneades argumentation schemes. The first acceptance condition can be represented as in Figure 1:

```
id: ioCollege
meta:
    source: College
weight:
    constant: 1.0
conclusions: [notio]
premises: [f15p]
```

Figure 1. First Acceptance Condition for Information Obtainable

Here, `ioCollege` is an identifier for the scheme. The `meta` property can be used to annotate the scheme with any desired information. Here we have used it to provide the source of the scheme, the precedent case *College Watercolor Group, Inc. v. William H. Newbauer, Inc* (*College*). The `weight` property is used only to order the schemes for a node, so that its particular value has no other significance. Here we are using constant weights for this purpose. Carneades also provides a variety of weighing functions which can be used to compute weights. This feature will be demonstrated later, when showing how factors with magnitude and dimensions can be handled. Using weights, arguments can be partially ordered, useful for giving acceptance conditions with alternative premises the same weight, as will be demonstrated below. The conclusion of this scheme, `notio`, means that the information was not obtainable, and denotes the negation of `io`, that the information was obtainable. The conclusion indicates whether the node in the ADP being represented (InfoObtainable) is accepted or rejected. The premise of this example scheme is `f15p`, denoting the F15p factor, Unique-Product, the body of the acceptance condition. Schemes may have multiple conclusions and premises, although this feature is not demonstrated in this example.

To make `io` and `notio` conflict, so that at most one of them can be labelled In, an issue scheme is added as in Figure 2:

To complete this example, the schemes for the remaining acceptance conditions for the InfoObtainable node of the ADP in Table 2 are shown in Figure 3:

The constant weights used in these schemes simply enforce the ordering of the acceptance conditions in the ADP. Any real numbers could have been used, so long as they preserve the desired ordering. Notice that two schemes were needed to represent the acceptance conditions for the *Ferranti* precedent, since it has two alternative premises,

```
issue_schemes:
    io: [io, notio]
```

Figure 2. Issue Scheme Information Obtainable

```
- id: ioFerranti1
  meta:
      source: Ferranti
  weight:
      constant: 0.9
  conclusions: [io]
  premises: [f24d]
```

```
- id: ioFerranti2
  meta:
     source: Ferranti
  weight:
     constant: 0.9
  conclusions: [io]
  premises: [f20d]
```

```
- id: ioDefault
  weight:
     constant: 0.1
  conclusions: [notio]
```

Figure 3. Argumentation Schemes for Information Obtainable

F24d OR F20d. Both of these schemes were given the same weight, 0.9, since they share the same position in the ordered list of acceptability conditions in the ADP.

3.2. Moving from facts to factors

Most factors are simply Boolean and depend on a single fact, and so they can be assumed directly on the basis of particular question answers. Other factors require some judgement to be ascribed to cases on the basis of the facts. These are the factors with magnitude [17]. Two types of such factors are used: those derived from a single dimension and which apply a threshold to determine whether the factor applies and those (e.g. CompetitiveAdvantage in CATO) ascribed on the basis of a weighted sum of two dimensions.

We illustrate this using the question relating to disclosures.

Q3 Was the Information disclosed (Check all that apply)
(a) In negotiations with the Defendant?
(b) To employees?
(c) To sub-contractors?
(d) To customers?
(e) To the public?
(f) Restrictions were placed on the disclosures
(g) The information was not disclosed

Answers (a) and (f) lead directly to factors F1d and F12p respectively, while (g) leads to no factor. Answers (b)-(e) represent the four points on the dimension leading to the ascription of F10p (InfoNotDisclosedOutsiders) and F10d (InfoDisclosedToOutsiders). Which factor applies depends on where the threshold for being an *outsider* is drawn, in the light of precedents and other domain knowledge.

3.2.1. Thresholds

In the trade secrets domain, answers (b)-(e) to the question about whether information has been disclosed to outsiders form a *dimension*, which can be satisfied to a greater or lesser extent. Let us identify the following points along the disclosure dimension, based on to whom the information has been disclosed: employees, subcontractors, customers and the public. These values will need to be mapped to factors, to show which party is favoured on the dimension. Where the line is drawn is established in precedents. Based on an analysis of the cases, we want to map disclosures to employees and subcontractors to the factor F10p (InfoNotDisclosedOutsiders) but map disclosure to customers and the public to F10d (InfoDisclosedToOutsiders), where F10p and F10d are alternative positions of an issue, so that at most one of these factors may be In. Effectively, the dimension will be partitioned with a threshold between the subcontractors and customers points. Moreover, we want to use argument weights to preserve the information about the relative position of the disclosure along the dimension. So that, for example, a disclosure to the public is a stronger argument for F10d than a disclosure to customers. And, conversely, so that a disclosure to employees is a stronger argument for F10p than a disclosure to subcontractors. One way to achieve these goals, using the multi-criteria decisions analysis (MCDA) weighing function provided by Carneades is shown in Figure 4.

This use of the MCDA weighing function does not illustrate its full capabilities, since there is only one dimension, “disclosed”, where typically MCDA will be used to combine multiple dimensions using a weighted sum. Notice that the permitted values

```
- id: ido1
  variables: [X,Y]
  weight:
    criteria:
      soft:
        disclosed:
          factor: 1
          values:
            customers: 0.75
            public: 1.0
  premises:
    - disclosed(X,Y)
  conclusions: [f10d]
```

```
- id: ido2
  variables: [X,Y]
  weight:
    criteria:
      soft:
       disclosed:
           factor: 1
           values:
              employees: 1.0
              subcontractors: 0.75
  premises:
    - disclosed(X,Y)
  conclusions: [f10p]
```

Figure 4. Argumentation Schemes for Disclosures To Outsiders

for the disclosed dimension (property) differ between these two schemes. The customers and public values are in the first scheme, and the employees and subcontractors values are in the second scheme. Alternatively, one could list all the values along the disclosure dimension in both schemes, but assign the weight 0.0 to the other values. Notice also that the argument weights do not increase from the beginning to the end of the dimension, but rather increase from the threshold in the middle out to the points on either end of the dimension. Finally, notice that the disclosed property declares a binary predicate, `disclosed/2`. Although not needed for this application, this feature is used when addressing practical reasoning. In the next section, we make fuller use of the expressivity of the MCDA weighing function to represent the mapping of two dimensions to a factor.

3.2.2. Weighted Sums

In the trade secrets domain, whether or not the defendant has obtained a competitive advantage due to acquiring knowledge of the trade secret is a function of two dimensions, the costs saved and the time saved by the defendant. Lesser time savings can be compensated by greater cost savings, and vice versa. If both time and cost savings have been substantial, the argument for having obtained a competitive advantage will be greatest.

Here we demonstrate how to map several dimensions (time and cost savings) to an abstract factor, (F9d, NoCompetitiveAdvantage) using a Carneades scheme with an MCDA weighing function shown in Figure 5.

The weight of the argument is computed by a weighted sum of the values for each dimension. The relative weight of each dimension is given by a "factor". In this example, time saved is given twice the weight of costs saved. Any integer or real number may be assigned as a weight. The weighted sum is determined by first computing the relative portion of each (Carneades) factor compared to the sum of the factors for all the dimensions. Thus, in this example, time gets 2/3 of the weight and costs 1/3.

The second scheme here, with the id `caThreshold`, sets the threshold weight which an argument for F9d (NoCompetitiveAdvantage) must have in order to succeed. We have set the threshold at 0.5, meaning that the weighted sum of the time and cost savings properties must be greater than 0.5 in order for F9d to be derived (labelled In). Otherwise F8p (CompetitiveAdvantage) will be In.

```
- id: ca1
  variables: [X,P,T]
  weight:
    criteria:
      soft:
        costs_saved:
          factor: 1
          values:
            very_large: 0.0
            significant: 0.25
            small: 0.5
            more_expensive: 0.75
        time_saved:
          factor: 2
          values:
            very_large: 0.0
            significant: 0.25
            small: 0.5
            took_longer: 0.75
  premises:
    - costs_saved(X,P)
    - time_saved(X,T)
  conclusions: [f9d]
```

```
- id: caThreshold
  weight:
    constant: 0.5
  conclusions: [f8p]
```

Figure 5. Argumentation Schemes for Competitive Advantage

4. Explanation From The Argument Map

The graph can be used to construct an explanation according to the well known Issue-Rule-Application-Conclusion method, widely used in US Law Schools and advocated in [18]. In this context the issue is the main point of contention in the case, rather than the top level nodes of the CATO hierarchy: thus any node may be the issue. A red node with a green child will indicate an issue, because the alternative position will have been preferred. In the case of *MBL (USA) Corp. v. Diekman* the relevant node is InfoValuable (see Figure 6). This is supported by InfoKnownToCompetitors, but excluded because SecurityMeasures is preferred, citing *Mason v. Jack Daniel Distillery* as a precedent. The issue is thus *whether the information is valuable if known to competitors when security measures were taken*, and, the rule from *Mason*, according to the modelling in [9] shown in Table 1, is that it is, so we can conclude that the information is valuable in *MBL*.

Sometimes, an issue will appear at the level of factor ascription. Consider *Arco Industries Corp. v. Chemcast Corp*: all the base level factors favour the defendant and establish that the information was not a trade secret. We must therefore look for a factor with magnitude, in which the facts give a relatively low weight to the ascription. Here we find that DistinctProducts was assigned even though there was some resemblance. The issue here is *whether the products should be considered distinct where there is some resemblance*. In the decision, this was indeed the main point at issue, and it was decided that, despite some similarities, the defendant's product did not have an "indentation below the planar surface of the grommet which in turn lies below the peripheral sealing ridge", and this was enough to consider them distinct. A second issue raised in the case concerned disclosure to outsiders, and here it was found that the plaintiff had indeed disclosed to outsiders, even though the outsiders concerned were restricted to customers. Thus the issue *does disclosure to customers count as disclosure to outsiders* was also

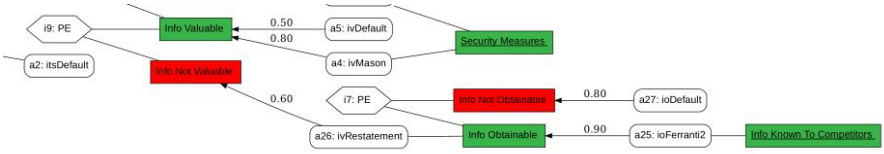


Figure 6. Argument map for the issue in MBL as modelled in [9] (see Table 1)

.

resolved in favour of the defendant. Once these factors have been described, there were no more points for the plaintiff to argue.

4.1. Refinement and Maintenance

It is rare that the ADP will be right first time, and so the development process will include running a set of test cases, and refining the ADP in the light of wrongly decided cases until all cases are explained correctly, or can be rejected as overruled. But even then new decisions must be monitored, so that the ADP and its realisation can be maintained as the case law evolves. The visual representation in argument maps greatly supports the refinement and maintenance of the theory recorded in the ADP, by identifying the nodes which need attention.

We began this section with an explanation for the outcome of *MBL* based on the ADP of [9], where the information was considered valuable, even though known to competitors, because security measures had been taken. This was a perfectly good explanation given the ADP in Table 1, but in fact *MBL* was decided for the *defendant*, and so the explanatory preference is incorrect and the ADP must be corrected.

Having identified the issue on which the case turned, whether the misappropriated information was valuable, we can examine the relevant nodes in the ADP. In this case they are the nodes shown in Table 1. It can be seen from the InfoValuable node that F6p, SecurityMeasures, is preferred to InfoObtainable, in virtue of the precedent *Mason*. But in *MBL*, InfoObtainable is accepted in virtue of F20d, InfoKnownToCompetitors, whereas in *Mason* it was accepted on the basis of F16d, InfoReverseEngineerable. As is clear from the second node in Table 1, F16d and F20d are given equal priority.

The solution is to recognise the lesser status of F16d by removing it from InfoObtainable: that the information could perhaps have been discovered by reverse engineering is clearly significantly weaker than it actually being known to competitors. Because it was the relative weakness of F16d that gave InfoObtainable its low priority in InfoValuable in Table 1, we remove F16d from InfoObtainable and can now increase the priority of InfoObtainable to reflect the decision in *MBL*. F16d remains with the original low priority, as derived from *Mason*. The revised nodes are shown in Table 2. Note that the modular design means that changes can be made to these nodes confident that other nodes will not be jeopardised.

In the case of *The Boeing Company v. Sierracin Corporation*, the defendant argued that the information had been disclosed to outsiders (see Figure 7). Here the issue is whether *if the information is disclosed to subcontractors, it is considered to have been disclosed to outsiders*. The current weights give the answer "no", but had *Boeing* been found for the defendant, this would need to be revised so that subcontractors were now included in outsiders when ascribing this factor.

Table 2. Fragment of ADP revised in the light of MBL

Node	Children	Acceptance Conditions	Justification
InfoValuable	F6p F8p F11d F15p F16d InfoObtainable	REJECT IF F11d ACCEPT IF F8p REJECT IF InfoObtainable ACCEPT IF F6p ACCEPT IF F15p REJECT IF F16d ACCEPT	Silfen Lewis MBL College Restatement Mason
InfoObtainable	F15p F20d F24d	REJECT IF F15p ACCEPT IF F24d OR F20d REJECT	College Ferranti

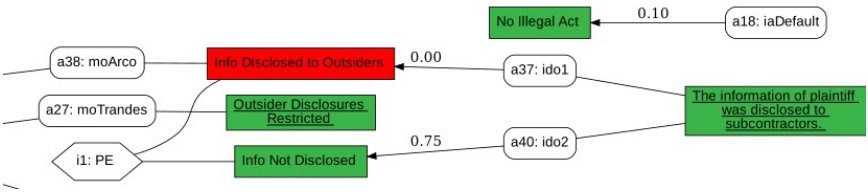


Figure 7. Fragment of Boeing showing disputed factor ascription

5. Discussion and Concluding Remarks

Angelic and Carneades are natural partners, since both conceptualise reasoning as moving from facts to conclusion through a series of arguments. Using Carneades to realise the acceptance conditions of an ADP encapsulating a theory of a legal domain is very straightforward: the node ID (or its negation) supplies the conclusion, the children in the body of the condition the premises, the weight indicates its priority and the source is also recorded. Factors with magnitude are also straightforwardly implemented through comparing weights as described in Section 3.2. This contrasts very favourably with previous implementations which required hand crafted code (e.g [19]). Moreover the argument schemes are held in a file separate from the program which executes them. This, given the simple correspondence between the schemes and the acceptance conditions, means that they can readily be edited to refine and maintain the theory by a knowledge engineer without any particular programming expertise.

A second major advantage of using Carneades to implement the ADP is that Caneades outputs an argument graph, directly supporting a visual representation rather than the rather cumbersome textual explanations produced in previous work such as [7] and [20]. Visual presentations of argument have been popular since the use of Toulmin's argument scheme, and make the reasoning far easier to follow than working through a sometimes lengthy set of textual conclusions. The colour coding in Carneades makes it especially easy to identify the main issues, the key points of contention in the case, essential for deploying IRAC explanations, and to locate problems in the theory.

Although our focus in this paper has been on visualising cases, we would also like to point out that the system can be run in batch mode, so that a large number of test cases can be run quickly, generating argument maps for all of them at once. This enables any

failing cases to be identified for visual examination and correction. Equally this could be helpful for to comparing the performance of competing theories expressed as ADPs.

Our theory has been validated by correctly deciding all the test cases taken from the available literature. The method is not limited to the common law or theories derived from cases, but can be used to implement legal theories derived from any source texts.

Taken together, the Angelic methodology and the Carneades argumentation system provide a means of producing a theory of a legal domain drawn from statutes, commentaries, experts and leading cases, which can readily be transformed into executable form to produce an evaluation and visual representation of the arguments in a particular case. As such it provides an excellent tool to support the analysis of a domain, refinement of the resulting theory and its maintenance as the law evolves. It could be used both to provide advice on particular cases, and to explain the domain to students and clients.

References

[1] Rissland EL, Ashley KD. A case-based system for Trade Secrets law. In: Proceedings of the 1st ICAIL; 1987. p. 60-6.
[2] Aleven V. Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment. Artificial Intelligence. 2003;150(1-2):183-237.
[3] Bench-Capon T, Atkinson K. Using Argumentation Schemes to Model Legal Reasoning. arXiv preprint arXiv:221000315: Presented at 4th European Conference on Argumentation. 2022.
[4] Skalak DB, Rissland EL. Arguments and cases: An inevitable intertwining. AI and Law. 1992;1(1):3-44.
[5] Prakken H. A formal analysis of some factor-and precedent-based accounts of precedential constraint. Artificial Intelligence and Law. 2021;29(4):559-85.
[6] Mumford J, Atkinson K, Bench-Capon T. Explaining Factor Ascription. In: Proceedings of JURIX 2021. IOS Press; 2021. p. 191-6.
[7] Al-Abdulkarim L, Atkinson K, Bench-Capon T. A methodology for designing systems to reason with legal cases using ADFs. Artificial Intelligence and Law. 2016;24(1):1-49.
[8] Al-Abdulkarim L, Atkinson K, Bench-Capon T, Whittle S, Williams R, Wolfenden C. Noise induced hearing loss: Building an application using the ANGELIC methodology. Argument and Computation. 2019;10(1):5-22.
[9] Bench-Capon T. Using Issues to Explain Legal Decisions. In: EXplainable and Responsible AI and Law 2021. vol. 3168. CEUR Workshop Proceedings; 2021. .
[10] Brewka G, Woltran S. Abstract Dialectical Frameworks. In: 12th International Conference on the Principles of Knowledge Representation and Reasoning; 2010. .
[11] Gordon TF, Prakken H, Walton D. The Carneades model of argument and burden of proof. Artificial Intelligence. 2007;171(10-15):875-96.
[12] Gordon TF. Visualizing Carneades Argument Graphs. Law, Probability and Risk. 2007;6(1-4):109-17.
[13] Gordon TF, Walton D. Formalizing Balancing Arguments. In: Proceedings of COMMA 2016; 2016. p. 327-38.
[14] Gordon TF. Defining argument weighing functions. IfCoLog Journal of Logics and their Applications. 2018;5(3):747-73.
[15] Frühwirth T. Constraint Handling Rules. Cambridge University Press; 2009.
[16] Gordon TF, Friederich H, Walton D. Representing Argumentation Schemes with Constraint Handling Rules (CHR). Argument and Computation. 2017;9(2):91-119.
[17] Horty JF. Reasoning with Dimensions and Magnitudes. In: Proceedings of the 16th ICAIL; 2017. p. 109-18.
[18] Bench-Capon T. Explaining legal decisions using IRAC. In: Computational Models of Natural Argument 2020. vol. 2669. CEUR Workshop Proceedings; 2020. p. 74-83.
[19] Bench-Capon T, Atkinson K. Implementing factors with magnitude. In: Computational Models of Argument. IOS Press; 2018. p. 449-50.
[20] Atkinson K, Collenette J, Bench-Capon T, Dzehtsiarou K. Practical tools from formal models: the ECHR as a case study. In: Proceedings of the 18th ICAIL; 2021. p. 170-4.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220445

Precedential Constraint Derived from Inconsistent Case Bases

Ilaria CANAVOTTO [a,1]
[a] *Department of Philosophy, University of Maryland, College Park, USA*

Abstract. I explore a factor-based model of precedential constraint that, unlike existing models, does not rely on the assumption that the background set of precedent cases is consistent. The model I consider is a generalization of the reason model of precedential constraint that was suggested by Horty. I show that, within this framework, inconsistent case bases behave in a sensible and interesting way, both from a logical and a more practical perspective.

Keywords. precedential constraint, reason model, factors, inconsistent case-base

1. Introduction

According to the doctrine of precedent, the decisions of earlier courts constrain the decisions of later courts through the requirement that later decisions ought to be consistent with precedent decisions. Explaining how, exactly, precedent cases constrain future decisions—or what, exactly, "consistency" means—is a traditional problem in legal theory but has become, through the development of the reason model of constraint by Horty and Bench-Capon [1,2], a central concern in AI and Law as well.

The reason model, which builds on Lamond's theory of precedential constraint [3], supplements a factor-based representation of legal cases in the style of HYPO [4] and CATO [5] with priority orderings between sets of factors representing the strength of the reasons underlying the decisions of different courts. With respect to earlier proposals based on similar ideas [6,7], the key innovation is that these priority orderings are used to define a notion of consistency, and so a notion of constraint.

This has led to a number of developments in AI and Law that aim at refining the analysis of constraint by tackling, e.g., factors that can have multiple values [8,9,10], framework precedents [11], or issues [12]. A problem that has not been taken up in this literature, however, is that the reason model notion of consistency presupposes that the background case base is consistent to start with, which is unrealistic. Horty [1] sketches a generalization of the reason model notion of constraint that applies to inconsistent case bases as well. Yet, the idea is only presented and not verified. My aim is to take Horty's suggestion and study how, exactly, according to the generalized notion of constraint, inconsistent case bases constrain future decisions.[2]

[1] E-mail: icanavot@umd.edu.

[2] A different approach to the problems presented by inconsistent case bases can be found in recent works by Peters and colleagues [13] and by van Woerkom and colleagues [14] where a version of the reason model is used to analyze how machine learning systems base their decisions on training data.

I proceed as follows. In Section 2, I review some basic definitions and present the generalized notion of constraint. In Section 3, I define what it means, in this framework, that it follows from a possibly inconsistent case base that a decision is obligatory or permitted and study the resulting logic of constraint. In Section 4, I address three more practical issues: first, whether it is feasible, in practice, to determine that a decision made in the context of an inconsistent case base is permitted; second, whether inconsistent case bases provide us with intuitive criteria to identify the fact situations that ought to be decided for a specific side; and, finally, whether inconsistent case bases provide us with intuitive criteria to compare different permissible decisions. Section 5 concludes.

2. Generalized reason model notion of precedential constraint

The reason model represents cases as consisting of three elements: a fact situation presented to the court; an outcome, which can be either a decision for the plaintiff or a decision for the defendant; and a rule that justifies the outcome on the basis of a reason that holds in the considered situation. I start by reviewing the definitions of these elements.

A *fact situation* is a set of facts that are legally relevant, called *factors*. Factors are assumed to have polarities: every factor favors either the plaintiff, denoted with π, or the defendant, denoted with δ. We take $\mathscr{F}^{\pi} = \{f_1^{\pi}, \ldots, f_n^{\pi}\}$ to be the set of factors favoring the plaintiff, $\mathscr{F}^{\delta} = \{f_1^{\delta}, \ldots, f_m^{\delta}\}$ to be the set of factors favoring the defendant, and $\mathscr{F} = \mathscr{F}^{\pi} \cup \mathscr{F}^{\delta}$ to be the set of all factors. Where s is one of the two sides, we will use $\bar{s}$ to represent the other, so $\bar{s} = \pi$ if $s = \delta$ and $\bar{s} = \delta$ if $s = \pi$. Where X is a fact situation, $X^s = X \cap \mathscr{F}^s$ is the set of factors from X that favor the side s.

Next, a *reason for the side s* is a set of factors uniformly favoring s; a *reason* is then a set of factors uniformly favoring *a* side. We say that a reason U *holds* in a fact situation X whenever $U \subseteq X$ and that U is *at least as strong as* another reason V favoring the same side as U whenever $V \subseteq U$. To illustrate, if $X_1 = \{f_1^{\pi}, f_2^{\pi}, f_1^{\delta}\}$, then $\{f_1^{\pi}\}$ and $\{f_1^{\pi}, f_2^{\pi}\}$ are reasons for π that hold in X_1 and such that $\{f_1^{\pi}, f_2^{\pi}\}$ is at least as strong as $\{f_1^{\pi}\}$.

We can now define a *rule* as a statement of the form $U \to s$, where U is a reason for the side s. Intuitively, $U \to s$ represents a defeasible rule that, roughly, says that, if U holds in a fact situation, then the court has a *pro tanto* reason to decide that situation for s. For any rule $r = U \to s$, we let $prem(r) = U$ and $conc(r) = s$. We say that r is *applicable in a fact situation* X whenever its premise holds in X, that is $prem(r) \subseteq X$.

Finally, a *case* is a triple of the form $\langle X, r, s\rangle$, where X is a fact situation, r is a rule applicable in X and whose conclusion is s, and s is either π or δ. For any case $c = \langle X, r, s\rangle$, we set $facts(c) = X$, $rule(c) = r$, and $out(c) = s$. A *case base* is simply a set of cases.

Turning to the notion of constraint, the reason model is based on two key ideas: first, that every case decided by a court induces a priority ordering among reasons and, second, that the decisions taken by later courts ought to be consistent with the priority ordering induced by precedent cases. We start by defining the priority ordering induced by a case:

Definition 1 (Priority ordering induced by a case)**.** Where $c = \langle X, r, s\rangle$ is a case, the priority ordering $<_c$ induced by c is defined by setting, for any pair of reasons $U \subseteq \mathscr{F}^{\bar{s}}$ and $V \subseteq \mathscr{F}^s$: $U <_c V$ if and only if $U \subseteq X$ and $prem(r) \subseteq V$.

To illustrate, let c_1 be the case $\langle X_1, r_1, \pi\rangle$, where X_1 is as above and $r_1 = \{f_1^{\pi}\} \to \pi$. The idea behind Definition 1 is that c_1 reveals that, according to the court, the reason

$\{f_1^\pi\}$ has higher priority than every reason for δ that holds in X_1—i.e., $\{f_1^\delta\}$—and that every reason for π that is at least as strong as $\{f_1^\pi\}$, for instance $\{f_1^\pi, f_2^\pi\}$, also has higher priority than every such reason. It is worth noting that Definition 1 ensures that the ordering $<_c$ is asymmetric: there are no reasons U and V such that $U <_c V$ and $V <_c U$.

We can now define the priority ordering induced by a case base as follows:

Definition 2 (Priority ordering induced by a case base)**.** Where Γ is a case base, the priority ordering $<_\Gamma$ induced by Γ is defined by setting, for any pair of reasons U and V: $U <_\Gamma V$ if and only if there is a case c in Γ such that $U <_c V$.

Definition 2 does not force $<_\Gamma$ to be asymmetric: there may be reasons U and V such that $U <_\Gamma V$ and $V <_\Gamma U$. This happens when some cases in Γ support conflicting information about the priority ordering among reasons. Such cases make Γ inconsistent. To make this precise, call a pair of reasons such that $U <_\Gamma V$ and $V <_\Gamma U$ (abbreviated as $U \perp_\Gamma V$) an *inconsistency in* Γ and let $inc(\Gamma)$ be the set of inconsistencies in Γ. We can then define:

Definition 3 (Inconsistent and consistent case base)**.** A case base Γ is *inconsistent* when $inc(\Gamma) \neq \emptyset$ and *consistent* when $inc(\Gamma) = \emptyset$.

So, if c_1 is as before and c_2 is the case $\langle X_2, r_2, \delta\rangle$, where $X_2 = \{f_1^\pi, f_1^\delta, f_2^\delta\}$ and $r_2 = \{f_1^\delta\} \rightarrow \delta$, then $\Gamma_1 = \{c_1, c_2\}$ is inconsistent, as $\{f_1^\delta\} \perp_{\Gamma_1} \{f_1^\pi\}$, and so $inc(\Gamma_1) \neq \emptyset$.

Now, the case base Γ_1 in our example is inconsistent in a way that is so obvious that it would be striking if any court actually had to work with a case base like it. But, in real life, case bases are much more complex than Γ_1 and it is not at all unusual that some precedents pull in different directions. The question *How do inconsistent case bases constrain?* thus becomes pressing. The reason model notion of constraint does not allow us to pose—let alone answer—this question. According to the reason model, decisions of later courts ought to preserve consistency of the underlying case base. Formally:

Definition 4 (Reason model notion of constraint)**.** Let Γ be a consistent case base. Then, against the background of Γ, the court is permitted to decide the fact situation X for the side s on the basis of the rule $r = U \rightarrow s$ just in case $inc(\Gamma \cup \{\langle X, r, s\rangle\}) = \emptyset$.

The problem is that Definition 4 explicitly requires that the underlying case base be consistent. Even worse, simply dropping this requirement would not give us a sensible account of how inconsistent case bases constrain—given an inconsistent case base, there would be no permitted way to decide any fact situation, which is absurd.

There is, however, another way to generalize the reason model notion of constraint so that it applies to inconsistent case bases as well. The idea, which was suggested in [1, p.15], is that, rather than being required to preserve consistency of a consistent case base, courts should be required to introduce no new inconsistencies into a possibly inconsistent case base. Let us take this suggestion and define:

Definition 5 (Generalized reason model notion of constraint)**.** Against the background of a case base Γ, the court is permitted to decide the fact situation X for the side s on the basis of the rule $r = U \rightarrow s$ just in case $inc(\Gamma \cup \{\langle X, r, s\rangle\}) \subseteq inc(\Gamma)$.

To make Definition 5 less abstract, suppose that a court has to decide the situation $X_3 = \{f_1^\pi, f_2^\pi, f_1^\delta, f_2^\delta\}$ against the background of the inconsistent case base Γ_1. Accord-

ing to Definition 5, the court is *not* allowed to extend Γ_1 to the case base $\Gamma_2 = \Gamma_1 \cup \{c_3\}$, where $c_3 = \langle X_3, \{f_1^\delta\} \rightarrow \delta, \delta \rangle$. In fact, c_3 induces the priority $\{f_1^\pi, f_2^\pi\} <_{c_3} \{f_1^\delta\}$, which is inconsistent with the priority $\{f_1^\delta\} <_{c_1} \{f_1^\pi, f_2^\pi\}$. In addition, it is neither the case that $\{f_1^\pi, f_2^\pi\} <_{c_1} \{f_1^\delta\}$ (as $\{f_1^\delta\} <_{c_1} \{f_1^\pi, f_2^\pi\}$ and $<_{c_1}$ is asymmetric) nor that $\{f_1^\pi, f_2^\pi\} <_{c_2} \{f_1^\delta\}$ (as $\{f_1^\pi, f_2^\pi\}$ does not hold in X_2). So, we have $\{f_1^\pi, f_2^\pi\} \perp_{\Gamma_2} \{f_1^\delta\}$ but not $\{f_1^\pi, f_2^\pi\} \perp_{\Gamma_1} \{f_1^\delta\}$—that is, deciding for δ on the basis of $\{f_1^\delta\} \rightarrow \delta$ introduces a new inconsistency. Still, this does not mean that it is not permissible to decide X_3 for δ at all; for instance, deciding X_3 for δ on the basis of the rule $\{f_1^\delta, f_2^\delta\} \rightarrow \delta$ would not introduce any new inconsistencies and is thus allowed.[3]

To conclude this section, Observation 1 shows that Definition 5 is indeed a generalization of the reason model notion of constraint: it is permissible—in the sense of Definition 5—to extend a *consistent* case base just in case the extension is also consistent.

Observation 1. *Let Γ be a consistent case base and c any case. Then, $inc(\Gamma \cup \{c\}) \subseteq inc(\Gamma)$ if and only if $inc(\Gamma \cup \{c\}) = \emptyset$.*

3. Conflict-free deontic logic from inconsistent case bases

Observation 1 supports the idea that the notion of constraint set out in Definition 5 is a natural generalization of the reason model notion of constraint. But how exactly does the generalized notion work when the background case base is inconsistent? In this section, I begin to explore this question from a logical perspective: assuming the generalized notion, I aim to define, first, what it means that it follows from a possibly inconsistent case base that a decision for a side is obligatory or permitted and, second, to study the logic of constraint underlying the resulting notions of permission and obligation.

As to the first task, following [15, Sect. 1.2.4], I begin from the observation that Definition 5 characterizes the rules that a court is permitted to use in the context of a certain case base to justify its decisions. Given this notion, we can say that *it follows from a case base that a decision for a side is permitted* whenever there is a rule that is permitted in the context of that case base that supports that side and that *it follows from a case base that a decision is obligatory* whenever all the rules that are permitted in the context of that case base support that side. To state this formally, let $Perm(\Gamma, X)$ be the set of rules that are applicable to the fact situation X and are permitted in the context of the case base Γ. In addition, let $\Gamma \mid\sim P_X(s)$ mean that it follows from Γ that deciding X for s is permitted and $\Gamma \mid\sim O_X(s)$ mean that it follows from Γ that deciding X for s is obligatory. Then, $\Gamma \mid\sim P_X(s)$ and $\Gamma \mid\sim O_X(s)$ are defined as follows:

Definition 6 (Deontic operators)**.** Let Γ be a case base and X a fact situation. Then, $\Gamma \mid\sim P_X(s)$ holds just in case there is an $r \in Perm(\Gamma, X)$ such that $conc(r) = s$, and $\Gamma \mid\sim O_X(s)$ holds just in case, for all $r \in Perm(\Gamma, X)$, $conc(r) = s$.

Turning to the second task, it immediately follows from our definitions that a court ought to decide for a side if and only if it is not allowed to decide for the opposite side:

Observation 2. *$\Gamma \mid\sim O_X(s)$ holds if and only if $\Gamma \mid\sim P_X(\bar{s})$ does not hold.*

[3] Of course, as the case for δ and the case for π are symmetric, an analogous reasoning shows that the court is allowed to decide X_3 for π on the basis of the rule $\{f_1^\pi, f_2^\pi\} \rightarrow \pi$ but not on the basis of the rule $\{f_1^\pi\} \rightarrow \pi$.

Observation 2 tells us that the deontic operators introduced above are interdefinable in the usual way. A key question is whether they are standard also in the sense that they exclude the possibility of contradictory obligations: Can we exclude that, in the context of an inconsistent case base, a court is required to decide for s and also required to decide for $\bar{s}$? The question is not trivial because Definition 6 mirrors the semantics of standard deontic logics and, in standard deontic logics, inconsistent normative information does give rise to contradictory requirements. Now, in our case, the only situation in which both $\Gamma \mathrel{|\!\sim} O_X(s)$ and $\Gamma \mathrel{|\!\sim} O_X(\bar{s})$ could hold is when the set $Perm(\Gamma, X)$ is empty—that is, when no rule applicable in X is permitted in the context of Γ. It turns out that this situation can be excluded: no matter whether Γ is inconsistent or which factors are present in X, there is a permitted rule that can be used to decide X.

Observation 3. *Let Γ be a case base and X a fact situation. Then there exists some rule $r : U \to s$ applicable in X such that $inc(\Gamma \cup \{\langle X, r, s\rangle\}) \subseteq inc(\Gamma)$.*

Proof. If Γ is consistent, then, by Observation 1, Observation 3 is equivalent to the claim that, for any fact situation X, there is a rule $r : U \to s$ applicable in X such that $\Gamma \cup \{\langle X, r, s\rangle\}$ is consistent. A proof of the latter claim can be found in [15, App. A.2]. So, let Γ be inconsistent. Suppose, toward contradiction, that there is no rule $r : U \to s$ applicable in X such that $inc(\Gamma \cup \{\langle X, r, s\rangle\}) \subseteq inc(\Gamma)$. Then, letting $c_1 = \langle X, X^s \to s, s\rangle$ and $\Gamma_1 = \Gamma \cup \{c_1\}$, it follows that there is a pair of reasons $U \subseteq \mathscr{F}^{\bar{s}}$ and $V \subseteq \mathscr{F}^{s}$ such that $U \perp_{\Gamma_1} V$ but $not(U \perp_{\Gamma} V)$. By unfolding the definitions, it is not difficult to see that this can only happen when the following facts hold: (1) $U <_{c_1} V$, i.e., $U \subseteq X^{\bar{s}}$ and $X^s \subseteq V$; (2) $V <_{\Gamma} U$, i.e., there is a case $c_3 = \langle X_3, r_3, \bar{s}\rangle$ in Γ s.t. $V \subseteq X_3$ and $prem(r_3) \subseteq U$; and (3) $U \not<_{\Gamma} V$. Similarly, letting $c_2 = \langle X, X^{\bar{s}} \to \bar{s}, \bar{s}\rangle$ and $\Gamma_2 = \Gamma \cup \{c_2\}$, it follows from our hypothesis that there is a pair of reasons $U' \subseteq \mathscr{F}^{\bar{s}}$ and $V' \subseteq \mathscr{F}^{s}$ such that $U' \perp_{\Gamma_2} V'$ but $not(U' \perp_{\Gamma} V')$. Again, this can only happen when the following facts hold: (4) $V' <_{c_2} U'$, i.e., $V' \subseteq X^s$ and $X^{\bar{s}} \subseteq U'$; (5) $U' <_{\Gamma} V'$, i.e., there is a case $c_4 = \langle X_4, r_4, s\rangle$ in Γ s.t. $U' \subseteq X_4$ and $prem(r_4) \subseteq V'$; and (6) $V' \not<_{\Gamma} U'$. It is now easy to see that: (7) $U \subseteq X_4$, since $U \subseteq X^{\bar{s}}$ by 1, $X^{\bar{s}} \subseteq U'$ by 4, and $U' \subseteq X_4$ by 5; (8) $prem(r_4) \subseteq V$, since $prem(r_4) \subseteq V'$ by 5, $V' \subseteq X^s$ by 4, and $X^s \subseteq V$ by 1. But 7 and 8 entail that $U <_{c_4} V$, and so that $U <_{\Gamma} V$, which contradicts 3. □

An immediate consequence of Observation 3 is that, regardless of whether the background case base is inconsistent, the court will never be subject to contradictory requirements; in addition, in any situation, the court will be either required to decide for a side, or required to decide for the opposite side, or permitted to decide for either side:

Observation 4. *It is never the case that both $\Gamma \mathrel{|\!\sim} O_X(s)$ and $\Gamma \mathrel{|\!\sim} O_X(\bar{s})$ hold. In addition, it is always the case that exactly one of the following holds: either $\Gamma \mathrel{|\!\sim} O_X(s)$, or $\Gamma \mathrel{|\!\sim} O_X(\bar{s})$, or $\Gamma \mathrel{|\!\sim} P_X(s)$ and $\Gamma \mathrel{|\!\sim} P_X(\bar{s})$.*

We can thus conclude that possibly inconsistent case bases support a natural, *conflict-free* deontic logic. Let me now move on to three more practical issues.

4. Applying the generalized notion of constraint

4.1. Feasibility

Suppose that we have to decide a fact situation X against the background of a possibly inconsistent case base Γ and that we want to determine whether we are allowed to decide X for the side s on the basis of the rule $r = U \to s$. How should we proceed?

Simply applying Definition 5 will often be unfeasible. If we do that, we will have to consider all pairs of opposing reasons that can be built from the sets of factors $\mathscr{F}^s$ and $\mathscr{F}^{\bar{s}}$ and check, for each pair, that either it does not form an inconsistency in the extended case base $\Gamma' = \Gamma \cup \{\langle X, r, s\rangle\}$ or, if it does, that it also forms an inconsistency in Γ. The problem is that the number of pairs of opposing reasons grows exponentially with the number of factors: if $\mathscr{F}^s$ contains m factors and $\mathscr{F}^{\bar{s}}$ contains n factors, then there are 2^m reasons for s and 2^n reasons for $\bar{s}$, which results in $2^m \times 2^n = 2^{m+n}$ pairs of opposing reasons. Unless we work with only a few basic factors—which may not be possible if we want to model real cases—simply applying Definition 5 will thus not do.

Now, if our task were just to check that Γ' is consistent, a simplification would be available: As shown in [15, Sect. 2.2.1], it turns out that a case base is inconsistent just in case it includes two cases $c_i = \langle X_i, r_i, s\rangle$ and $c_j = \langle X_j, r_j, \bar{s}\rangle$ such that $prem(r_j) <_{c_i} prem(r_i)$ and $prem(r_i) <_{c_j} prem(r_j)$. This means that, if Γ' includes p cases decided for s and q cases decided for $\bar{s}$, we would have to check $p \times q$ pairs of case rule premises favoring opposite sides, which would make our problem more tractable—the search space would be polynomial in the number of cases rather than exponential in the number of factors. The central result of this section is that, fortunately, a similar simplification is available for our original task:

Observation 5. *Let Γ be a case base, c be the case $\langle X, r, s\rangle$, and $\Gamma' = \Gamma \cup \{c\}$. Then, $inc(\Gamma') \not\subseteq inc(\Gamma)$ if and only if the following two conditions obtain:*
1. there is a case $c_i = \langle X_i, r_i, \bar{s}\rangle$ in Γ s.t. $prem(r_i) <_c prem(r)$ and $prem(r) <_{c_i} prem(r_i)$;
2. there is no case $c_j = \langle X_j, r_j, s\rangle$ in Γ s.t. $X^{\bar{s}} \subseteq X_j^{\bar{s}}$ and $prem(r_j) \subseteq prem(r)$.

Proof. ***Left-to-right.*** Assume that there are reasons $U \subseteq \mathscr{F}^{\bar{s}}$ and $V \subseteq \mathscr{F}^s$ such that $U \perp_{\Gamma'} V$ but $not(U \perp_\Gamma V)$. By unfolding the definitions, it is easy to see that this can only happen when: (1) $U <_c V$, i.e., $U \subseteq X^{\bar{s}}$ and $prem(r) \subseteq V$, (2) $V <_\Gamma U$, i.e., there is $c_i = \langle X_i, r_i, \bar{s}\rangle$ in Γ s.t. $V \subseteq X_i$ and $prem(r_i) \subseteq U$; and (3) $U \not<_\Gamma V$. But then: (4) $prem(r_i) <_c prem(r)$, since $prem(r_i) \subseteq X$ by 2 and 1; and (5) $prem(r) <_{c_i} prem(r_i)$, since $prem(r) \subseteq X_i$ by 1 and 2. Facts 4 and 5 suffice to establish condition 1 in Observation 5. In order to establish condition 2, suppose, toward contradiction, that there is $c_j = \langle X_j, r_j, s\rangle$ in Γ such that (a) $X^{\bar{s}} \subseteq X_j^{\bar{s}}$ and (b) $prem(r_j) \subseteq prem(r)$. Then: (6) $U \subseteq X_j$, as $U \subseteq X^{\bar{s}}$ by 1, $X^{\bar{s}} \subseteq X_j^{\bar{s}}$ by (a), and $X_j^{\bar{s}} \subseteq X_j$ by definition; and (7) $prem(r_j) \subseteq V$, as $prem(r_j) \subseteq prem(r)$ by (b) and $prem(r) \subseteq V$ by 1. It follows from 6 and 7 that $U <_{c_j} V$, and so that $U <_\Gamma V$, which contradicts 3. ***Right-to-left.*** Suppose that Γ satisfies conditions 1 and 2 in Observation 5. So, let $c_i = \langle X_i, r_i, \bar{s}\rangle$ be a case in Γ satisfying condition 1. Then: (1) $prem(r) \subseteq X_i$, since $prem(r) <_{c_i} prem(r_i)$; and (2) $prem(r_i) \subseteq X^{\bar{s}}$, since $prem(r_i) <_c prem(r)$. Hence, $prem(r) <_{c_i} X^{\bar{s}}$. Since $X^{\bar{s}} <_c prem(r)$ by Definition 1, we have that $prem(r) \perp_{\Gamma'} X^{\bar{s}}$. It remains to be shown that $not(prem(r) \perp_\Gamma X^{\bar{s}})$, i.e., that there is no $c_j = \langle X_j, r_j, s\rangle$ in Γ such that $X^{\bar{s}} <_{c_j} prem(r)$. Suppose, toward contradiction, that there is such a c_j. Then, by Definition 1, $X^{\bar{s}} \subseteq X_j^{\bar{s}}$ and $prem(r_j) \subseteq prem(r)$. But this contradicts the hypothesis that Γ satisfies condition 2. □

According to Observation 5, we can determine whether it is permissible to extend Γ with the case $\langle X, r, s\rangle$ by applying the following procedure: first, take all pairs of opposing reasons consisting of $prem(r)$ and of a case rule premise from Γ that favors $\bar{s}$; if no pair is an inconsistency in $\Gamma \cup \{\langle X, r, s\rangle\}$, it is permissible to extend Γ with $\langle X, r, s\rangle$; otherwise, take, for each case c_j in Γ decided for s, two additional pairs of reasons, i.e., the pair consisting of $X^{\bar{s}}$ and $facts(c_j)^{\bar{s}}$ and the pair consisting of $prem(rule(c_j))$ and $prem(r)$; if there is a case for s for which the two pairs satisfy condition 2, then it is permissible to extend Γ with $\langle X, r, s\rangle$; otherwise, it is not. This means that, if Γ includes p cases decided

for $\bar{s}$ and q cases decided for s, then, in the worst case scenario, we need to consider $p + (2 \times q)$ pairs of opposing reasons. So, even when Γ is inconsistent, our search space is polynomial in the number of cases rather than exponential in the number of factors.

4.2. Identifying fact situations that ought to be decided for a side

Suppose now that we have to decide a fact situation X in the context of a possibly inconsistent case base Γ and that we want to determine whether we ought to decide X for a specific side. Are there intuitive relations between X and the fact situations already decided in Γ that would help us do that? Observation 5 is key to answer this question. To see this, I define two notions that will make its content more transparent, namely the notions of a defeater and of a supporter of a potential decision.

The idea is simple: when, given a case base Γ, we want to determine whether we can decide a fact situation X for the side s on the basis of the rule r, we consider the potential decision $c = \langle X, r, s \rangle$ and see if it is defeated or supported by some cases in Γ. A *defeater of c in Γ* is any case in Γ that is decided for $\bar{s}$ on the basis of a reason that is inconsistent with the reason presented in c. The set of deafeaters of c in Γ is thus:

$$\begin{aligned} \mathit{defeaters}_\Gamma(c) &= \{c_i = \langle X_i, r_i, \bar{s} \rangle \in \Gamma \mid \mathit{prem}(r) <_{c_i} \mathit{prem}(r_i) \text{ and } \mathit{prem}(r_i) <_c \mathit{prem}(r)\} \\ &= \{c_i = \langle X_i, r_i, \bar{s} \rangle \in \Gamma \mid \mathit{prem}(r) \subseteq X_i \text{ and } \mathit{prem}(r_i) \subseteq X\} \end{aligned}$$

A *supporter of c in Γ* is any case in Γ that is decided for s and such that c is at least as strong for s as that case, where c is at least as strong for s as a case $c_j = \langle X_j, r_j, s \rangle$ just in case, first, the reason for s presented in c is at least strong as the reason for s presented in c_j and, second, the strongest reason for $\bar{s}$ that holds in X is weaker than the strongest reason for $\bar{s}$ that holds in X_j.[4] The set of supporters of c in Γ is thus:

$$\mathit{supporters}_\Gamma(c) = \{c_j = \langle X_j, r_j, s \rangle \in \Gamma \mid \mathit{prem}(r_j) \subseteq \mathit{prem}(r) \text{ and } X^{\bar{s}} \subseteq X_j^{\bar{s}}\}$$

With these notions in place, Observation 5 simply says that, according to the generalized notion of constraint, a court is *not allowed*, in the context of Γ, to decide X for s on the basis of r just in case the potential decision $c = \langle X, r, s \rangle$ has at least one defeater and no supporter in Γ:

Observation 5′. Let Γ be a case base and c be the case $\langle X, r, s \rangle$. Then, $\mathit{inc}(\Gamma \cup \{c\}) \not\subseteq \mathit{inc}(\Gamma)$ if and only if $\mathit{defeaters}_\Gamma(c) \neq \emptyset$ and $\mathit{supporters}_\Gamma(c) = \emptyset$.

Now, according to Observation 2, a court ought to decide for s just in case it is not allowed to decide for $\bar{s}$—i.e., no applicable rule supporting $\bar{s}$ is permitted. Observation 5′ then says that X ought to be decided for s just in case it has two features:

Feature 1. Every potential decision of the form $\langle X, r, \bar{s} \rangle$ has a defeater in Γ.

This happens when (and only when) every reason for $\bar{s}$ holding in X also holds in a situation already decided for s on the basis of a reason that itself holds in X. This, in turn, happens when (and only when) Γ includes a case whose reason contradicts the strongest reason for $\bar{s}$ holding in X—i.e., when $\langle X, X^{\bar{s}} \rightarrow \bar{s}, \bar{s} \rangle$ has a defeater in Γ.

Feature 2. No potential decision of the form $\langle X, r, \bar{s} \rangle$ has a supporter in Γ.

[4] The notion of strength between cases proposed here is a generalization of the notion of an *a fortiori* case discussed in [16] and formalized in [1].

This happens when (and only when) every fact situation already decided for $\overline{s}$ *either* does not include X^s *or* is decided on the basis of a reason that does not hold in X.

To get a better sense of the two features and their intuitiveness, let us consider two cases in which an individual, who moved to the foreign country Z, files a request to change fiscal domicile with respect to income tax.[5] The defendant is the individual and the plaintiff her home country. The relevant factors are: δ spent only one month in country Z (f_1^π); δ owned a house in her home country (f_2^π); δ had a permanent job in country Z (f_1^δ); δ opened a bank account in country Z (f_2^δ); δ registered a car in country Z (f_3^δ).

Example 1. Suppose that Mr. C presents the situation $X_7 = \{f_1^\pi, f_2^\pi, f_1^\delta\}$ and that there are two mutually inconsistent precedents: the case of Mr. A is $c_5 = \langle X_5, r_5, \pi\rangle$, where $X_5 = \{f_1^\pi, f_1^\delta\}$ and $r_5 = \{f_1^\pi\} \to \pi$, and the case of Mrs. B is $c_6 = \langle X_6, r_6, \delta\rangle$, where $X_6 = \{f_1^\pi, f_1^\delta, f_2^\delta\}$ and $r_6 = \{f_1^\delta\} \to \delta$. Are we allowed to find for Mr. C? The answer is negative. We can see this formally as follows: The strongest reason for δ that holds in X_7 (i.e., $\{f_1^\delta\}$) also holds in X_5; in turn, the reason that decides X_5 (i.e., $\{f_1^\pi\}$) holds in X_7; so, c_5 defeats all potential decisions for δ (i.e., $\langle X_7, r_6, \delta\rangle$), as per Feature 1. What is more, f_2^π is a pro-π factor present in X_7 but not in X_6; so, no potential decision for δ is supported by c_6, as per Feature 2. Thus, we ought to decide for π. This formal argument has an intuitive informal counterpart: The only reason in favor of Mr. C is that he had a permanent job in country Z. But the case of Mr. A already established that this reason is not strong enough to find for Mr. C, given that he spent only one month in country Z. It is true that Mrs. B won because she, as Mr. C, had a permanent job in country Z and despite the fact that she, as Mr. C, spent only one month in country Z. Yet, unlike Mrs. B, Mr. C owned a house in his home country. Thus, the case for Mr. C is weaker than the case for Mrs. B, while the case for his home country is stronger than the case for Mr. A's home country. So, we ought to decide for Mr. C's home country.

Example 2. Instead of c_5 and c_6, suppose that the cases of Mr. A and Mrs. B are, respectively, $c_8 = \langle X_8, r_5, \pi\rangle$, where $X_8 = \{f_1^\pi, f_1^\delta, f_2^\delta, f_3^\delta\}$ and $r_5 = \{f_1^\pi\} \to \pi$, and $c_9 = \langle X_9, r_6, \delta\rangle$, where $X_9 = \{f_1^\pi, f_2^\pi, f_1^\delta, f_2^\delta\}$ and $r_6 = \{f_1^\delta\} \to \delta$. The case base $\{c_8, c_9\}$ is inconsistent, but Mr. C presents the situation $X_{10} = \{f_1^\pi, f_2^\pi, f_2^\delta, f_3^\delta\}$. As before, we are not allowed to find for Mr. C. Formally: The strongest reason for δ that holds in X_{10} (i.e., $\{f_2^\delta, f_3^\delta\}$) also holds in X_8 and, in turn, the reason that decides X_8 (i.e. $\{f_1^\pi\}$), holds in X_{10}; so, c_8 defeats all potential decisions for δ, as per Feature 1. What is more, the reason that decides c_9 (i.e., $\{f_1^\delta\}$) does not hold in X_{10}; so, c_9 does not support any potential decision for δ, as per Feature 2. Thus, we ought to decide for π. Again, this formal argument has an intuitive informal counterpart: The case of Mr. A already established that no reason in favor of Mr. C is strong enough to find for him, given that he spent only one month in country Z. It is true that Mrs. B won even if she, as Mr. C, spent only one month in country Z; but, unlike Mr. C, Mrs. B had a permanent job in country Z—and this is why she won. Since this reason does not apply to Mr. C, his case cannot be compared to Mrs. B's case. In contrast, the case for Mr. C's home country is stronger than the case for Mr. A's home country. We then ought to decide for Mr. C's home country.

Interestingly, Example 1 and Example 2 suggest that there is nothing mysterious in the fact that we can derive consistent obligations from conflicting precedent cases; the

[5] The examples are inspired by some hypothetical cases introduced by Prakken and Sartor [7]

key is that two conflicting precedents can have different reach—one precedent supports all possible decisions for a side and the other no possible decision for the opposite side.

4.3. Comparing different permissible decisions

Example 1 and Example 2 concern situations that cannot be decided without contradicting a precedent. Yet, in the examples, the only potential decisions that, despite contradicting a precedent, are supported by some other precedents are decisions for π, which makes it clear that π should win. What about situations that cannot be decided without contradicting some precedents but that we are permitted to decide for either side? Are there situations of this sort? If so, are there criteria to guide our decisions in such cases?

The answer to the first question is that there are, indeed, situations of the kind just described. Suppose, for example, that the case base Γ_3 includes the case $c_{11} = \langle X_{11}, r_{11}, \pi \rangle$, where $X_{11} = \{f_1^\pi, f_1^\delta, f_2^\delta\}$ and $r_{11} = \{f_1^\pi\} \rightarrow \pi$, $c_{12} = \langle X_{12}, r_{12}, \pi \rangle$, where $X_{12} = \{f_1^\pi, f_2^\pi, f_1^\delta, f_2^\delta\}$ and $r_{12} = \{f_2^\pi\} \rightarrow \pi$, and $c_{13} = \langle X_{13}, r_{13}, \delta \rangle$, where $X_{13} = \{f_1^\pi, f_2^\pi, f_3^\pi, f_1^\delta, f_2^\delta, f_3^\delta\}$ and $r_{13} = \{f_2^\delta\} \rightarrow \delta$. Our task is to decide the situation $X_{14} = \{f_1^\pi, f_2^\pi, f_2^\delta\}$ in the context of the inconsistent Γ_3. It is easy to see that all the potential decisions available to us, i.e., $d_1 = \langle X_{14}, \{f_1^\pi\} \rightarrow \pi, \pi \rangle$, $d_2 = \langle X_{14}, \{f_2^\pi\} \rightarrow \pi, \pi \rangle$, $d_3 = \langle X_{14}, \{f_1^\pi, f_2^\pi\} \rightarrow \pi, \pi \rangle$, $d_4 = \langle X_{14}, \{f_2^\delta\} \rightarrow \delta, \delta \rangle$, contradict some precedent decisions but are supported by some other precedents. What should we do?

One answer might be that, since they all have some defeaters and some supporters, the four decisions are equally good; we can thus decide X_{14} however we like. But, if we look more carefully at their defeaters and supporters, it is not so obvious that d_1 to d_4 are really on a par: On the one hand, the pro-π cases d_1, d_2, and d_3 are defeated by only one case (i.e., c_{13}) and supported by at least one case (d_1 by c_{11}, d_2 by c_{12}, and d_3 by c_{11} and c_{12}). On the other hand, the pro-δ case d_4 is defeated by two cases (i.e., c_{11} and c_{12}), while it is supported by only one case (i.e., c_{13}). Since each of d_1, d_2, and d_3 has fewer defeaters than d_4 but at least as many supporters as d_4, it is reasonable to conclude that the former cases are better than the latter. This gives us an argument to decide X_{14} for π.

Generalizing from our example, the number of defeaters and supporters of competing potential decisions seems to provide us with an intuitive, and sensible, criterion to guide our decisions in fact situations like X_{14}. To make this precise, suppose that Γ is an inconsistent case base and X a situation that cannot be decided without contradicting some cases in Γ but that the court is allowed to decide for either side. Where $d = \langle X, r, s \rangle$ and $d' = \langle X, r', \bar{s} \rangle$ are two competing potential decisions, we can say that *d is better than d'* when d has fewer defeaters in Γ than d' and at least as many supporters in Γ as d', while d' is better than d when the opposite is true—in the other cases, the number of defeaters and supporters of d and d' does not support a preference for either case.

5. Conclusion

I investigated a generalization of the reason model notion of precedential constraint that can be used to address the question *How does an inconsistent set of precedents constrain?* The generalized notion provides an interesting, and sensible answer: inconsistent case bases support a natural, conflict-free deontic logic; they do not make verification that a decision is permissible substantially more complex than consistent case bases; finally,

they provide us with intuitive criteria both to identify the fact situations that ought to be decided for a specific side and to compare different permissible decisions.

Concerning permissible decisions, I left a number of questions unanswered that are worth investigating. First, I suggested how to compare pairs of *potential decisions* for opposite sides, but I have not considered how to lift the comparison to a comparison between opposite *outcomes*. The example in Section 4.3 is simple in this respect because all potential decisions for π are better than the only potential decision for δ, which clearly makes π a better outcome than δ. Can we say something about cases in which some potential decisions for π are better than some potential decisions for δ and vice versa? Another issue concerns the criteria to evaluate potential decisions *for the same side*. Going back, once again, to the example in Section 4.3, once we have established that it is better to decide X_{14} for π, would it be best to extend Γ_3 with d_1, d_2, or d_3? Does the number of supporters and defeaters of the three cases help us answer this question? Finally, I said nothing about fact situations that may be permissibly decided by contradicting either some or no precedents. According to the criteria suggested in Section 4.3, contradicting no precedents is better than contradicting some. It would be interesting to consider arguments in favor and against this consequence.

References

[1] Horty J. Rules and reasons in the theory of precedent. Legal Theory. 2011;17:1–33.

[2] Horty J, Bench-Capon T. A factor-based definition of precedential constraint. Artificial Intelligence and Law. 2012;20:181–214.

[3] Lamond G. Do precedents create rules? Legal Theory. 2005;11:1–26.

[4] Ashley K. Modeling Legal Argument: Reasoning with Cases and Hypotheticals. The MIT Press; 1990.

[5] Aleven V. Teaching Case-Based Argumentation Through a Model and Examples. PhD Thesis, Intelligent Systems Program, University of Pittsburgh; 1997.

[6] Bench-Capon T. Some observations on modelling case based reasoning with formal argument models. In: Proceedings of the Sixth International Conference on Artificial Intelligence and Law (ICAIL-99). The Association for Computing Machinery Press; 1999. p. 36–42.

[7] Prakken H, Sartor G. Modelling reasoning with precedents in a formal dialogue game. Artificial Intelligence and Law. 1998;6:231–287.

[8] Horty J. Reasoning with dimensions and magnitudes. Artificial Intelligence and Law. 2019;27:307–345.

[9] Prakken H. A formal analysis of some factor- and precedent-based accounts of precedential constraint. Artificial Intelligence and Law. 2021;29:559–585.

[10] Rigoni A. Representing dimensions within the reason model of precedent. Artificial Intelligence and Law. 2018;26:1–22.

[11] Rigoni A. An improved factor based approach to precedential constraint. Artificial Intelligence and Law. 2015;23:133–160.

[12] Bench-Capon T, Atkinson K. Precedential constraint: The role of issues. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law (ICAIL 2021). The Association for Computing Machinery Press; 2021. p. 12–21.

[13] Peters J, Prakken H, Bex F. Justification derived from inconsistent case bases using authoritativeness. In: Proceedings of the First International Workshop on Argumentation for eXplainable AI (ArgXAI). vol. 3209. CEUR Workshop Proceedings; 2022. .

[14] van Woerkom W, Grossi D, Prakken H, Verheij B. Landmarks in Case-based Reasoning: From Theory to Data. In: Proceedings of the First International Confefence on Hybrid Human-Machine Intelligence. IOS Press; 2022. .

[15] Horty J. *The Logic of Precedent: Constraint and Freedom in Common Law Reasoning*; 20xx. Forthcoming with Cambridge University Press. Available at http://www.horty.umiacs.io/articles/2022-7-15-logic-precedent.pdf.

[16] Alexander L. Constrained by precedent. Southern California Law Review. 1989;63:1–64.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220446

Linking Appellate Judgments to Tribunal Judgments - Benchmarking Different ML Techniques

Charles CONDEVAUX [a,1], Bruno MATHIS [b] Sid Ali MAHMOUDI [a]
Stéphane MUSSARD [a] and Guillaume ZAMBRANO [a]
[a] *CHROME, University of Nîmes, France*
[b] *Centre Européen de Droit et d'Economie, ESSEC Business School, France*

Abstract. The typical judicial pathway is made of a judgment by a tribunal followed by a decision of an appellate court. However, the link between both documents is sometimes difficult to establish because of missing, incorrect or badly formatted references, pseudonymization, or poor drafting specific to each jurisdiction. This paper first shows that it is possible to link court decisions related to the same case although they are from different jurisdictions using manual rules. The use of deep learning afterwards significantly reduces the error rate in this task. The experiments are conducted between the Commercial Court of Paris and Appellate Courts.

Keywords. legal document linking, document similarity, long document processing, named entity recognition, Siamese network

1. Introduction

Although a case may be transferred from one court to another, each tribunal (court of first instance), court of appeal or supreme court gives its own identifier to each decision referring to the same case. In most countries, any decision from a higher court does not necessarily identify the underlying judgment with a unique number, nor is it always associated with publicly available metadata. This is because tribunals and courts are independent from each other. Their information system is more geared toward the production of the decision than to the management of the relationship with the parties involved by the case.

France is implementing an open data program of all judicial decisions. We anticipate that re-users will feel the lack of associated metadata and will seek alternative means to construct the judicial pathway of a case.

The aim of this research is to find out the most efficient method to link first-instance decisions (by tribunals) to second-instance one (by courts of appeals) despite the possible absence of metadata and explicit references in the text body. In practice, this model should help determine, for every outgoing appellate decision, what its original judgment

[1]Corresponding Author: charles@condevaux@unimes.fr

was, to determine the pair <First-instance,Appeal> relating to the same case. For this purpose, different techniques can be used.

- Matching pairs with rules may be a first solution to solve the problem. However success highly depends on how dates, party names and jurisdiction names are written.
- Matching pairs using automatically extracted metadata (NER) and standard ML algorithms (SVM, Random Forests...) allows for better generalization, but performance remains dependent on the underlying extraction models and the text representation (bag of words, Word2vec, etc.).
- Matching pairs with transformer models:
  Transformers ([1]) applied on raw text should outperform the previous approaches. However, due to their limitations in processing long documents, the use of sentence embedding seems to be a good option (see [2]). Another one is to summarize the document, see for instance [3] and [4] for some applications to legal documents. This significantly reduces the length of the text and eases comparisons. Finally, the use of efficient versions of self-attention (i.e. local attention) makes it possible to handle long documents efficiently while ensuring low memory usage.

The aim of this paper is threefold:

1. Extract metadata that can be used to link court decisions.
2. Build a dataset of pairs of court decisions (that match and do not match) in order to further train some machine learning and transformers models.
3. Compare traditional machine learning algorithms (SVM, XGBoost, etc...) to transformer based models to gauge similarity between documents.

2. Related works

Recent papers relate to similarity between judicial documents with different perspectives.

The first category of papers seek to predict a judgment given previous ones on similar cases. Research was first focused on long-standing statistical models like SVM, see [5]. It moved later on to neural networks, see [6], and then on those specialized in similarity, such as Siamese networks. For instance, [7] introduces a model mixing one-shot learning with recurrent neural networks and an attention mechanism to predict, on the basis of similarity of facts, the polarity of a judicial outcome (confirmation or reversal). These results are robust for different types of embeddings.

The second category of papers study Similar Case Matching (SCM). One key use-case to lawyers is to assess consistency of case law, that is, to check that rulings from a supreme court do not diverge too much between similar cases. [8] analyze the similarity and the relevance of rulings of the Court of Justice of the European Union, where relevant cases are identified through their cosine similarity, Jaccard's similarity and words mover's distance based on different vectorization schemes. Recently, [9] proposed a similarity model in order to measure the divergence level among French Cour de Cassation rulings for similar cases.

The third perspective focuses on document and information retrieval. Retrieval-oriented models are able to extract similar elements from a large corpus of legal docu-

ments (see [10] for a transformer-based technique and [11] for an ontology-based one). [12] made a literature review about legal information retrieval systems. They found that CNN models may outperform all other models including BERT, however their performance remains questionable for SCM.

The *jurisdictional linkage* may be both associated to the first and the third categories. At least in France, there is no process to build the pathway of a case through different jurisdictions. In this research, judgment dates, and names of parties, lawyers and jurisdictions are key similarity factors, while in works mentioned in the second category, facts and citations are major similarity factors.

3. Methodology: *jurisdictional linkage*

Solving a jurisdictional linkage problem requires several steps. It is necessary to collect a corpus of compatible documents, to process them to build a sufficiently large training dataset and then to apply a set of machine learning methods to solve a matching task.

3.1. The corpora

The first-instance corpus is made of 360,000 decisions from the Tribunal de Commerce de Paris, dated from 2000 to 2010. It includes a majority of judgments proper, which can be subject to an appeal. On average, one out of ten judgments is subject to an appeal. The corpus also includes interim orders, divestment orders, and other procedural acts, which are also considered as judicial decisions.

The second-instance corpus is made of appellate decisions that include the word sequence "Tribunal de Commerce de Paris", which refers to the biggest jurisdiction of judgment among the 130 or so commercial tribunals in France. This represents some 13,000 decisions. The vast majority of them follow a first-instance judgment. The rest follow a decision from Court of cassation, whose corpus is out of scope of this study.

Less than a quarter in our corpus explicitly designate their underlying trial court judgment by their number. Fortunately, every appellate decision does identify the date of judgment, the jurisdiction of judgment, and the parties (claimants and defendants), mentions that are mandatory by law.

3.2. Dataset creation

Both corpora being available in a version deprived of metadata, a NER model is used to extract such metadata in the first place. Most of these are located in the header or in the upper part of the text body, when no structured header is present. The following labels are created:

- Number ('numéro de répertoire général')
- Date of judgment
- Jurisdiction of judgment
- Party (claimant or defendant)

Two datasets are built, one of 638 appellate decisions, the other made of 1496 judgments from several areas of law, not only in the commercial area. They are manually an-

notated with these labels. Each decision may be annotated with several claimants and/or several defendants.

The data labelling is done on the Kairntech platform [13] and statistics are reported in Table 1.

Label	First-instance	Appeal
Number of decisions	1496	638
Date of appellate decision	N/A	606
Date of judgment	2468	604
Party	4483	2086
Id of appellate decision	N/A	553
Id of first-instance decision	1438	230
Appellate court	N/A	611
Tribunal	1487	614
Total number of annotations	9876	5304

Table 1. Manual annotations per label (the annotations on the Jurisdiction label are split by a simple rule between Appellate Court and Tribunal)

3.3. Post-processing and normalization

Since NER is an extractive and not a generative task, some labels require prior normalization. As dates can be written in either literal or numeric form, a small BART model [14] is trained to convert dates to *yyyy-mm-dd* format. For this purpose a dataset is artificially created mixing numerical format, literal format, punctuation, typos and noise. Thousand of examples are generated, i.e:

- Original string: "*18 Juin 2010*"
- Noisy string: "*e le 18 jiun, 2O1O...*"
- Label: "*2010-06-18*"

As the NER task also gives the location of the extracted sequence in the document, the Seq2Seq generative task is performed on an extended segment of a few words before and after it to correct some truncated predictions.

This normalization allows to classify documents by date and to extract the subset of 13,000 court of appeal decisions that are potential candidates for the linking process with the Paris Commercial Court.

3.4. Matching pairs with rules

The objective is to find the maximum number of matching pairs for a minimal effort.

In the absence of metadata, matching two documents requires a preliminary NER step. The inference of the model described above on the whole corpus allows us to get the underlying judgment number for the appeal decisions, and the judgment number for the judgments. Since these identifiers are unique, 2770 pairs are reliably built at the end of this step.

The case where the underlying judgment number is not available is also addressed. Many judgments are given on the same day in any jurisdiction, and examining the parties, either claimants or defendants, is a good way to disambiguate them. It is highly unlikely that the same party will be involved in several separate cases judged on the

same day and in the same jurisdiction. Even if some parties are natural persons dutifully pseudonymized in publicly available decisions and thus not suitable for disambiguation, legal persons can help disambiguate multiple judgments. The second proposed method therefore consists of matching date/jurisdiction/party triples, after eliminating pseudonyms and lower-casing legal persons. As several parties can match for the same judgment, the duplicates are finally removed.

This method remains largely imperfect since the parties, which cannot be normalized, are marred by numerous syntactic variants. Indeed, it only provides 44 more matching pairs, bringing our total to 2814. The method could be extended to the names of the lawyers, which are in principle available on the judgment and appeal decisions, and are non-pseudonymized. However, this approach is not applied because lawyers were not annotated in the first stage of the NER. We hypothesize that this would have increased our number of pairs by a few dozen at the cost of a significant additional annotation effort. The lawyers' names, while more homogeneous than the parties' names, are also subject to syntactic variants.

3.5. Matching pairs from meta-data

The manual rule-based program described above is used to build a dataset. Since document pairs are now known, the production of counter-examples is relatively simple and is done by randomly matching appellate with first-instance decisions. The task is defined as a binary classification task on tabular data made of two times 2770 rows and 5 columns (features) represented as strings:

- Date of appellate decision
- Date of judgment found in the appellate decision
- Jurisdiction of judgment found in the appellate decision
- Date of judgment found in the first-instance decision
- Jurisdiction of judgment found in the first-instance decision.

A binary classification task is then defined by concatenating the features of the document pairs. This task is then solved using standard machine learning algorithms such as SVM, logistic regression, multilayer perceptron (MLP) and decision-tree methods.

3.6. Matching pairs of documents with transformers

The classification task presented above can be adapted to a contemporary Transformer-based approach. Contrary to the experiments presented so far, these models do not need to define a set of features but are applied directly on the raw document if a sufficiently large annotated dataset is available. This has several advantages:

- It is not necessary to define manual rules
- The model will determine itself the necessary and discriminating tokens
- Its generalization is usually better and more robust given the high number of parameters and the pretraining process.

Since the proposed task is actually a similarity task, a Siamese neural network [15,16,17,18] approach is preferred, comparing the sentence embedding of the documents. A siamese network works on two parallel neural networks sharing their weights

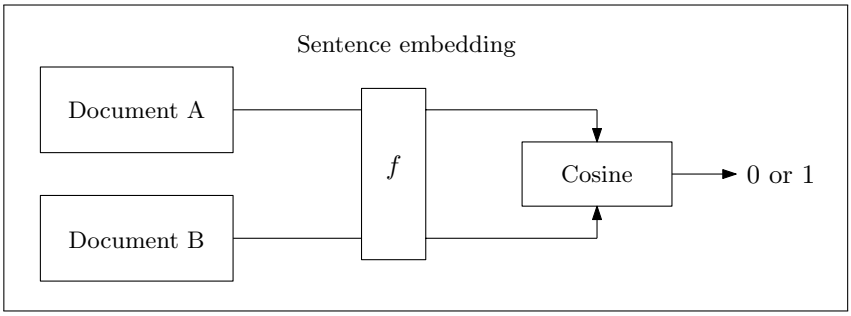


Figure 1. Architecture of a Siamese Neural Network with a Sentence Embedding (CamemBERT)

but working on different inputs. It allows to obtain a unique and static representation with a fixed length to ease comparisons. The architecture is presented in Fig 1.

The general architecture is similar to the one proposed by SBERT [19] but the underlying model is a CamemBERT [20] model warm-started from the official checkpoint and fine-tuned on 30Gb of legal data (MLM).

However, the processing of legal documents imposes an additional constraint since transformer based models are for most of them trained for 512 tokens sequences and they cannot process long sequences due the quadratic complexity of self-attention. To measure which part of a decision is likely discriminating for the similarity task, four experiments are proposed on different parts of the document:

- First 512 tokens
- Last 512 tokens
- Full document (up to 16384 tokens)
- Summarized document (up to 512 tokens)

The data extracted by NER and used for the construction of the dataset are mostly concentrated in the first part of the document, so we can assume that the first 512 tokens are sufficient to reliably constitute the pairs. Since the decisions can be long, especially for the court of appeal, the last 512 tokens may not contain any discriminating information, lower performances are to be expected. In order to process the documents without truncating them, we rely on a conversion script to extrapolate from an existing model.[2]

The fourth experiment based on summarization is particular since it is necessary to develop a summarization model alongside. This process is done in two steps. First, a dataset of 70K pairs of decisions/summaries is extracted from Légifrance[3] and focused on the Cour de Cassation and the Conseil d'Etat (summaries are available only from supreme courts). A BARThez model [14,21] (Seq2Seq) is warm-started and fine tuned on the text in-filling and de-noising tasks on the 30Gb corpus of legal data. The model is then extrapolated (16384 tokens) thanks to the above script and fine-tuned this time on the summarization task. Although the jurisdictions processed are different from those of our matching task, the trained model generates concise summaries of the main elements that the judge motivates in his/her decision.

[2] https://github.com/ccdv-ai/convert_checkpoint_to_lsg
[3] https://www.legifrance.gouv.fr/

4. Experiments and results

Three types of experiments are presented, one related to the NER tasks and another one based on extracted metadata to which standard machine-learning models are fitted. The last one based on transformers that directly process the content of the documents. The experiments are performed with a five-fold cross-validation except for the NER task.

4.1. NER results

Two dataset configurations are tesed. The first one related to the tribunal only and the second one to the court of appeal. The model used is the legal CamemBERT presented above. Results are reported in Table 2.

F-measure	**First-instance**	**Appeal**
Date of Appellate decision	N/A	92.93
Date of Judgment	90.86	82.11
Parties	83.1	76.02
ID of Appeal	N/A	90.35
ID of First-instance	88.43	82.61
Appellate Court	N/A	91.12
Tribunal	96.32	88.26

Table 2. F-measure on the NER task.

4.2. Metadata-based classification

Since the task is defined as a classification problem on tabular data and the features are strings, an embedding method is needed to transform the inputs before solving the task.

For this, two methods are used, a Bag-of-words (BoW) method where the features are summed up and another one based on a pre-trained embedding model where the features are averaged for each document. To represent the pairs, those of the two documents are concatenated.

The experiments are conducted on 6 standard machine learning models for processing tabular data:

- Linear SVM
- SVM (rbf kernel)
- Logistic regression
- Multilayer perceptron (1e-4 learning rate, ReLU activation, 3 layers of size 100)
- XGBoost (500 estimators, max depth of 10, 0.1 learning rate)

Results are reported in Table 3.

We can directly observe that the performances vary widely depending on the representation method. The simplest to implement (BoW) gives higher results on average except for the random forest. From the results, providing (noisy) metadata is enough to correctly find over 95% of the pairs.

	Linear SVC	SVM	Logistic Reg.	MLP	XGBoost
BOW					
F-measure	88.48	96.23	87.14	96.06	95.08
Avg. variation	2.00	0.51	0.81	0.23	0.29
Embedding					
F-measure	71.53	71.60	73.40	75.43	86.02
Avg. variation	0.58	0.50	0.93	1.44	0.94

Table 3. F-measure of standard ML algorithms on the matching task.

4.3. Raw document based classification with transformers

Transformers with Siamese architecture allow to compare documents without providing metadata. Their use can thus be generalized quite easily as long as a labeled set of pairs is available. The summarization task is presented first because it is reused later and the best performing models remain the non-linear ones (SVM, MLP, XGBoost).

4.3.1. Summarization task

The summarization task performed on 70k pairs of decisions/summaries from the Cour de Cassation and the Conseil d'Etat is trained on a specialized BARThez capable of processing sequences of 16384 tokens. The model is fine-tuned during 10 epochs, with an Adam optimizer, a learning rate of 8e-5, 10% of warmup steps, a linear learning rate decay, a batch size of 32 and a maximum generation length of 512 tokens. We use the summarization example script from HuggingFace[4] to train and evaluate. The model achieves 61.74/49.39/55.47 as Rouge1/Rouge2/RougeL scores [22] with a length penalty of 2 and 5 beams. Generated summaries for the matching task rely on the same hyperparameters.

4.3.2. Matching task

For the following tasks relying on the Siamese architecture, the model is trained during 15 epochs with an Adam optimizer, a learning rate of 5e-5, 500 warmup steps, a linear learning rate decay and a batch size of 32. The loss function used is a cosine similarity loss which computes the euclidian distance between the cosine similarity between `<s>` tokens from both documents and the associated label. Results are reported in Table 4.

	512 First tokens	512 Last tokens	Full document	Summary
F-measure	99.20	96.84	99.44	90.15
Avg. variation	0.17	0.54	0.11	0.78

Table 4. F-measure of transformer models on the matching task.

As expected, using the first tokens of the decision produces much better performances than using the last tokens, since discriminating data such as decision references, dates and lawyers' names are mostly located at the beginning. Processing up to 16384 tokens seems to show a marginal effect but this gain reduces the error rate by 30% (0.80% to 0.56%) which is significant for performances above 99%.

[4] https://github.com/huggingface/transformers/blob/main/examples/pytorch/summarization

The summary-based method provides disappointing results, possibly because summaries are very concise and therefore less informative, which makes the comparison of documents difficult. There is also some bias to train summaries on supreme courts rulings, which are oriented to doctrine, and infer them to lower-court decisions that focus more on facts.

The Siamese architecture is useful in production because a sentence embedding is learned during training. Thus, these embeddings can be pre-computed for each document, allowing inference on large volumes efficiently, the cosine distance being a simple normalized inner product.

5. Conclusion

We have presented a methodology to reconstitute the judicial path of a case. Through NER and simple predefined rules, a dataset of document pairs was built, allowing the task to be solved using machine and deep learning. Two methods have been proposed, one leveraging metadata, the other one focusing directly on the raw document. The experiments carried out have shown that the use of transformer based models, trained on the first part of the documents (512 first tokens), achieve very high performance and commit only few errors compared to the other methods presented and related to metadata. In practice, the use of Siamese architecture also has advantages in production. Because each decision can be represented by a pre-computed fixed length vector through sentence embedding, the comparison of a document against a large number of candidates can be done efficiently. On the other hand, methods exploiting metadata remain very dependent on the quality of the underlying NER model, the embedding method and the algorithm used to solve the task.

Acknowledgments

The authors would like to thank the French National Research Agency for funding the LAWBOT project ANR-20-CE38-0013: Deep Learning for Judicial Outcome Prediction.

References

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All you Need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc.; 2017. Available from: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[2] Westermann H, Savelka J, Walker VR, Ashley KD, Benyekhlef K. Sentence Embeddings and High-speed Similarity Search for Fast Computer Assisted Annotation of Legal Documents. CoRR. 2021;abs/2112.11494. Available from: https://arxiv.org/abs/2112.11494.

[3] Arpan Mandal SM Paheli Bhattacharya, Ghosh S. Improving Legal Case Summarization Using Document-Specific Catchphrases. In: JURIX, Legal Knowledge and Information Systems; 2021. p. 76-81.

[4] Aniket Deroy KG Paheli Bhattacharya, Ghosh S. An Analytical Study of Algorithmic and Expert Summaries of Legal Cases. In: JURIX, Legal Knowledge and Information Systems; 2021. p. 90-9.

[5] Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. PeerJ Computer Science. 2016;2:e93.
[6] Chalkidis I, Androutsopoulos I, Aletras N. Neural Legal Judgment Prediction in English. arXiv preprint arXiv:190602059. 2019.
[7] Condevaux C, Harispe S, Mussard S, Zambrano G. Weakly Supervised One-shot Classification using Recurrent Neural Networks with Attention: Application to Claim Acceptance Detection. In: JURIX 2019 32nd International Conference on Legal Knowledge and Information Systems. Madrid, Spain; 2019. Available from: https://hal.archives-ouvertes.fr/hal-02407405.
[8] Moodley K, Serrano PVH, van Dijck G, Dumontier M. Similarity and relevance of court decisions: A computational study on CJEU cases. In: JURIX; 2019. p. 63-72.
[9] Charmet T, Cherichi I, Allain M, Czerwinska U, Fouret A, Sagot B, et al. Complex Labelling and Similarity Prediction in Legal Texts: Automatic Analysis of France's Court of Cassation Rulings. In: LREC 2022 - 13th Language Resources and Evaluation Conference. Marseille, France; 2022. Available from: https://hal.inria.fr/hal-03663110.
[10] Vuong YTH, Bui QM, Nguyen HT, Nguyen TTT, Tran V, Phan XH, et al. SM-BERT-CR: a deep learning approach for case law retrieval with supporting model. Artificial Intelligence and Law. 2022 Aug. Available from: https://doi.org/10.1007/s10506-022-09319-6.
[11] Castano S, Falduti M, Ferrara A, Montanelli S. A knowledge-centered framework for exploration and retrieval of legal documents. Information Systems. 2022;106:101842. Available from: https://www.sciencedirect.com/science/article/pii/S0306437921000788.
[12] Zhong H, Xiao C, Tu C, Zhang T, Liu Z, Sun M. How does NLP benefit legal system: A summary of legal artificial intelligence. arXiv preprint arXiv:200412158. 2020.
[13] Geißler S. The Kairntech Sherpa – An ML Platform and API for the Enrichment of (not only) Scientific Content. In: Proceedings of the 1st International Workshop on Language Technology Platforms. Marseille, France: European Language Resources Association; 2020. p. 54-8.
[14] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics; 2020. p. 7871-80. Available from: https://aclanthology.org/2020.acl-main.703.
[15] Bromley J, Guyon I, LeCun Y, Säckinger E, Shah R. Signature Verification using a "Siamese" Time Delay Neural Network. In: Cowan J, Tesauro G, Alspector J, editors. Advances in Neural Information Processing Systems. vol. 6. Morgan-Kaufmann; 1993. .
[16] Chopra S, Hadsell R, LeCun Y. Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1; 2005. p. 539-46 vol. 1.
[17] Koch GR. Siamese Neural Networks for One-Shot Image Recognition; 2015. .
[18] Chicco D. In: Cartwright H, editor. Siamese Neural Networks: An Overview. New York, NY: Springer US; 2021. p. 73-94. Available from: https://doi.org/10.1007/978-1-0716-0826-5_3.
[19] Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics; 2019. Available from: https://arxiv.org/abs/1908.10084.
[20] Martin L, Muller B, Suárez PJO, Dupont Y, Romary L, de La Clergerie ÉV, et al. Camembert: a tasty french language model. arXiv preprint arXiv:191103894. 2019.
[21] Kamal Eddine M, Tixier A, Vazirgiannis M. BARThez: a Skilled Pretrained French Sequence-to-Sequence Model. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics; 2021. p. 9369-90. Available from: https://aclanthology.org/2021.emnlp-main.740.
[22] Lin CY. ROUGE: A Package for Automatic Evaluation of Summaries. In: Text Summarization Branches Out. Barcelona, Spain: Association for Computational Linguistics; 2004. p. 74-81. Available from: https://aclanthology.org/W04-1013.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220447

Stable Normative Explanations

Guido GOVERNATORI [a] Francesco OLIVIERI [b],
Antonino ROTOLO [c], Matteo CRISTANI [d]

[a] *Centre for Computational Law, Singapore Management University, Singapore*
[b] *Institute for Integrated and Intelligent Systems, Griffith University, Australia*
[c] *Alma AI, University of Bologna, Italy*
[d] *Department of Computer Science, University of Verona, Italy*

Abstract. Modelling the concept of explanation is a central matter in AI systems, as it provides methods for developing eXplainable AI (XAI). When explanation applies to normative reasoning, XAI aims at promoting normative trust in the decisions of AI systems: in fact, such a trust depends on understanding whether systems predictions correspond to legally compliant scenarios. This paper extends to normative reasoning a work by Governatori *et al.* (2022) on the notion of stable explanations in a non-monotonic setting: when an explanation is stable, it can be used to infer the same normative conclusion independently of other facts that are found afterwards.

Keywords. Defeasible Deontic Logic, Stable Explanation, Symbolic XAI

1. Introduction

The literature on the concept of *explanation* is vast (especially in philosophy; see, among many others, [1,14]), and the AI community is recently paying more and more attention to it due to the development of eXplainable AI (XAI) [12]. The AI&Law community has, in turn, a long tradition in this direction [3], since '*transparency*' and '*justification*' of legal decision-making both require formalising normative explanations.

We propose the novel idea of *stable normative explanation* extending the notion of *stable inference* of [4]. Roughly speaking, the problem of determining a stable normative explanation for a certain legal conclusion means to identify a set of facts, obligations, permissions, and other normative inputs able to ensure that such a conclusion continues to hold when new facts are added to a case. This notion is interesting from a logical point of view (think about the classical idea of inference to the best explanation), but it can also pave the way to develop symbolic models for XAI when applied to the Law (consider, for instance, systems of predictive justice [11]).

Reasoning with legal norms exhibits features that distinguish it from other types of reasoning. For instance, while examining a case, we are limited to (i) the facts presented (and the admissible ones) for that case, and (ii) the norms in force for the time relevant to the case itself, norms that, sometimes, stem from different sources. Given the facts of the case, the proceeding aims at determining what *legal requirements* (expressed as obligations, prohibitions and permissions) hold, and whether such legal requirements have been fulfilled. If more/new facts were presented, the outcome of a case might be quite different or can even be modified; moreover, such additional facts may be themselves the outcome

of some norms establishing other legal requirements (i.e., obligations, prohibitions and permissions), possibly not in the sphere of control of the current proceeding.

Accordingly, one of the major issues is how to ensure a specific outcome for a case, which, in an adversarial setting, can be understood as how to ensure that the facts presented by a party are 'resilient' to the attacks from the opponent. Furthermore, such an issue is not restricted to adversarial situations only. Still, it is relevant even in cases where one party does their due diligence to determine if they comply with a particular piece of legislation (e.g., to identify all requirements that a business must satisfy to legally serve alcohol in an entertainment environment, according to the Queensland liquor licensing and gaming regulations).

Let us ground the above discussion with a concrete scenario. The Australian Spent Conviction discipline governs (1) the conditions under which a conviction is spent, and (2) when it is permitted to withhold information about it (or when it is mandatory to disclose that a conviction has occurred). At the federal level, Spent Conviction is regulated by Part VII C of the Crimes Act 1914; in addition, States and Territories enacted their own legislation and schemas supplementing and complementing the federal one.

Part VII C of the Crimes Act 1914 consists of six divisions. Division 1 gives the terms and definitions for the topics. Divisions 2, 3, and 4 establish the baseline conditions for (i) when a conviction is spent, (ii) when a person is permitted to withhold information about a (spent) conviction, (iii) when a person is required to provide such information, and (iv) when third parties are either permitted or forbidden to disclose information they might have about a (spent) conviction. Division 5 deals with "administrative" aspects (complaints) of improper releases of spent conviction information. Finally, Division 6 specifies the exclusions (exceptions) to Divisions 2 and 3. Among the various provisions, Section 85ZZGB (Exclusion: disclosing information to a person or body) recites:

> Divisions 2 and 3 do not apply in relation to the disclosure of information to a prescribed person or body if:
>
> (a) The person or body is required or permitted by or under a prescribed Commonwealth law, a prescribed State law or a prescribed Territory law, to obtain and deal with information about persons who work, or seek to work, with children; and
>
> (b) The disclosure is for the purpose of the person or body obtaining and dealing with such information in accordance with the prescribed law.

Part VII C of the Crimes Act 1914 clearly demonstrates the abovementioned issues. First of all, examining the baseline conditions given in Divisions 2 and 3 is not sufficient to determine if the information about a spent conviction can be withheld: the exclusions specified in Division 4 must be considered. Moreover, Section 85ZZGB specifies that the conditions set in Division 2 or 3 depend upon deontic aspects ("required" or "permitted") determined by regulatory instruments outside what is specified by Part VII C, which can be assumed as (external deontic) facts of the case.

In this paper, we work with a non-monotonic formalism (Defeasible Deontic Logic) apt to model norms and able to deal with exceptions and deontic concepts. The logic is needed to provide a precise and formal grounding of the problem of *stability* and *stable normative explanation*. We also examine the computational complexity of ensuring stability.

2. Stable Explanations

Finding a normative explanation for a certain normative conclusion l (such as an obligation) means determining as input a piece F of normative information that supports the derivation of l through norms and other rules of the normative system. If F is a stable explanation, then adding new facts to that explanation does not affect its power to explain the normative conclusion. Stable explanations are naturally considered because they are insensitive to input knowledge changes. In other terms, stable explanations are, to some extent, *monotonic*, even when the considered logic is not.

In this investigation, we work on a deontic extension of Defeasible Logic (DL), called Defeasible Deontic Logic (DDL). In DDL, we have three types of elements: (1) facts, (2) rules, and (3) superiority relation $>$. Facts are the input knowledge describing those indisputable things that are true beyond any doubt. Rules are ways to obtain (normative) conclusions that are considered plausible (or typical), whereas the superiority relation is thought of as a means to establish whether one rule for a conclusion might prevail against another rule for the opposite conclusion.

DDL is DL plus the deontic operators O and P, respectively, for obligations and permissions, and the operator $\otimes$ according to which an expression $a \otimes b$ means that a is obligatory, but if such an obligation is violated, then b is obligatory and compensates this violation [8]. In addition to standard rules (which are hereafter referred to as *constitutive rules*, with arrow $\Rightarrow_{\mathsf{C}}$), we have deontic rules, such as

$$\alpha\colon a_1, \ldots, a_n \Rightarrow_{\mathsf{O}} b \otimes c \qquad\qquad \beta\colon \mathsf{O}c, d_1, \ldots, d_m \Rightarrow_{\mathsf{P}} e.$$

If α is applicable (namely, $a_1, \ldots, a_n$ are the case), then we derive $\mathsf{O}b$. Suppose that we know $\neg b$, meaning that $\mathsf{O}b$ is violated. In this case, we derive $\mathsf{O}c$. Accordingly, if we also know that $d_1, \ldots, d_m$ are the case, then we conclude that e is permitted, i.e., that $\mathsf{P}e$.

Two peculiar features of DDL make the idea of normative explanation not obvious:

- The set F of facts may include deontic expressions such as $\mathsf{O}p$, and $\neg\mathsf{P}q$. Such deontic facts encode a *normatively* indisputable input. For example, suppose the set of rules represents norms of the Italian legal system. In that case, we can take $\mathsf{O}b \in F$ as indisputable as grounded on the Italian constitution or because it is imported from European law.
- DDL adopts the concept of rule conversion [9], which amounts here to use non-deontic rules to derive obligations and permissions. Consider the rule $\alpha\colon a \Rightarrow_{\mathsf{C}} b$, and assume we prove $\mathsf{P}a$. We can use α to determine that b is permitted. For example, in football, if the ball passing completely over the goal line between the goal posts and under the crossbar 'counts as' scoring a goal and it is permitted for the ball to pass such a goal line, then we can indeed derive that scoring a goal is permitted.

To illustrate the idea of stable normative explanation, consider the following example.

Example 1. *Suppose the Law forbids engaging in credit activities without a credit license. If you violate this prohibition, the civil penalty is 2,000 penalty units. Furthermore, such activities are permitted for a person acting on behalf of another person (the principal) when the person is an employee or the director of the principal and the principal holds a credit license. Moreover, some conditions are specified under which a person could be banned from credit activities. For example, a person is banned if they become insolvent. Finally, using the equity mobilised by the credit institutions counts as a credit activity.*

γ: $\Rightarrow_{\mathsf{O}} \neg creditActivity \otimes civilPenalty$
δ: $creditLicense \Rightarrow_{\mathsf{P}} creditActivity$
ϵ: $actsOnBehalfPrincipal, principalCreditLicense \Rightarrow_{\mathsf{P}} creditActivity$
ζ: $\mathsf{O}banned \Rightarrow_{\mathsf{O}} \neg creditActivity$
η: $insolvent \Rightarrow_{\mathsf{O}} banned$
θ: $\mathsf{O}pay, \neg pay \Rightarrow_{\mathsf{C}} insolvent$
ι: $equity \Rightarrow_{\mathsf{C}} creditActivity$

where $\delta > \gamma$, $\epsilon > \gamma$, $\zeta > \epsilon$, $\iota > \gamma$, *and* $\iota > \zeta$.

- *The set* {*creditLicence*} *is stable for* P*creditActivity;*
- *The set* {*actsOnBehalfPrincipal*, *principalCreditLicense*} *is not stable for*

- *The set* {P*equity*} *is stable for* P*creditActivity;*
- *The set* {*creditAcvitiy*} *is not stable for the conclusion* O*civilPenalty (if we add, e.g., creditLicence).*

3. Defeasible Deontic Logic

Defeasible Logic [13,2] is a simple, flexible, and efficient rule-based non-monotonic formalism, whose strength lies in its constructive proof theory that allows drawing meaningful conclusions from a (potentially) conflicting and incomplete knowledge base. In non-monotonic systems, more accurate conclusions can be obtained when more pieces of information become available. Many variants of DL have been proposed for the logical modelling of different application areas, especially for legal reasoning (for an overview of the literature, see [10]).

In this research, we focus on the Defeasible Deontic Logic's framework advanced in [5], that allows us to model a large variety of normative concepts, as well as to determine what prescriptive behaviours are in force in a given situation.

We shall now briefly recall the main elements of the logic, and start by defining the language of a defeasible deontic theory. Let PROP be a set of propositional atoms, and Lab be a set of arbitrary labels (the names of the rules). Lower-case Roman letters denote literals, whereas lower-case Greek letters denote rules. Accordingly, PLit = PROP $\cup$ $\{\neg l \mid l \in \text{PROP}\}$ is the set of *plain literals*, the set of *deontic literals* is ModLit = $\{\Box l, \neg\Box l \mid l \in \text{PLit} \wedge \Box \in \{\mathsf{O}, \mathsf{P}\}\}$, and finally, the set of *literals* is Lit = PLit $\cup$ ModLit. The *complement* of a literal l is denoted by $\sim l$: if l is a positive literal p then $\sim l$ is $\neg p$, and if l is a negative literal $\neg p$ then $\sim l$ is p.

Definition 1 (Defeasible Deontic Theory)**.** *A* defeasible deontic theory D *is a tuple* $(F, R, >)$*, where* F *is the set of facts,* R *is the set of rules, and* $>$ *is a binary relation over* R *(called* superiority relation*).*

The set of facts $F \subseteq$ Lit denotes simple pieces of information that are considered to be always true. A theory is meant to represent a normative system, where the rules encode the norms of such a system, and the set of facts corresponds to "*factual information*", or "*given deontic positions*" (as, for instance, indisputable obligations or deontic positions imported from other higher-ranked normative systems). The rules are used to conclude the institutional facts, obligations and permissions that hold given the set of facts.

The set of rules R is finite and contains three *types* of rules: *strict rules*, *defeasible rules*, and *defeaters*. Rules are also of two *kinds*:

- *Constitutive rules* (non-deontic, counts-as rules) R^{C} model constitutive statements;
- *Deontic rules* model prescriptive behaviours, which are either *obligation rules* R^{O} determining when and which obligations are in force, or *permission rules* representing *strong* (or *explicit*) permissions R^{P}.

Lastly, the *superiority* relation, $> \subseteq R \times R$, solves conflicts among rules' conclusions.

Following the ideas of [8], obligation rules gain more expressiveness with the *compensation operator* $\otimes$ for obligation rules, which is to model reparative chains of obligations. Intuitively, $a \otimes b \otimes c$ means that a is the primary obligation, but if, for some reason, we fail to comply with a, then b becomes the new obligation in force, and so on for c if we also fail with b. This operator, called $\otimes$-expressions, is hence used to build chains of "preference reparations" (c is still acceptable but less preferred than b).

Definition 2 (Rule)**.** *A* rule *is an expression of the form* $\alpha\colon A(\alpha) \hookrightarrow_{\Box} C(\alpha)$*, where*

1. $\alpha \in$ Lab *is the unique name of the rule;*
2. $A(\alpha) \subseteq$ Lit *is the set of antecedents;*
3. *An arrow* $\hookrightarrow \in \{\Rightarrow, \rightsquigarrow\}$ *denotes, respectively, defeasible rules, and defeaters;*
4. $\Box \in \{\mathsf{C}, \mathsf{O}, \mathsf{P}\}$*;*
5. $C(\alpha)$ *is the consequent, which is either*
   (a) *a single plain literal* $l \in$ PLit*, if (i)* $\hookrightarrow \equiv \rightsquigarrow$ *or (ii)* $\Box \in \{\mathsf{C}, \mathsf{P}\}$*, or*
   (b) *an* $\otimes$*-expression, if* $\Box \equiv \mathsf{O}$.

If $\Box = \mathsf{C}$ then the rule is used to derive non-deontic literals (constitutive statements), whilst if $\Box$ is O or P then the rule is used to derive deontic conclusions (prescriptive statements). The conclusion $C(\alpha)$ is a single literal in case $\Box = \{\mathsf{C}, \mathsf{P}\}$, or an $\otimes$-expression when $\Box = \mathsf{O}$. Note that $\otimes$-expressions can only occur in prescriptive rule though we do not admit them on defeaters (see [5]).

We use some standard abbreviations on rule sets. The set of defeasible rules is $R_{\Rightarrow}$, the set of defeaters is R_{dft}. $R^{\Box}[l]$ is the set of rules with conclusion l and modality $\Box$, while $R^{\mathsf{O}}[l, i]$ denotes the set of obligation rules where l is the i-th element in the $\otimes$-expression. Given that the consequent of a rule is either a single literal or an $\otimes$-expression, in what follows, we are going to shorten the notation and use $l \in C(\alpha)$.

Definition 3 (Tagged modal formula)**.** *A* tagged modal formula *is an expression of the form* $\pm\partial_{\Box} l$*, with the following meanings*

- $+\partial_{\Box} l$*: l is* defeasibly provable *(short,* provable*) with mode* $\Box$*,*
- $-\partial_{\Box} l$*: l is* defeasibly refuted *(short,* refuted*) with mode* $\Box$*;*

Accordingly, the meaning of $+\partial_{\mathsf{O}} p$ is that p is provable as an obligation, and $-\partial_{\mathsf{P}} \neg p$ is that we have a refutation for the permission of $\neg p$. Similarly, for the other combinations.

Definition 4 (Proof)**.** *Given a defeasible deontic theory D, a* proof P of length m in D *is a finite sequence* $P(1), P(2), \ldots, P(m)$ *of tagged modal formulas, where the proof conditions defined in the rest of this paper hold.* $P(1..n)$ *denotes the first n steps of P.*

The notational convention '$D \vdash \pm\partial_{\Box} l$*' means that there is a proof P for* $\pm\partial_{\Box} l$ *in D.*

Core notions in DL are that of *applicability/discardability* of a rule. This paper uses the one developed in [9,6]. As knowledge in a defeasible theory is circumstantial, given a defeasible rule like '$\alpha\colon a, b \Rightarrow_{\Box} c$', there are four possible scenarios: the theory defeasibly proves both a and b, the theory proves neither, the theory proves one but not the other. Naturally, only in the first case, where both a and b are proved, we can use α to *support/try to conclude* $\Box c$.

Definition 5 (Applicability)**.** *Assume a defeasible deontic theory* $D = (F, R, >)$.

1. *Rule* $\alpha \in R^{\mathsf{C}} \cup R^{\mathsf{P}}$ *is* applicable *at* $P(n+1)$*, iff for all* $a \in A(\alpha)$
   (a) *if* $a \in$ PLit*, then* $+\partial_{\mathsf{C}} a \in P(1..n)$,
   (b) *if* $a = \Box q$*, then* $+\partial_{\Box} q \in P(1..n)$*, with* $\Box \in \{\mathsf{O}, \mathsf{P}\}$,
   (c) *if* $a = \neg\Box q$*, then* $-\partial_{\Box} q \in P(1..n)$*, with* $\Box \in \{\mathsf{O}, \mathsf{P}\}$.
2. *Rule* $\alpha \in R^{\mathsf{O}}$ *is* applicable at index i and $P(n+1)$ *iff Conditions 1a–1c hold, and*
   (d) $\forall c_j \in C(\alpha)$, $j < i$*, then* $+\partial_{\mathsf{O}} c_j \in P(1..n)$ *and* $+\partial_{\mathsf{C}} {\sim} c_j \in P(1..n)$.
3. *Rule* $\alpha \in R^{\mathsf{C}}$ *is* applicable *at* $P(n+1)$ *for* $+\partial_{\Box} l$ *where* $\Box \in \{\mathsf{O}, \mathsf{P}\}$*, iff*
   (e) $\alpha \in R^{\mathsf{C}}[l]$,
   (f) *for all* $a \in A(\alpha)$, $a \in$ PLit*, and* $A(\alpha) \neq \emptyset$,
   (g) $+\partial_{\Box} a \in P(1..n)$.

Note that *discardability* of a rule is obtained by applying the principle of *strong negation* to the definition of applicability [7], and thus omitted.

Condition 1 establishes that (a, b) every positive literal has been proved, and (c) every deontic negative literal has been rejected at a previous derivation step.

Condition 2 deals with $\otimes$-chains: a rule is applicable at a certain index when each element c_j before have been proved as obligation $+\partial_{\mathsf{C}} c_j$ and violated $+\partial_{\mathsf{O}} {\sim} c_j$.

Lastly, Condition 3 formalises *rule conversion* mentioned in Section 2, which is a way to derive a conclusion with a certain modality by using rules for another modality [9]. In our case, constitutive rules can be used to derive obligations and permissions. This is formalised by Condition 3, which reads very easily: there must be a constitutive rule whose none of the antecedents is an obligation or permission (hence all of them plain literals), if all such antecedents are proved as obligations (resp. permissions) then the rule becomes applicable in supporting its conclusion as an obligation (resp. permission). Therefore, if we change rule α of above as '$\alpha\colon a, \mathsf{O}b \Rightarrow_{\mathsf{C}} c$', then this new α cannot be used through conversation and may only be used to support 'a constitutive c'.

For space reasons, we provide conditions for $+\partial_{\mathsf{O}}$ and $+\partial_{\mathsf{P}}$ only, since (i) $-\partial_{\mathsf{O}}$ and $-\partial_{\mathsf{P}}$ can be obtained by applying the strong negation principle to the positive counterparts, and (ii) the proof conditions for constitutive statements are the standard for DL [2].

Definition 6 (Obligation Proof Conditions)**.**

$+\partial_{\mathsf{O}} l$*: If* $P(n+1) = +\partial_{\mathsf{O}} l$ *then*
(1) $\mathsf{O}l \in F$*, or*
(2) $\mathsf{O}{\sim}l, \neg\mathsf{O}l, \mathsf{P}{\sim}l, \neg\mathsf{P}l \notin F$*, and*
(3) $\exists \beta \in R^{\mathsf{O}}_{\Rightarrow}[l, i] \cup R^{\mathsf{C}}_{\Rightarrow}[l]$ *s.t.*
(3.1) β *is applicable at index* i *if* $\beta \in R^{\mathsf{O}}_{\Rightarrow}[l, i]$*, or*
β *is applicable for* $+\partial_{\mathsf{O}} l$ *if* $\beta \in R^{\mathsf{C}}_{\Rightarrow}[l]$*, and*
(3.2) $\forall \gamma \in R^{\mathsf{O}}[{\sim}l, j] \cup R^{\mathsf{P}}[{\sim}l] \cup R^{\mathsf{C}}[{\sim}l]$ *either*
(3.2.1) γ *is discarded at index* j *if* $\gamma \in R^{\mathsf{O}}[{\sim}l, j]$*, or*
(3.2.2) $\exists \zeta \in R^{\mathsf{O}}[l, k] \cup R^{\mathsf{C}}_{\Rightarrow}[l]$ *s.t.*
(3.2.2.1) ζ *is applicable at index* k *if* $\zeta \in R^{\mathsf{O}}_{\Rightarrow}[l, k]$ *or*
ζ *is applicable for* $+\partial_{\mathsf{O}} l$ *if* $\zeta \in R^{\mathsf{C}}_{\Rightarrow}[l]$*, and*
(3.2.2.2) $\zeta > \gamma$.

Definition 7 (Permission Proof Conditions)**.**

$+\partial_{\mathsf{P}} l$: *If* $P(n+1) = +\partial_{\mathsf{P}} l$ *then*
(1) $\mathsf{P}l \in F$, *or*
(2) $\neg\mathsf{P}l, \mathsf{O}{\sim}l \notin F$, *and*
(3) $+\partial_{\mathsf{O}} l \in P(1..n)$, *or*
(4) $\exists \beta \in R^{\mathsf{P}}_{\Rightarrow}[l] \cup R^{\mathsf{C}}_{\Rightarrow}[l]$ *s.t.*
(4.1) β *is applicable if* $\beta \in R^{\mathsf{P}}_{\Rightarrow}[l]$ *or*
β *is applicable for* $+\partial_{\mathsf{P}} l$ *if* $\beta \in R^{\mathsf{C}}_{\Rightarrow}[l]$, *and*
(4.2) $\forall \gamma \in R^{\mathsf{O}}[{\sim}l, j] \cup R^{\mathsf{C}}_{\Rightarrow}[{\sim}l]$ *either*
(4.2.1) γ *is discarded at index* j *if* $\gamma \in R^{\mathsf{O}}[{\sim}l, j]$, *or*
(4.2.2) $\exists \zeta \in R^{\mathsf{P}}[l] \cup R^{\mathsf{O}}[l, k] \cup R^{\mathsf{C}}_{\Rightarrow}[l]$ *s.t.*
(4.2.2.1) ζ *is applicable at index* k *if* $\zeta \in R^{\mathsf{O}}[l, k]$,
or for $+\partial_{\mathsf{P}} l$ *if* $\zeta \in R^{\mathsf{C}}_{\Rightarrow}[l]$ *and*
(4.2.2.2) $\zeta > \beta$.

The set of positive and negative conclusions of a theory is called *extension*. The extension of a theory is computed based on the literals that appear in it; more precisely, the literals in the Herbrand Base of the theory $HB(D) = \{l, {\sim}l \in \mathrm{PLit} \mid l \text{ appears in } D\}$.

Definition 8 (Extension)**.** *The* extension $E(D)$ *of a defeasible deontic theory* D *is*

$$E(D) = (+\partial_{\mathsf{C}}, -\partial_{\mathsf{C}}, +\partial_{\mathsf{O}}, -\partial_{\mathsf{O}}, +\partial_{\mathsf{P}}, -\partial_{\mathsf{P}}),$$

where $\pm\partial_{\Box} = \{l \in HB(D) \mid D \vdash \pm\partial_{\Box} l\}$, *with* $\Box \in \{\mathsf{C}, \mathsf{O}, \mathsf{P}\}$.

Theorem 1. *[See [5,9]] Given a defeasible theory* D, *its extension* $E(D)$ *can be computed in time polynomial to the size of the theory.*

4. Normative Explanation: The Formal Definition

As outlined in the previous sections, we explore the idea of stable deontic explanation by identifying those facts that ensure to prove a certain deontic conclusion. More precisely,

- Facts are added to an initial theory D_{init} and used to explain deontic conclusions in the resulting theory;
- We impose that only literals that do not appear as a consequence of any rule can be admissible facts for our purpose (factual literals).

The output theory (obtained by adding factual literals), as well as the whole operations, must thus satisfy certain properties.

Definition 9 (Admissible factual literals)**.** *Given (an initial) theory* $D_{init} = (\emptyset, R, >)$, *we define the set of* admissible factual literals *(shortly,* factual literals*) as*

$$\{a, \neg a, \mathsf{O}b, \mathsf{O}\neg b, \mathsf{P}c, \mathsf{P}\neg c \mid R^{\mathsf{C}}[a] \cup R^{\mathsf{C}}[\neg a] = \emptyset,$$
$$R^{\mathsf{O}}[b, i] \cup R^{\mathsf{C}}[b] \cup R^{\mathsf{O}}[\neg b, i] \cup R^{\mathsf{C}}[\neg b] = \emptyset$$
$$R^{\mathsf{O}}[c, i] \cup R^{\mathsf{P}}[c] \cup R^{\mathsf{C}}[c] \cup R^{\mathsf{O}}[\neg c, i] \cup R^{\mathsf{P}}[\neg c] \cup R^{\mathsf{C}}[\neg c] = \emptyset\}.$$

We say that an admissible factual literal l *is* deontic *iff* $l \in \mathrm{ModLit}$.

It follows that the set of factual literals is the set of literals for which there are no rules; consequently, such literals can only be derived if they are facts of the theory.

Definition 10 (Consistent set of literals)**.** *A set of literals is* consistent *if it does not contain any pair of literals* $(p, \neg p)$, $(\Box p, \Box\neg p)$, $(\Box p, \neg\Box p)$, $(\mathsf{O}p, \mathsf{P}{\sim}p)$, *or* $(\mathsf{O}p, \neg\mathsf{P}p)$.

Example 2. *Assume D_{init} is $(\emptyset, R, \emptyset)$, with*

$$R = \{\alpha\colon a \Rightarrow_{\mathsf{C}} z, \zeta\colon z \Rightarrow_{\mathsf{O}} l, \beta\colon \mathsf{O}b \Rightarrow_{\mathsf{P}} \neg z, \gamma\colon \neg\mathsf{P}z \Rightarrow_{\mathsf{O}} l \otimes c, \delta\colon d, \mathsf{P}z \Rightarrow_{\mathsf{O}} \neg c\}.$$

Here, literals such as a, d, $\mathsf{O}b$, l, and $\neg z$ are (admissible) factual literals, whilst z, $\mathsf{O}l$, $\mathsf{P}\neg z$, $\mathsf{O}\neg c$, and $\mathsf{O}c$ are not.

Secondly, the output theory must be stable, i.e., consistently adding facts does not change the provability of the target literal. To formalise when a theory is stable, we firstly define which characteristics the output theory must satisfy, and we name such "valid" output theories *normative cases*.

Definition 11 (Normative Case)**.** *Let theory $D_{init} = (\emptyset, R, >)$ be the initial theory and $l \in$ Lit $\cup$ ModLit be the target literal, we say that a theory $D = (F, R, >)$ is a* normative case for *l* of D_{init} *iff*

1. *F is consistent,*
2. *For all $f \in F$, f is a factual literal, and*
3. *$D \vdash +\partial_{\Box} p$ if $l = \Box p$ with $\Box \in \{\mathsf{O}, \mathsf{P}\}$, or $D \vdash +\partial_{\mathsf{C}} p$ if $l = p$.*

Definition 12 (Stable Normative Case)**.** *Given the initial theory $D_{init} = (\emptyset, R, >)$ and the target l, we say that a theory $D = (F, R, >)$ is*

1. *A* stable normative case for *l* of D_{init} *iff (1) D is a case for l (of D_{init}), and (2) for all $D' = (F', R, >)$ s.t. if $F \subsetneq F'$ and F' is consistent, then $D' \vdash +\partial_{\Box} p$ if $l = \Box p$, or $D' \vdash +\partial_{\mathsf{C}} p$ if $l = p$;*
2. *A* deontically stable normative case for *l* of D_{init} *iff (1) D is a case for l (of D_{init}), and (2) for all $D' = (F', R, >)$ s.t. if $F \subsetneq F'$, $F' \backslash F \subseteq$ ModLit, and F' is consistent, then $D' \vdash +\partial_{\Box} p$ if $l = \Box p$, or $D' \vdash +\partial_{\mathsf{C}} p$ if $l = p$.*

We thus observe that, in contrast with the non-normative case [4], two sub-types of normative cases can be distinguished:

- A general case where a certain conclusion follows whatever facts are added to the initial theory;
- A deontic case where a certain conclusion follows whatever deontic facts (such as $\mathsf{O}p$ or $\neg\mathsf{P}q$) are added to the initial theory, but it is not ensured the stability if non-deontic facts are added.

Example 3. *Consider the theory D_{init} as in Example 2. The case theory $D = (F = \{\mathsf{P}a\}, R, \emptyset)$ is not stable for $\mathsf{P}z$ as $D' = (F' = \{\mathsf{P}a, \mathsf{O}b\}, R, \emptyset)$ proves $-\partial_{\mathsf{P}} z$. On the contrary, theories where the set of facts is $\{\mathsf{P}a, \mathsf{O}b, \sim l\}$ are stable normative cases for $\mathsf{O}c$.*

Suppose we add the rule '$\epsilon\colon \neg\mathsf{P}z, e \Rightarrow_{\mathsf{P}} \sim c$'. Theories with set of facts $\{\mathsf{P}a, \mathsf{O}b, \sim l\}$ are deontically stable normative cases, but we can no longer hold that they are stable in general, since adding the non-deontic fact e would prevent deriving $\mathsf{O}c$.

Symmetric to the concept of case, the notion of *normative refutation case* is:

Definition 13 (Normative Refutation Case)**.** *Let $\Box \in \{\mathsf{O}, \mathsf{P}\}$. Given the initial theory $D_{init} = (\emptyset, R, >)$ and the target literal $l \in$ Lit $\cup$ ModLit, we say that a theory $D = (F, R, >)$ is a* normative refutation case for *l* of D_{init} *iff*

1. *F is consistent,*
2. *For all $f \in F$, f is a factual literal, and*

3. $D \vdash -\partial_{\Box} p$ *if* $l = \Box p$, *or* $D \vdash -\partial_{\mathsf{C}} p$ *if* $l = p$.

Definition 14 (Stable Normative Refutation Case)**.** *Given the initial theory* $D_{init} = (\emptyset, R, >)$ *and the target* l, *we say that a theory* $D = (F, R, >)$ *is*

1. *A* stable normative refutation case for l of D_{init} *iff (1)* D *is a normative refutation case for* l *(of* D_{init}*), and (2) for all* $D' = (F', R, >)$ *s.t. if* $F \subsetneq F'$ *and* F' *is consistent, then* $D' \vdash -\partial_{\Box} p$ *if* $l = \Box p$, *or* $D' \vdash -\partial_{\mathsf{C}} p$ *if* $l = p$;
2. *A* deontically stable normative refutation case for l of D_{init} *iff (1)* D *is a normative refutation case for* l *(of* D_{init}*), and (2) for all* $D' = (F', R, >)$ *s.t. if* $F \subsetneq F'$, $F' \backslash F \subseteq$ ModLit, *and* F' *is consistent, then* $D' \vdash -\partial_{\Box} p$ *if* $l = \Box p$, *or* $D' \vdash -\partial_{\mathsf{C}} p$ *if* $l = p$.

Clearly, the following result trivially holds.

Proposition 1. *For any theories* D_{init} *and* D, *if* D *is a stable (refutation) case for* l *of* D_{init}, *then* D *is a deontically stable (refutation) case for* l *of* D_{init}.

The notion of *unstable case* can be directly introduced, which is the situation when a case is not resilient to the addition of facts to the theory.

Definition 15 (Unstable Normative Case)**.** *Given the initial theory* $D_{init} = (\emptyset, R, >)$ *and the target* l, *we say that a theory* $D = (F, R, >)$ *is* (deontically) normative unstable for l of D_{init} *iff (1)* D *is a normative case for* l *(of* D_{init}*), and (2) there exists* $D' = (F', R, >)$ *s.t. if* $F \subsetneq F'$ *(if* $F \subsetneq F'$, $F' \backslash F \subseteq$ ModLit*) and* F' *is consistent, then* $D' \vdash +\partial_{\Box} {\sim} p$ *if* $l = \Box p$, *or* $D' \vdash +\partial_{\mathsf{C}} {\sim} p$ *if* $l = p$

Note that, naturally, D is "just" a case for l, and not a stable (refutation) case.

A final interesting property is to identify a stable normative explanation which optimises the degree of deontic compliance.

Definition 16 (Optimal Stable Explanation)**.** *Given a theory* D *and its extension*

$$E(D) = (+\partial_{\mathsf{C}}, -\partial_{\mathsf{C}}, +\partial_{\mathsf{O}}, -\partial_{\mathsf{O}}, +\partial_{\mathsf{P}}, -\partial_{\mathsf{P}}),$$

the degree of compliance $\mathbf{Degree}(D)$ of D *is* $|\mathbf{Compl}(D)|$ *where* $\mathbf{Compl}(D) = \{l | D \vdash +\partial_{\mathsf{O}} l$ *and* $D \vdash -\partial_{\mathsf{C}} {\sim} l\}$. *A theory* D *is an* optimal stable normative explanation *for a target literal* l *iff* D *is a stable normative case and there is no other stable normative case* D' *for* l *such that* $\mathbf{Degree}(D') \leq \mathbf{Degree}(D)$.

Example 4. *Consider again Example 2. A theory* D *with set of facts '*$F = \{\mathsf{P}a, \mathsf{O}b, {\sim} l\}$*' is an optimal stable normative cases for* $\mathsf{O}c$. *However, if we have in* R *also '*$\theta :\Rightarrow_{\mathsf{O}} c$*', such that* $\theta > \delta$, *then* D *is no longer optimal.*

5. Complexity Results

The problem of determining if a case is stable is intractable in standard propositional DL [4]. The same results hold for DDL and the proof already developed can be directly used here as well: it is enough to show that for each Defeasible Deontic Theory, an equivalent propositional Defeasible Theory can be defined. The transformation is based on the procedure given in [15] to reduce a Defeasible Deontic Theory into a conclusion equivalent Defeasible Theory.

Theorem 2. *[15, Theorem 40] There is a polynomial transformation from any theory in DDL into its counterpart in DL.*

This theorem allows us to extend the complexity results for stability in DL [4, Theorems 2, 3 and 4] to the case of DDL.

Theorem 3. *Given a Defeasible Theory and a case, the problem of determining if the case is stable is co-NP-complete.*

Theorem 4. *Given a Defeasible Theory and a refutation case, the problem of determining if the refutation case is stable is co-NP-complete.*

Theorem 5. *Given a Defeasible Theory and a case, the problem of determining if the case is unstable is NP-complete.*

6. Summary

We examined the notion of deontic stability. We used Defeasible Deontic Logic, a tractable computationally oriented logic for the formalisation of norms, to provide a formal definition of the stability problem. We proved that to determine if an extension of a case is stable is computationally intractable even when the underlying (legal) reasoning system is tractable. The result indicates that, in general, creating an automated question-answering system posing questions to a user to determine a legal status (e.g., to determine what set of facts warrants a given legal outcome) is not feasible without additional heuristics. Accordingly, having determined the complexity, we plan to investigate suitable heuristics and identify tractable instances of the stability problem.

References

[1] Peter Achinstein. *The Nature of Explanation*. Oxford University Press, Oxford, 1983.
[2] Grigoris Antoniou, David Billington, Guido Governatori, and Michael J. Maher. Representation results for defeasible logic. *ACM Trans. Comput. Log.*, 2(2):255–287, 2001.
[3] Katie Atkinson, Trevor Bench-Capon, and Danushka Bollegala. Explanation in ai and law: Past, present and future. *Artificial Intelligence*, 289:103387, 2020.
[4] Guido Governatori, Francesco Olivieri, Antonino Rotolo, and Matteo Cristani. Inference to the stable explanations. In *LPNMR 2022*, pages 245–258, Cham, 2022. Springer.
[5] Guido Governatori, Francesco Olivieri, Antonino Rotolo, and Simone Scannapieco. Computing strong and weak permissions in defeasible logic. *J. Philos. Log.*, 42(6):799–829, 2013.
[6] Guido Governatori, Francesco Olivieri, Simone Scannapieco, Antonino Rotolo, and Matteo Cristani. The rationale behind the concept of goal. *Theory Pract. Log. Program.*, 16(3):296–324, 2016.
[7] Guido Governatori, Vineet Padmanabhan, Antonino Rotolo, and Abdul Sattar. A defeasible logic for modelling policy-based intentions and motivational attitudes. *Log. J. IGPL*, 17(3):227–265, 2009.
[8] Guido Governatori and Antonino Rotolo. Logic of violations: A gentzen system for reasoning with contrary-to-duty obligations. *Australasian Journal of Logic*, 4:193–215, 2006.
[9] Guido Governatori and Antonino Rotolo. BIO logical agents: Norms, beliefs, intentions in defeasible logic. *Auton. Agents Multi Agent Syst.*, 17(1):36–69, 2008.
[10] Guido Governatori, Antonino Rotolo, and Giovanni Sartor. Logic and the law: Philosophical foundations, deontics, and defeasible reasoning. In *Handbook of Deontic Logic and Normative Systems*, volume 2, pages 657–764. College Publications, London, 2021.
[11] Masha Medvedeva, Michel Vols, and Martijn Wieling. Using machine learning to predict decisions of the european court of human rights. *Artificial Intelligence and Law*, 28:237–266, 2020.
[12] Tim Miller, Robert Hoffman, Ofra Amir, and Andreas Holzinger, editors. *Artificial Intelligence journal: Special issue on Explainable Artificial Intelligence (XAI)*, volume 307, 2022.
[13] Donald Nute. Defeasible reasoning. In *Proceedings of the Hawaii International Conference on System Science*, volume 3, pages 470–477, 1987.
[14] Joseph C. Pitt. *Theories of Explanation*. Oxford University Press, Oxford, 1988.
[15] Simone Scannapieco. *Towards a Methodology for Business Process Revision Under Norm and Outcome Compliance*. PhD thesis, Griffith University, Brisbane, Australia, 2014.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220448

Toward Automatically Identifying Legally Relevant Factors

Morgan GRAY [a,1], Jaromír ŠAVELKA [c] Wesley OLIVER [d] and Kevin ASHLEY [a,b]
[a] *Intelligent Systems Program, University of Pittsburgh, USA*
[b] *School of Law, University of Pittsburgh, USA*
[c] *School of Computer Science, Carnegie Mellon University, USA*
[d] *School of Law, Duquesne University, USA*

Abstract. In making legal decisions, courts apply relevant law to facts. While the law typically changes slowly over time, facts vary from case to case. Nevertheless, underlying patterns of fact may emerge. This research focuses on underlying fact patterns commonly present in cases where motorists are stopped for a traffic violation and subsequently detained while a police officer conducts a canine sniff of the vehicle for drugs. We present a set of underlying patterns of fact, that is, factors of suspicion, that police and courts apply in determining reasonable suspicion. We demonstrate how these fact patterns can be identified and annotated in legal cases and how these annotations can be employed to fine-tune a transformer model to identify the factors in previously unseen legal opinions.

Keywords. Automatic Text Identification, Multi-label Classification, Sentence Classification, Totality of the Circumstances Test

1. Introduction

We are investigating legal cases that assess the constitutionality of police decisions to search an automobile for drugs or to detain it for a canine search. Drug interdiction automobile stops have led to thousands of cases at the state and federal level, including the U.S. Supreme Court. Courts assess whether police had reasonable suspicion or probable cause to believe drugs are present in a car. With reasonable suspicion, an officer can briefly detain a motorist until a drug dog confirms or dispels the officer's suspicion. With probable cause, an officer may search the car. Officers can consider anything that enhances their suspicion of motorists, but officers frequently identify factors such as rental cars, strange travel plans, and the presence of strong air fresheners as increasing the likelihood that drugs will be found. These and others are referred to as "factors of suspicion." Officers must then decide whether they have probable cause to search the car or the lesser standard of reasonable suspicion to detain it until a drug dog arrives. If a search is conducted and something illegal is discovered in the search, only then will a judge determine whether there was adequate legal suspicion for detention.

[1]Corresponding Author: Morgan Gray, Learning Research and Development Center: 3420 Forbes Ave. Pittsburgh, PA 15260, USA; Email: mag454@pitt.edu

The open-ended nature of these legal standards, the enormous number of decisions, and the delay in judicial review cause uncertainty in deciding if there is reasonable suspicion in auto stop scenarios. Judges, lawyers, and police officers cannot read all 40,000 published decisions and determine how, courts, in the aggregate, view each suspicious factor an officer identified in a drug interdiction stop. Processing large amounts of text, however, is exactly what machines can do very well.

We hypothesize that a methodology for automatically assessing reasonable suspicion in drug interdiction stops is possible. We apply text analytic methods to auto stop cases in order to answer the following research question: Can a computer program apply ML/NLP to learn automatically to identify auto stop case outcomes and the factors of suspicion that courts consider in assessing the legality of the stops?

In this paper we report our initial progress in developing a type system of factors of suspicion, annotating auto stop cases in terms of factors, computing the level inter annotator agreement, and automatically identifying factors in case texts. Our ultimate goals are to compute the weight courts assign to each of the factors in isolation, as well as in combination with other factors and to assess the likelihood that a court will find facts sufficient for a search, thus providing a metric for assessing the legal merits of the police stops. This could lead to an empirically-based definition of reasonable suspicion and probable cause, provide insights about the efficacy of such searches, and inform new policies and procedures to improve the accuracy of traffic stops and lessen the influence of implicit bias and its effect on minority citizens.

2. Related Work

Factors are stereotypical patterns of facts that tend to strengthen or weaken a plaintiff's argument in favor of a legal claim. [1]. Since their introduction in the HYPO program [2], factors have become a staple knowledge representation technique in AI and Law. See [3]. Computational models of case-based legal argument such as [4], [5] and [6] now employ factors in modeling arguments with legal rules, cases, and underlying values. Researchers have also made some progress in automatically identifying applicable factors in the texts of legal decisions. [7] trained a machine learning approach with case summaries to identify trade secret misappropriation factors in other summaries. Wyner and Peters [8] employed an annotation pipeline to extract information related to factors in trade secret law from the full texts of cases. [9] trained a model to identify such factors in full texts of cases.

In a promising approach [10] Branting and colleagues trained a machine learning program called SCALE (semi-supervised case annotation for legal explanations) to label text excerpts in WIPO (World Intellectual Property Organization) domain name dispute cases by applicable legal issues and factors. The ultimate goal is for SCALE to employ the issue and factor labels in explaining its predictions of outcomes of new cases in terms of reasons that legal professionals would understand. Beyond connecting case sentences to reasons, issue and factor tags could connect the case to computational models of legal argument like those mentioned above. A text analytic program like SCALE could identify the applicable issues and factors, and a case-based model, equipped with a database of cases annotated by SCALE, could assist in explaining and testing its predictions.

We expect that our approach in this work will improve performance in learning to identify factors in a new and factually more diverse legal domain. We believe that

automatically identifying factors of suspicion in drug interdiction auto stop cases is more difficult than identifying factors in the WIPO domain name cases of [10]. For one thing, auto stop cases involve greater stylistic diversity than WIPO domain name arbitration cases. Auto stop decisions are written by judges, not arbitrators, in trial and appellate courts from jurisdictions across the country. Auto stop cases also likely involve more diverse factual scenarios than WIPO cases. Unlike Branting, et al. we apply a transformer model to identify the factors of suspicion in the auto stop case texts. The language model, RoBERTa, [11] has been pretrained on vocabulary from an extensive text corpus. The process of training the model on our corpus of auto stop cases then fine tunes the model.

3. Data

The raw data used consists of legal opinions collected from the Harvard Law School Case Law Access Project (HCAP. https://case.law/.) In order to collect this data two sets of search terms were used. The first was ("reasonable suspicion" and "canine"). The second added the phrase ("drug interdiction") to the first set of search terms. These searches returned roughly 2,500 cases, of which a legal expert selected 211 cases having confirmed that they dealt with the legal issue of interest. These cases were processed into text files containing relevant metadata and the raw text of the majority opinion, and the raw text of minority opinions if available. Of the cases retrieved, 70 cases were from federal courts, 141 were from state courts, accounting for 67[2] unique jurisdictions. Cases across different jurisdictions can be used in this task because the legal issue and surrounding case law is almost identical if not identical from jurisdiction to jurisdiction. There were 182 legal opinions drafted by an appeals court, and only 29 drafted by a trial court.

Based on those annotations from our corpus of 211 cases, we learned that 57% state decisions concluded that suspicion was present while 43% concluded that it was not. Of the federal courts, we learned that 77% of the decisions determined that suspicion was present and 23% determined that it was not. Thus, the overall percentage of suspicion found was 63%, with 37% finding that it was not present. These processed cases are the basis for the annotation task described in the next section.

3.1. Annotation

The targeted data include the sentences describing factors officers rely on in concluding reasonable suspicion exists. When a court assesses if reasonable suspicion is present, it assesses if all of the officer's observations, taken together, warrant reasonable suspicion.

3.1.1. Type System

The type system contains 19 factors of suspicion, shown in the Table 1 under five boldface headings **Occupant Appearance or Behavior**, **Occupant Status**, **Travel Plans**, **Vehicle**, and **Vehicle Status**.[3] Each factor is associated with a number indicating the *broad* category into which a factor falls and a letter.

[2]The two most frequent jurisdictions, each comprising 12 cases, were the State of Texas and the United States Court of Appeals for the Tenth Circuit. The top ten jurisdictions accounted for 90 of the 211 cases.

[3]The factors were determined by legal experts based on litigation experience and reading and analyzing hundreds of legal opinions.

Table 1. Factor Type System

1 Occupant Appearance or Behavior	**2 Occupant Status**
1A Furtive Movement	2E Motorist License
1B Physical Appearance of Nervousness	2F Driver Status
1C Nervous Behavior	2G Refused Consent
1D Suspicious or Inconsistent Answers	2H Legal Indications of Drug Use
	2I Motorist's Appearance Related to Drug Use
3 Travel Plans	**4 Vehicle**
3J Possible Drug Route	4L Expensive Vehicle
3K Unusual Travel Plans	4M Vehicle License Plate or Registration
	4N Unusual Vehicle Ownership
5 Vehicle Status	**6 Other Annotation Labels**
5O Indicia of Hard Travel	6T Other
5P Masking Agent	6U Possibly Off Point
5Q Vehicle Contents Suggest Drugs	6V Suspicion Found? - No
5R Suspicious Communication Device	6W Suspicion Found? - Yes
5S Suspicious Storage	

Each category encompasses factors related to a particular topic. For example, Category 2 includes factors related to the status of the vehicle's occupant. Factor 2H, *Legal Indications of Drug Use* is an appropriate label for sentences describing a situation where a motorist has a prior conviction for a drug offense, or has an active warrant for a suspected drug offense. Thus, although the **Occupant Status** factors describe different facts, they all concern legal indications that an individual may be a drug user.

Category 6, **Other Annotation Labels** contains options used to annotate other important aspects of a legal opinion. The *Other* category should only be used if an annotator believes that sentence, which clearly describes a factor the court found to be suspicious, does not *reasonably fit into any defined type, i.e., types 1A-5S*. Annotators can use the *Possibly Off Point* category to indicate that the case was not relevant to the legal issue of whether reasonable suspicion existed to extend a traffic stop. The *Suspicion Found? - No/Yes* categories are to be used to annotate the legal conclusion reached by the court as to whether reasonable suspicion existed.

3.1.2. Annotation Task

We hired 6 law students and 1 recent law school graduate to annotate each factor in the corpus of 211 cases. All of the annotators had undergone more than one year of formal legal education at an accredited law school.

The annotators received 10–15 hours of training. They were introduced to the legal problem in some depth. We introduced them to the type system outlined in Table 1 with a lecture and a detailed discussion. A *Factor Glossary* provided specific descriptions and examples of each factor type. We introduced the annotators to *Gloss*, a convenient online annotation environment developed by Jaromir Savelka that supports annotating sentences by color-coded highlighting. [12] In addition to the *Factor Glossary*, the annotators employed an *Annotation Guideline* containing detailed instructions on how to annotate cases. The Guideline provides specific instructions as to the spans of text to (or not to) be annotated and the labels to be applied. Working in a group, the annotators worked

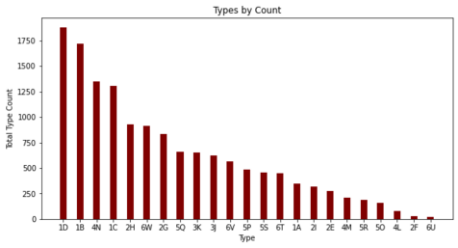


Figure 1. All Annotations: Counts by Type

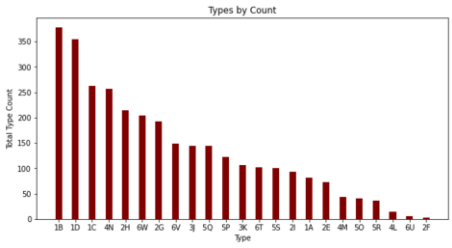


Figure 2. Gold Annotations: Counts by Type

through an initial case with the aid of a legal expert. The annotators then individually annotated five cases. The legal expert reviewed the annotations, corrected mistakes, and clarified points of misunderstanding. This continued for one more session with expert feedback before the students began annotating on their own.

All cases in the data set were annotated by at least two annotators. For the first 100 cases, both annotators were required to "resolve" any conflicts in their annotations. If they could not agree on an annotation, both annotations were kept. Finally, a legal professional reviewed all annotations and made his own judgment of how any disputed sentences should be labeled. The result served as the gold standard for the training data.

3.1.3. Annotation Outcome

The total number of annotations (i.e., annotations from all cases and all annotators) in the data set is 14,434.[4] The bar chart in Figure 1 shows the total number of annotations in the data set by type. The training data in the experiments consisted of 3,121 annotations. The breakdown of these annotations by type is show in Figure 2. The histograms appear to have a similar shape, and the bar for each type remained relatively in a similar position. This indicates that the overall frequency of annotations and the frequency in the gold standard annotations used for training data are similar.

The co-occurence matrix, which accounts for all annotations, reveals some interesting features of the data. The strong co-occurence between 6W, suspicion found, and 2G, the motorist denied consent to search is noteworthy. Under U.S. law, it is illegal to consider a motorist's refusal to consent as a factor making it more likely that drugs are present. Although officers may not be expressly rely on a motorist's refusal, refusing consent shows up frequently when suspicion is found and infrequently when suspicion is not found.

We employed Cohen's κ [13] metric to measure the agreement between two annotators. We measured inter-annotator agreement on all cases in the data set. The mean coefficient for the first 100 annotated cases was 0.544. The mean coefficient for the second 100 annotated cases was 0.601. The overall mean was 0.57. These scores indicate moderate agreement according to [14]. From these scores it appears that annotators improved with repetition and correction. Importantly, sentences labeled by the same factor typically do *not* describe the same facts. Different facts may fall into the same category, and similar facts may be described differently. Given the complexity of factor identification, moderate agreement seems reasonable.

[4] After annotations were assessed, cases that were to be determined to be off point were removed from the data set. Only five cases were removed on these or similar grounds.

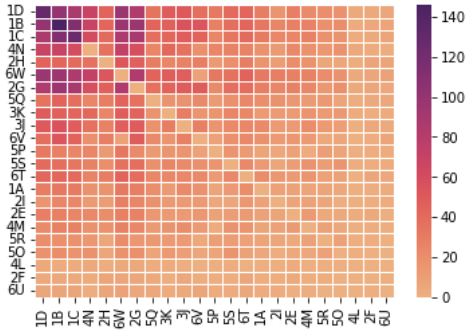


Figure 3. Co-occurrence Matrix by Type

4. Experiments

To assess whether automatic identification of factors in full text is feasible with this annotation scheme, we employed a pre-trained and subsequently fine-tuned language model. We classified the annotated types in order to distinguish among sentences that describe a factor, a conclusion, and non-typed sentences. Factor-related sentences were characterized by the applicable factors. Approximately 125,000 sentences in the data set were non-types and 14,000 cases in the data set represented a type. This is a multi-label classification problem in that sentences can describe more than one factor. For example:

> At the hearing, the officer testified that the reasons for the search were: Berry's nervousness, his uncertainty about whether his son was working or not, the fact that he was driving a rental car, the rental contract, Berry's looking down the interstate before answering some questions, Berry's failure to remember that his son lived in Decatur, the plastic garbage bag in the backseat, and the long trip from South Carolina only to stay a few hours.[5]

This sentence would be labeled in terms of the physical appearance of nervousness (1B), suspicious answers (1D), unusual vehicle ownership (4N), nervous behaviour (1C), suspicious storage (5S), and unusual travel plans (3K). Roughly 1,500 annotated sentences were annotated with one or more type. In order to deal with multi-labelled sentences, one-hot encoding was employed. The length of each encoding was 24, due to the 24 different possible types. An encoding representing one type was represented as follows:

[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

An encoding of the sentence shown above would look like this:

[0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]

The sentences were split into testing and training sets using a 60/40 training to testing split. The sentences and labels were then used to fine tune the roBERTa model over the course of 15 epochs, with evaluation occurring during training.

[5] Berry v. State, 547 S.E.2d 664 (Ga. Ct. App. 2001)

Table 2. Classification Report: Multilabel Classification

	precision	recall	f1-score	support
no_type	0.99	0.99	0.99	8381
1B Physical Appearance of Nervousness	0.92	0.89	0.90	62
2H Legal Indications of Drug Use	0.89	0.78	0.83	54
4N Unusual Vehicle Ownership	0.82	0.78	0.80	51
2G Refused Consent	0.79	0.73	0.76	30
3J Possible Drug Route	0.90	0.75	0.82	24
6W Suspicion Found? - Yes	0.92	0.83	0.88	42
5P Masking Agent	0.80	0.84	0.82	19
1C Nervous Behavior	0.78	0.85	0.81	53
3K Unusual Travel Plans	0.82	0.64	0.72	14
1D Suspicious or Inconsistent Answers	0.93	0.72	0.81	75
5Q Vehicle Contents Suggest Drugs	0.79	0.79	0.79	34
4M Vehicle License Plate or Registration	0.67	0.25	0.36	8
1A Furtive Movement	0.79	0.58	0.67	19
6V Suspicion Found? - No	0.71	0.87	0.78	31
6U Possibly Off Point	0.00	0.00	0.00	1
6T Other	0.79	0.58	0.67	19
2E Motorist License or Identification	1.00	0.91	0.95	11
2F Driver Status	0.00	0.00	0.00	0
5R Suspicious Communication Device	0.00	0.00	0.00	3
5S Suspicious Storage	0.73	0.84	0.78	19
5O Indicia of Hard Travel	0.00	0.00	0.00	6
2I Motorist's Appearance Related to Drug Use or Sale	0.92	0.85	0.88	13
4L Expensive Vehicle	0.00	0.00	0.00	3
micro avg	0.98	0.98	0.98	8972
macro avg	0.66	0.60	0.63	8972
weighted avg	0.98	0.98	0.98	8972
samples avg	0.92	0.92	0.92	8972

5. Results

The results of these experiments are shown in Table 2. Where the training data contained more than 10 instances of a type, the f-1 scores for classifying typed sentences ranged from 0.67 for factor 1A, Furtive Movement to 0.90 for 1B, Physical Appearance of Nervousness and 0.95 for 2E, Motorist License or Identification. Types that occurred in the test set fewer than eight times were not predicted or posted very low f-1 scores.

5.1. Discussion and Error Analysis

The results provide promising evidence that automatically identifying and extracting the relevant factors in legal opinions is feasible. On a number of occasions the classifier correctly identified a sentence that had been mislabelled as 'no type':

> "I don't have to let you search."

> "In determining the legality of a stop, courts do not attempt to divine the arresting officer's actual subjective motivation for making the stop; rather, they consider from an objective standpoint whether, given all of the circumstances, the officer had a reasonable and articulable suspicion of wrongdoing."

The first sentence in the above quote is describing the factual situation where an individual is refusing consent to search. Although the human annotators did not label it, the classifier correctly assigned the label of Refused Consent (2G). The second sentence describes a court's conclusion that there was not reasonable suspicion. Again, the human annotators missed it, but the classifier correctly assigned the appropriate label.

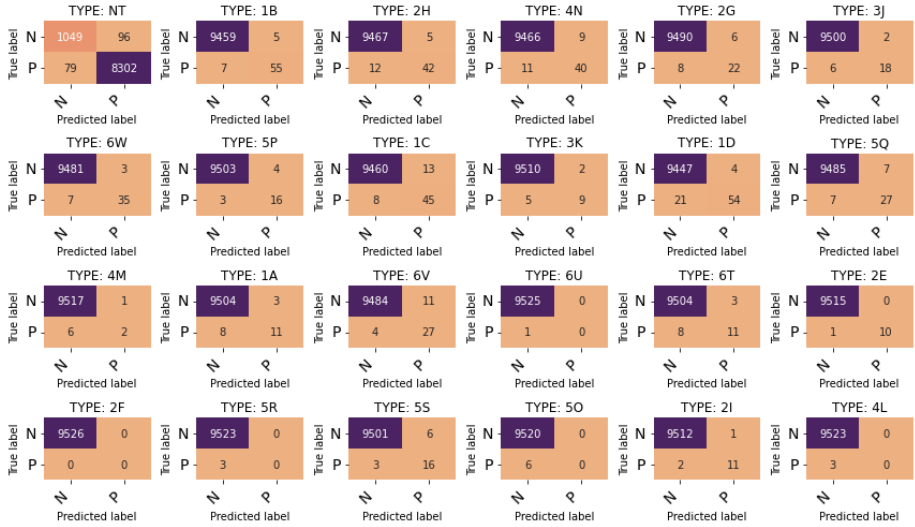


Figure 4. Confusion Matrix by Type

Whether precision and recall will be high enough for the intended downstream tasks is another question. As noted, we aim to compute weights for the factors across a large number of cases; high f-1 scores will be important.

In order to understand how we might improve performances, we undertook an error analysis. The confusion matrices are shown in Figure 4. For each matrix, the upper left quadrant reflects true negatives, and the bottom right represents true positives. For those factors that seemed to have higher numbers of errors, we examined some of the mislabeled instances to see if we could explain the errors.

The annotation guidelines require annotators to mark up only those sentences that indicate that a factor is present in the case. They were instructed *not* to annotate similar language in the opinion where the court discusses the legal significance of a factor generally or describes factors in other cases as courts may do in drawing comparisons across cases. Take, for example, these two sentences:

> Finally, Deputy Trammel testified that Mr. Powell appeared "excessively nervous" and remained so throughout the entire encounter, even when the deputy returned to Mr. Powell's vehicle to let him know the deputy was only giving him a warning citation. . . .
>
> Of course, even law-abiding citizens exhibit signs of nervousness when confronted by a law enforcement officer, and we have repeatedly held that "nervousness is of limited significance in determining reasonable suspicion."

Semantically, these sentences are similar. The first sentence should be annotated because it describes a fact present in the case at hand. The second sentence should not be annotated because the court is describing the legal treatment of nervousness as a factor. Given the semantic similarity of the sentences, the classifier would likely treat them the same. During one of the runs of the experiment described above, the classifier erroneously treated the first sentence as not belonging to a type.

Ignoring sentences that appear near legal citations may be a feasible path to eliminate at least some sentences that may confuse the classifier. As mentioned, some sentences describing a factor in another case may appear in a case at issue, for instance where a judge draws an analogy to or distinction from another case. Due to the nature of legal case citation rules, these semantically similar sentences will appear near case citations. It should be possible to filter sentences appearing close to a legal citation.

Another problem is the enormous variety of ways in which the facts characterized by a factor are expressed in the cases. The confusion matrix for Type 1D, Suspicious or Inconsistent Answers, shows a many false negatives. Upon inspecting some of these classification errors, we observed the following sentence:

> "While Anguiano attempts to highlight some of the consistencies in the men's stories, any general consistency cannot serve to dispel the contradictions of basic details, such as the name of the car's owner and the person for whom they were looking."

The sentence was properly labeled as a suspicious answer, however, the classifier predicted it as a not belonging to a type. Given the complexity of this example, we suspect that the high number of sentences properly labeled as 1D but misclassified by the model as 'no type' may be an example of this problem.

Another sentence describes a legal indication of drug use, factor 2H: having been convicted of a drug crime in the past.

> "Coupled with his observations of the items which, based on his training and experience, indicated narcotics trafficking, Officer MacMurdo was also very familiar with Defendant's prior criminal history and knew that Defendant was previously found in possession of drugs and a firearm."

The model erroneously predicted 'no type'. The example illustrates how differently the concept 'having been convicted of a drug crime in the past' can be expressed. We believe that increasing the quantity of annotated training data will assist the model to learn to correctly classify instances of factors like these.

Type 3J, Possible Drug Route, also had a relatively high number of false negatives. In the following sentence, we observed a different classification error also related to the variety of ways to express similar facts:

> "As he testified, it is commonly used to transport contraband throughout this state."

The sentence described a route known for drug trafficking, but the classifier predicted 'no type' when it should have predicted an instance of factor 3J. The anaphoric reference to 'it', a 'route' mentioned only in a previous sentence, probably interferes with a correct classification. Ambiguous pronoun references are likely to remain problematic.

We do aim to fix another common error in the data caused by problems with our sentence splitting algorithm [15]. It sometimes failed to correctly identify an annotated sentence. Essentially, the algorithm "missed" annotation, treating it as unannotated text. In future iterations we will improve the splitting algorithm to catch more annotations.

6. Conclusion

The results provide evidence that a program can learn to automatically identify factors in a new and factually more diverse legal domain, drug interdiction auto stop cases. In

future iterations of this work we intend to move from an off-the-shelf classifier to a model that has been specifically tuned for this task, for example by employing Legal-BERT, which has been pretrained on legal vocabulary from a large case law corpus. See [16]. Since the frequency of types appears to dramatically affect performance, we will collect more annotated data. For example, we are exploring how to integrate the annotation activities into pedagogical activities so that students can learn skills of close reading and legal argumentation as they annotate cases. Where we cannot increase the annotated data sufficiently, we will explore generating synthetic training examples for low frequency categories using resampling or similar techniques.

References

[1] Ashley KD. Artificial Intelligence and Legal Analytics. Cambridge U. Press; 2017.
[2] Ashley KD. Modeling Legal Arguments: Reasoning with Cases and Hypotheticals. MIT Press; 1990.
[3] Bench-Capon T. HYPO's legacy: introduction to the virtual special issue. Artificial Intelligence and Law. 2017;25(1):205-50.
[4] Grabmair M. Modeling Purposive Legal Argumentation & Case Outcome Prediction using Argument Schemes in the Value Judgment Formalism. U. Pitt.; 2016.
[5] Chorley A, Bench-Capon T. An empirical investigation of reasoning with legal cases through theory construction and application. AI and Law. 2005;13(3):323-71.
[6] Chorley A, Bench-Capon T. AGATHA: Using heuristic search to automate the construction of case law theories. Artificial Intelligence and Law. 2005;13(1):9-51.
[7] Ashley KD, Brüninghaus S. Automatically classifying case texts and predicting outcomes. Artificial Intelligence and Law. 2009;17(2):125-65.
[8] Wyner A, Peters W. Towards annotating and extracting textual legal case factors. In: SPLeT-2012; 2010. p. 36-45.
[9] Falakmasir M, Ashley K. Utilizing Vector Space Models for Identifying Legal Factors from Text. In: JURIX 2017. vol. 302. IOS Press; 2017. p. 183-92.
[10] Branting LK, Pfeifer C, Brown B, Ferro L, Aberdeen J, Weiss B, et al. Scalable and explainable legal prediction. Artificial Intelligence and Law. 2021;29(2):213-38.
[11] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:190711692. 2019.
[12] Savelka J, Ashley KD. Segmenting US Court Decisions into Functional and Issue Specific Parts. In: JURIX; 2018. p. 111-20.
[13] Cohen J. A coefficient of agreement for nominal scales. Educational and psychological measurement. 1960;20(1):37-46.
[14] Landis JR, Koch GG. The measurement of observer agreement for categorical data. biometrics. 1977;33(1):159-74.
[15] Savelka J, Walker VR, Grabmair M, Ashley KD. Sentence boundary detection in adjudicatory decisions in the united states. Traitement automatique des langues. 2017;58:21.
[16] Zheng L, Guha N, Anderson B, Henderson P, Ho D. When does pretraining help? assessing self-supervised learning for law and the CaseHOLD dataset of 53,000+ legal holdings. In: Proc. 18th Int'l Conf. on AI and Law; 2021. p. 159-68.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220449

Semantic Querying of Knowledge Rich Legal Digital Libraries Using Prism[1]

Hasan JAMIL [a,2]
[a] *Department of Computer Science, University of Idaho, USA*
ORCiD ID: Hasan Jamil https://orcid.org/0000-0002-3124-3780

Abstract. Contemporary legal digital libraries such as Lexis Nexis and WestLaw allow users to search case laws using sophisticated search tools. At its core, various forms of keyword search and indexing are used to find documents of interest. While newer search engines leveraging semantic technologies such as knowledgebases, natural language processing, and knowledge graphs are becoming available, legal databases are yet to take advantage of them fully. In this paper, we introduce an experimental legal document search engine, called *Prism*, that is capable of supporting legal argument based search to support legal claims.

Keywords. Document Engineering, Story Understanding, Premise Graph, Knowledge Graphs, Graph Matching, Natural Language Processing.

1. Introduction

Search is a basic function supported in all digital archives of information. While search techniques have evolved in structured and unstructured databases, it is still an ongoing research issue in digital libraries and document databases [10], in stark contrast with other types of digital libraries such as music [6], mathematics [14], judicial [4], etc. in terms of techniques and applications. Among the search techniques currently in use, some form of keyword search [17] or text mining based association search [1] are prevalent. In recent research, however, an emerging trend of searching digital libraries using knowledge graphs (KG) is gaining popularity [8] with the goal to improve semantic matching [9].

It is not uncommon in digital library search for users to land on useful documents almost by accident [13]. This is because most of the search engines do not allow queries that make sense semantically. For example, the following legal query

> Q_1: *"List case laws where parents retained jurisdiction in Virginia despite the opposing parent having the home state jurisdiction in another state under UCCJEA."*

is unlikely to return any case laws that meet the exact legal criteria expressed in the query. Most likely a keyword search will return all Michigan cases under UCCJEA mentioning

[1]This publication was partially made possible by an Institutional Development Award (IDeA) from the National Institute of General Medical Sciences of the National Institutes of Health under Grant #P20GM103408.
[2]Corresponding Author: Hasan Jamil, Department of Computer Science, University of Idaho, USA. Email: jamil@uidaho.edu.

home state and nothing much. To appreciate the complexities inherent in this query, it is important that we understand the structure of legal briefs, and the USA UCCJEA law.

1.1. Structure of Case Laws

Roughly, there are two types of laws – black letter law, or the articles of the constitution and case laws, or the adjudicated proceedings of the cases in the courts of law. Case laws are specific litigation in which black letter laws and other case laws, called legal precedents, are applied and the legal merits of opposing arguments are decided by the courts. In the USA, we have a three tier court system – 1) trial court where litigation first starts, merit is decided based on facts, and logic of the arguments by direct application of the laws; 2) appeals court where constitutionality of decisions made by trial courts is challenged and decided relative to the case at hand; and finally 3) the supreme court which adjudicates any misinterpretation of the laws by the lower courts.

With respect to digital representation of judicial documents, counsels of the parties involved in a litigation submit legal briefs, courts rule on the briefs, and produce another document called ruling or judgment. These rulings become legal precedents and enter the database as case laws. A ruling has roughly four parts: 1) court, party, counsel details, and case details such as case number, dates and jurisdiction. 2) a preamble that states the overall description of the litigation and applicable case laws. 3) facts that lay out the "truth" about the case as seen by each party which can be established by evidence. 4) legal argument why or why not the facts lead to legal conclusions supported by case laws. Finally, 5) relief sought or final opinion of the courts after deliberations and argument.

1.2. Article UCCJEA

UCCJEA stands for Uniform Child Custody Jurisdiction and Enforcement Act (we refer the readers to [3] for a detailed exposition to UCCJEA provisions in Virginia), which is an article of USA federal constitution dealing with jurisdiction of litigating parents or custodians of minor children residing in distinct court jurisdictions, often in multiple states. Some versions of the UCCJEA act has been adopted by all 50 states.

1.2.1. Essence of UCCJEA

Two of the main purposes of the article were to (i) stop competing jurisdictions from abusing their power and forcing families to needlessly waste resources in multiple states by usurping the jurisdiction from a state having a legitimate claim on the jurisdiction, and (ii) prevent parents from seeking a more convenient forum across state boundaries to make it difficult for the other parent to seek relief from a rightful jurisdiction or to frustrate them. The article does so in many ways but mainly by (i) removing the use of "best interest of the child" argument from the clauses of UCCJA (the predecessor of UCCJEA), (ii) prioritizing the jurisdictional bases in a tie proof hierarchy so that a state/court at a higher strata can assume jurisdiction, (iii) allowing a state at a higher strata exercise jurisdiction without any regard for a court in a competing jurisdiction at a lower strata in the absence of an exclusive continuing jurisdiction by another court, (this sounds confusing) (iv) requiring the courts to evaluate their jurisdictional authority anew (even if the court has exclusive continuing jurisdiction) every single time a new cause is brought before them by recognizing the fact that jurisdiction of any court is not

permanent and may change primarily due to the mobility of the child, and (v) giving the home state the absolute priority and preemptive jurisdiction over all courts again in the absence of exclusive continuing jurisdiction by another court.

1.3. Use Case

Abebi and Pierre had a child named Fiia when they divorced in a Mississippi court at which time they agreed to joint custody of Fiia. Subsequently, both Abebi and Pierre moved to Virginia and Michigan respectively, and Fiia moved to Michigan according to the terms of the Mississippi court order which granted each parent a two year primary rotational custodianship. However, upon moving to Virginia, Abebi filed for sole primary custody in Virginia and a jurisdictional litigation ensued involving three different states. The primary question being debated was which court has jurisdiction over Fiia, and where this custody matter will be decided.

On pleading with Virginia by Abebi, the court assumed jurisdiction despite objection that Michigan was Fiia's home state and Virginia did not have jurisdiction to make an initial determination. Furthermore, Mississippi still had exclusive continuing jurisdiction and did not decline to exercise jurisdiction. Michigan, on the other hand, deferred to Virginia stating "since" Virginia is exercising jurisdiction, it (MI) cannot despite having home state jurisdiction, and despite the UCCJEA stating that Michigan is not required to extend full faith and credit to Virginia court because they did not have the jurisdiction in the first place.

With the intention of appealing the two decisions in Virginia as well as Michigan courts, Pierre is looking for case laws that show precedent supporting Virginia's stance, and then researching if that erroneous decision was reversed by superior courts in Virginia . In fact, there are plenty of case laws that refute the Virginia and Michigan position in the case of Abebi v Pierre in favor of Pierre that existing legal search engines cannot find or link multiple case laws to offer a more complete picture.

For example, the Janet Miller-Jenkins v. Lisa Miller-Jenkins, Virginia (2006)[3] case is almost exactly identical to Abebi v. Pierre and supports Pierre's position, which was denied by both Virginia and Michigan, Janet and Lisa lived together in Virginia in the 1990's where Lisa gave birth to their daughter IMJ in April 2002. Soon after in August 2002 they moved to Vermont and entered into a civil union. Unfortunately, in September 2003, the parties ended their relationship and Lisa moved to Virginia with IMJ while Janet remained in Vermont.

Lisa asked a court in Vermont to dissolve their union in November 2003 and sought legal and physical custody of IMJ. In June 2004, Vermont issued a temporary order awarding Lisa primary custody. On July 1, 2004 Lisa filed for sole custody with sole parental rights in Virginia upon Virginia's affirmation of Marriage Affirmation Act. Upon learning the Virginia action, the Vermont court on July 19, 2004, exercised its exclusive continuing jurisdiction, stating that it will not defer to Virginia and ordered that its previous custody order be followed. When the Virginia courts proceeded with the litigation in Virginia, the court of Vermont forcefully ignored all Virginia orders holding that Virginia lacked subject matter jurisdiction and retained its (Vermont's) right to exercise jurisdiction. The Vermont Supreme court held that the state acted according to its established law, had jurisdiction to do so and that Parental Kidnapping Prevention Act (PKPA)

[3] http://www.courts.state.va.us/opinions/opncavwp/2654044.pdf

afforded preemptive jurisdiction to Vermont and denied full faith and credit to Virginia orders that contradicted those entered by the Vermont court. On appeal in Virginia, the Virginia court of appeals affirmed and upheld Vermont's position.

Both Vermont and Virginia appeals court's positions have been affirmed in several other similar courts such as in Rogers v. Rogers, Alaska (1995)[4], Swalef v. Anderson, Virginia (2007)[5], Key v. Key, Virginia (2004)[6], and in numerous other cases. In particular, in Markle v. Markle, Michigan (2007)[7], the Michigan court of appeals denied to extend full faith and credit to Texas court's custody order citing Texas court's lack of subject matter jurisdiction. In Johnson v. Johnson, Michigan (2005)[8], the Michigan court of appeals reversed the Michigan trial court's order that denied Michigan jurisdiction in favor of Idaho without determining that Michigan was an inconvenient forum by simply determining that Michigan lacked jurisdiction under a scenario similar to that of Abebi v. Pierre case even though Michigan was the home state.

The critical point is that none of these cases were systematically discovered using existing search engines in law libraries; rather, they were accidentally discovered [13] on the internet by Pierre. The research issue we are addressing is designing a search engine to help find cases that support or refute the position of a plaintiff or defendant given the case facts as the user sees it. We call the reasoning a user uses to establish a claim a *premise graph*. The task the search engine assumes is to find the cases that at least partially match the edges in the premise graph, and possibly all the edges to render a conclusion. We lay out our experimental model in the sections to follow.

2. Document Understanding using *Prism*

Abebi v. Pierre illustrates a complex system of information structure that most likely will not lend itself to traditional query engines such as keyword search, layered indexing, and other techniques discussed earlier, to produce the documents these litigants seek. More novel approaches based on knowledge graphs [5, 7, 11] or knowledge driven querying of digital documents [2, 12] were not shown to be effective in the type of search we are interested in. We therefore propose a document authoring and engineering model to enrich legal documents with meta-information at creation time so that improved semantic search becomes possible. Our goal is to make the enrichment steps as user transparent as possible.

A careful examination of the UCCJEA black letter law suggests a premise-conclusion relationship in the form a logic structure $\alpha \leftarrow \beta_1, \wedge, \ldots, \wedge\beta_m$, where β_is are the conjuncts in the antecedent and α is the consequent of a logical implication. For example, for the following facts,

resident(pierre,fiia,michigan).
jurisdiction(Cust,Subject,State,homestate) ← resident(Cust,Subject,State),
    ¬ jurisdiction(Cust,Subject,State,exclusivecontinuing).
jurisdiction(Cust,Subject,State, homestate) ← resident(Cust,Subject,State),

[4] http://touchngo.com/sp/html/sp-4293.htm
[5] http://www.courts.state.va.us/opinions/opncavwp/2510061.pdf
[6] http://www.courts.state.va.us/opinions/opncavwp/1079041.pdf
[7] https://www.michbar.org/opinions/appeals/2007/081407/36789.pdf
[8] http://www.michbar.org/opinions/appeals/2005/030105/26467.pdf

jurisdiction(Cust,Subject,State,exclusivecontinuing),
declined(Cust,Subject,State,exclusivecontinuing).
jurisdiction(Cust,Subject,State, convenientforum) ←
resident(Cust,Subject,State), deferred(Cust,Subject,State,ExState)
jurisdiction(Cust,Subject,ExState,exclusivecontinuing).

the above rules codifying Home State jurisdiction under UCCJEA will determine that Pierre, as a custodian of Fiia, has home state jurisdiction in Michigan. However, if we add this fact to the database,

jurisdiction(pierre,fiia,mississippi,exclusivecontinuing).

Pierre will not gain home state jurisdiction in Michigan. This rule base then can act as a recommendation system to suggest Pierre to seek a convenient forum determination, or home state deferral by the state of Mississippi.

Some of the facts claimed in the legal briefs or pleadings are subject to dispute, and a ruling is necessary. For example, in Miller-Jenkins v. Miller-Jenkins, Virginia (2006), as well as in Abebi v. Pierre, Michigan (2009), both Lisa and Abebi claimed home state jurisdiction. In Lisa's case, home state was obvious since IMJ lived with Lisa in Virgina for more than six months. Lisa could not exercise the home state jurisdiction because Vermont was exercising its exclusive continuing jurisdiction, which takes precedent under UCCJEA. However, for Abebi, Fiia lived in Virginia for two weeks, after moving from Mississippi, and then lived with Pierre for more than four months at the time Abebi filed for custody. In such cases, both parties need to state why they believe their respective states have jurisdiction. A judge then decides the correct status based on case laws, which is clearly spelled out in the UCCJEA article. We can capture the premises for residency as the following set of rules.

resident(Cust,Subject,State) ← livedin(Cust,Subject,State,From,To),
duration(Days,From,To), filed(Date), Date=To, Days>183.
resident(Cust1,Subject,State1) ← livedin(Cust1,Subject,State1,From1,To1),
livedin(Cust2,Subject,State2,From2,To2), priorto(To2,From1),
duration(Days1,From1,To1), duration(Days2,From2,To2), filed(Date),
Date=To1, Days1>Days2.

The rules above say that on the date of filing the case, a custodian gains residency in a state if the child lived in that state six months or more continuously until the date of filing, or if the child lived in that state the most compared to the state she lived immediately prior. Note that both cannot simultaneously hold true. Now given the following facts, Pierre is certain to gain residency, i.e., home state residency.

livedin(abebi,fiia,virginia,1/1/2007,1/14/2007).
livedin(pierre,fiia,michigan,1/15/2007,5/20/2007).

The duration predicate above can be implemented as a computable function that will return the difference between two dates in number of days, and priorto as a Boolean function that returns true or false given two dates To and From if To is prior to From.

The technical issue now is, how do we arrive at these logical conclusions from a search of the available digital documents? One way to accomplish this is to design a text understanding system in ways similar to [7, 15] that is capable of deriving fact predicates, e.g., livedin or resident, from the case laws, and applying these rules to decide if a

document is relevant and meets the query conditions. In this approach, no additional manipulation of the documents will be necessary except the knowledge extraction engine. However, we can expect the search cost to be high because all documents will need to be understood and mined first for discovering the predicates. An alternative is to create these knowledge at the time of document authoring. We adopt the latter approach because it is efficient, even though slightly demanding and intrusive for users authoring the documents. We, however, contend that our document engineering approach is efficient for both creating documents, and processing queries.

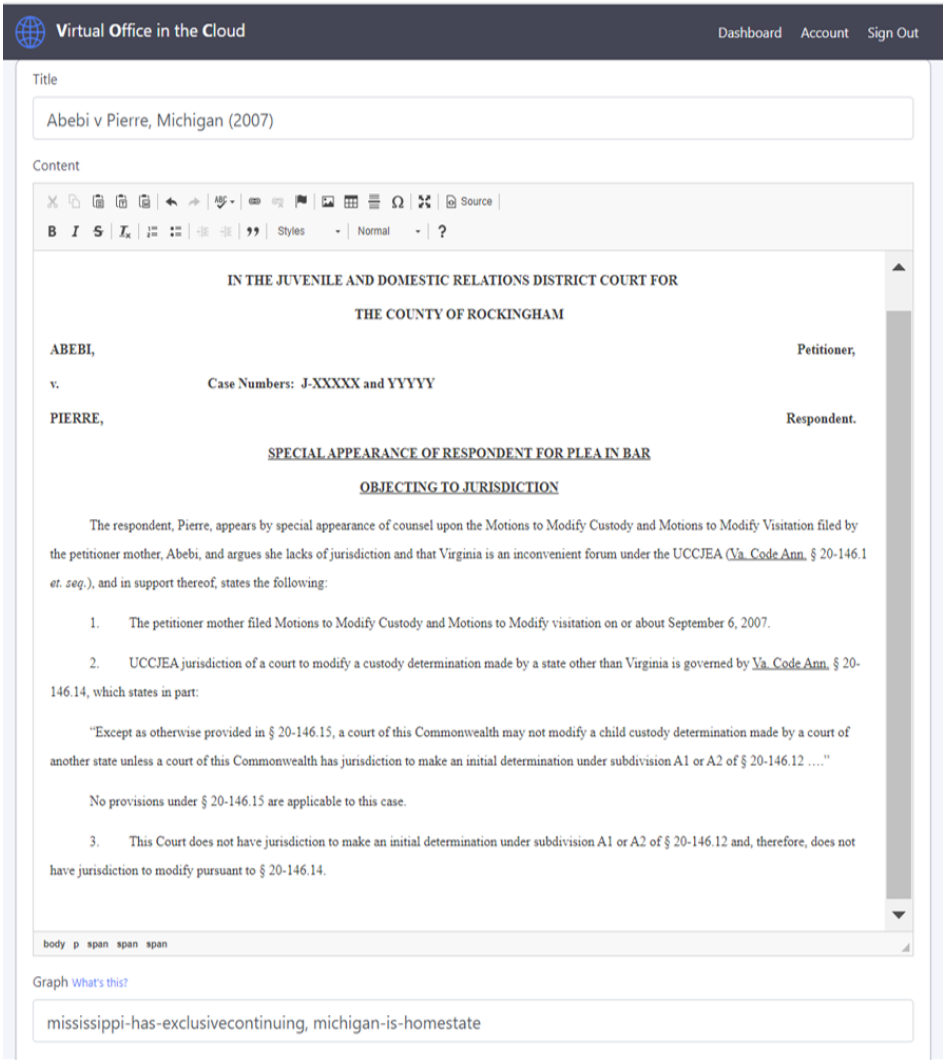


Figure 1. Prism user interface for document engineering with premise graph embedding.

The main idea is to design an HTML WYSIWYG legal document editor that will transparently embed a premise graph into the document as a searchable meta-data, which will not be rendered, yet the authors of the document will be able to view and edit it.

To help authors embed the graph, we design a type-ahead searchable legal terms such as *resident, exclusive continued jurisdiction, convenient forum*, etc. from which authors are able to pick node descriptions for a premise graph along with the required parameters. For example, they will be able to construct a node "Mississippi" has "exclusive continuing jurisdiction" over "Fiia" as a triple ⟨Fiia, Mississippi, Exclusive Continuing⟩ that we call *c-term*, or complex term. Subsequently, with a click of a mouse, this c-term can be added to an edge as a node, and stored as the document meta-data. Figure 1 shows the editor in use by the attorney of Pierre filing the objection to Abebi's attempt to retain jurisdiction in Virginia.

2.1. AND/OR Graphs

The major reasons question answering systems or legal search engines such as Lexis Nexis or WestLaw fail to respond to queries such as Q_1 is because they require causal reasoning or causality determination [16] which none of these contemporary digital libraries support. Since such causalities are application specific and orthogonal to document authoring, we believe they need to be addressed separately. Current approaches to such discoveries tend to be based on learning models, are quite involved and computationally expensive, in systems that support something of similar nature. In Prism, we seek to find a cheaper and more direct solution using the concept of directed AND/OR graphs that was exploited in past research [18].

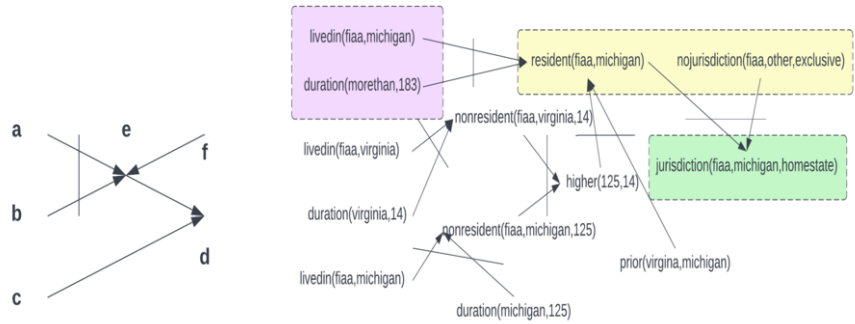
The process we have adopted to capture the premise graphs in Prism exploits the AND/OR graph representation. For example, the jurisdiction/4 rule[9] can be represented in the form of a modified AND/OR graph as shown in figure 2(b). In this modified AND/OR graph, the nodes are pre-processed and made grounded, and unlike the logic rules discussed in section 2, there are no variables. In other words, the rules are instantiated with ground facts. Users select these facts from a fixed set of terms which come with predefined slots to be filled in. For example, when the term resident is selected, the interface asks for two values, one, the name of the child, another, the state, and once supplied, generates the c-term. In order to support more complex premise graphs, Prism also allows expressing premise graphs in the form of RDF-like triples, node1-edgename-node2 type of edges, as shown in figure 1 with the document rendered and in 2(c) as the HTML document representation. Note that the premise graph is not visible to readers, yet remains visible to the author during editing.

Users are also able to visualize the premise graph before they save the document. Prism allows validation functions to check if the premise graph is legally valid, and semantically accurate. It reports mistakes using color coding of the edges. All semantically and legally accurate edges are shown in green, and the others in red. Edges being edited or not validated are shown in black.

2.2. Semantic Search using Premise Graphs

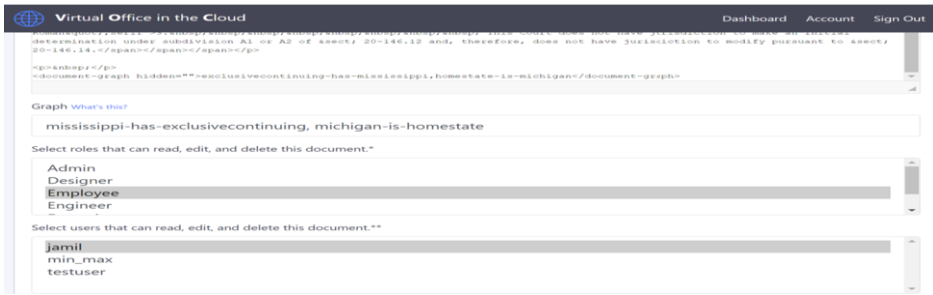
Usually, a counsel will try to find case laws that support even part of their claims. In other words, it is usually difficult to find case laws that have the exact circumstances that will warrant identical outcomes in the court of law. Given that a large percentage of cases are decided on erroneous premises and often get redressed in appeals or in supreme

[9] jurisdiction/4 means the predicate jurisdiction has four arguments.



(a) Causal net in the form of AND/OR graphs.

(b) jurisdiction/4 (home state) rules captured using AND/OR graph.



(c) Premise graphs are embedded in the document using a non-rendering mode.

Figure 2. Representing Premise graphs in Prism.

courts, it is not unlikely to find contradictory case laws. Therefore, case laws need to be interlinked so that the whole decision process is clear . Consequently, counsels can piece together their legal claims by citing cases that support parts of their arguments in the premise graphs, with the hope of finding such support for every part of their premise graph.

We, therefore, support a maximal constrained subgraph isomorphic matching search of the premise graphs of case laws in a systematic way. To understand the process, let us consider another case, Michelle v. Maxwell, 2002 (Nebraska) over the custody of Elli. In her case, let us assume that the case law contains the purple, yellow and green shaded parts of the premise graph shown in figure 2(b) with the following details: livedin(elli,nebraska), resident(elli,nebraska) and nojurisdiction(elli,other,exclusive) replacing the corresponding nodes in the premise graph. A search by Pierre's attorney with the entire premise graph in figure 2(b) which he intends to prove as his whole case, will match with Elli's case law since it supports "maximally" his argument that Michigan has jurisdiction over Fiia. This is because circumstances are identical to Fiia's with the only substitution being of Nebraska for Michigan, an isomorph.

On the other hand, if Prism can find another case law that supports the other branch of Pierre's premise graph, namely the non-shaded branches, Prism will include that case law as well, which only strengthens his argument even though one support is logically

sufficient. In reality, Prism will list all such matches. The important issue to note here is that Prism will also find partial matches. For example, consider a case in which Prism could only find support (matching) for the purple branch, and nothing else. In that event, Prism will list this case as a possible partial match if and only if it could not find any more case laws to support the yellow or the green shaded portions (i.e., it did not find Elli's case). This is called the maximal constrained subgraph isomorphic search – i.e., Prism always searches for maximum possible matches. Technically, Prism breaks down every AND and OR into individual subgraphs to match isomorphically, then constructs the maximal matches from the parts within the same document, and discards a match the moment a relatively more maximal match is found.

3. Implementation of Prism as a Virtual Office Environment in the Cloud

We have implemented Prism as a virtual legal office environment, called *VOiC*, in which document privacy and controlled sharing are top priorities. We have used Flask for its well-known support for web applications, using its two core components Werkzeug for web server functions and Jinja for HTML templating. Flask is extendable by virtue of its support for many extensions, and it also works with the majority of third-party Python libraries, which we have used as well. In addition to several other extensions of Flask, we have used the flask_ckeditor extension. CKEditor is an embeddable rich text editor with full support for HTML editing. This extension allows a core feature of Prism-HTML editing and embedding of graphs directly into documents. Bootstrap 4 open source front-end framework was used for creating platform-agnostic and responsive websites using its wide range of CSS styling options. For data management, SQLAlchemy was used to seamlessly convert data from a SQLite relational database into Python objects.

VOiC provides a comprehensive document management system – a Virtual Office in the Cloud. It consists of four main processes: storing, sharing, searching, and rendering. Storage includes the holding of users, roles, and documents in a relation database. Sharing relates to access control, and allowing users to access pertinent documents through their username and role. Searching serves users with a tool for data discovery with documents searchable by their search graph, title, and content. Rendering forms the front-end portion of VOiC. Together, these elements form a robust solution for document management in a virtual setting.

VOiC has a powerful role-based access control for document sharing. It also supports two search options – keyword or substring search, and graph search. On submitting the keywords, a SQLAlchemy query then executes and retrieves all documents in which the search query keywords are a substring in the title or content. The graph search uses the maximal isomorphic search as described earlier, and is thus a more powerful search. However, the graph search is substantially slower than text search.

4. Conclusion

Both VOiC and Prism are ongoing research projects to support experimentation on a new approach to document authoring, sharing and searching, and collecting enough usage data to understand the usefulness of this new digital office environment. We feel that the

approach and the technology can also be used across other scientific domains including ecology, computational biology, and network science to search for scholarly documents to discover interacting entities, such as cause-effect relationship in nature, gene regulatory networks, and so on. However, more research will be necessary to understand how the effectiveness of our system in other scientific disciplines.

References

[1] Y. Asiri. Short text mining for classifying educational objectives and outcomes. *Comput. Syst. Sci. Eng.*, 41(1):35–50, 2022.

[2] T. Aso, T. Amagasa, and H. Kitagawa. A method for searching documents using knowledge bases. In *iiWAS2021, Linz, Austria, 29 November 2021 - 1 December 2021*, pages 250–258. ACM, 2021.

[3] V. Constitution. Uniform child custody jurisdiction and enforcement act. https://law.lis.virginia.gov/vacode/title20/chapter7.1/. Accessed: 8/18/2022.

[4] M. de Lourdes da Silveira, B. A. Ribeiro-Neto, R. de Freitas Vale, and R. T. Assumpção. Vertical searching in juridical digital libraries. In F. Sebastiani, editor, *ECIR 2003, Pisa, Italy, April 14-16, 2003,*, volume 2633 of *LNCS*, pages 491–501. Springer, 2003.

[5] J. S. Dhani, R. Bhatt, B. Ganesan, P. Sirohi, and V. Bhatnagar. Similar cases recommendation using legal knowledge graphs. *CoRR*, abs/2107.04771, 2021.

[6] B. Duggan and B. O'Shea. Tunepal: Searching a digital library of traditional music scores. *OCLC Syst. Serv.*, 27(4):284–297, 2011.

[7] E. Filtz. Building and processing a knowledge-graph for legal data. In *ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part II*, volume 10250 of *LNCS*, pages 184–194, 2017.

[8] G. Heidari, A. Ramadan, M. Stocker, and S. Auer. Leveraging a federation of knowledge graphs to improve faceted search in digital libraries. In *TPDL 2021, Virtual Event, September 13-17, 2021*, volume 12866 of *LNCS*, pages 141–152. Springer, 2021.

[9] T. T. Huynh, N. V. Do, T. N. Pham, and N. T. Tran. A semantic document retrieval system with semantic search technique based on knowledge base and graph representation. In *SoMeT 2018, Granada, Spain, 26-28 September 2018*, volume 303 of *Frontiers in Artificial Intelligence and Applications*, pages 870–882. IOS Press, 2018.

[10] P. G. Ipeirotis. Searching digital libraries. In L. Liu and M. T. Özsu, editors, *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

[11] A. C. Junior, F. Orlandi, D. Graux, M. Hossari, D. O'Sullivan, C. Hartz, and C. Dirschl. Knowledge graph-based legal search over german court cases. In *ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 - June 4, 2020*, volume 12124 of *LNCS*, pages 293–297. Springer, 2020.

[12] I. Kollia, K. Rapantzikos, G. B. Stamou, and A. Stafylopatis. Semantic query answering in digital libraries. In *SETN 2012, Lamia, Greece, May 28-31, 2012*, volume 7297 of *LNCS*, pages 17–24, 2012.

[13] S. W. Kumpulainen and H. Kautonen. Accidentally successful searching: Users' perceptions of a digital library. In *CHIIR 2017, Oslo, Norway, March 7-11, 2017*, pages 257–260. ACM, 2017.

[14] A. Oviedo, N. Kasioumis, and K. Aberer. $5e^{\{x+y\}}$: Searching over mathematical content in digital libraries. In *ACM/IEEE-CE Joint Conference on Digital Libraries, Knoxville, TN, USA, June 21-25, 2015*, pages 283–284. ACM, 2015.

[15] F. Sovrano, M. Palmirani, and F. Vitali. Legal knowledge extraction for knowledge graph based question-answering. In *JURIX 2020, Brno, Czech Republic, December 9-11, 2020*, volume 334 of *Frontiers in Artificial Intelligence and Applications*, pages 143–153. IOS Press, 2020.

[16] P. Thagard. Causal inference in legal decision making: Explanatory coherence vs. bayesian networks. *Appl. Artif. Intell.*, 18(3-4):231–249, 2004.

[17] V. Yellepeddi, P. Manimegalai, and S. B. Suvanam. Accurate approach towards efficiency of searching agents in digital libraries using keywords. *J. Medical Syst.*, 43(6):164:1–164:6, 2019.

[18] H. Yu, Q. Zhou, and M. Liu. A dynamic composite web services selection method with qos-aware based on AND/OR graph. *Int. J. Comput. Intell. Syst.*, 7(4):660–675, 2014.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220450

Investigating Strategies for Clause Recommendation

Sagar JOSHI [a,1], Sumanth BALAJI [a,2], Jerrin THOMAS [a], Aparna GARIMELLA [b] and Vasudeva VARMA [a]
[a] *International Institute of Information technology, Hyderabad, India*
[b] *Adobe Research, India*
ORCiD ID: Sagar Joshi https://orcid.org/0000-0001-6095-9713, Sumanth Balaji https://orcid.org/0000-0002-5669-7519

Abstract. Clause recommendation is the problem of recommending a clause to a legal contract, given the context of the contract in question and the clause type to which the clause should belong. With not much prior work being done toward the generation of legal contracts, this problem was proposed as a first step toward the bigger problem of contract generation. As an open-ended text generation problem, the distinguishing characteristics of this problem lie in the nature of legal language as a sublanguage and the considerable similarity of textual content within the clauses of a specific type. This similarity aspect in legal clauses drives us to investigate the importance of similar contracts' representation for recommending clauses. In our work, we experiment with generating clauses for 15 commonly occurring clause types in contracts expanding upon the previous work on this problem and analyzing clause recommendations in varying settings using information derived from similar contracts.

Keywords. Clause recommendation, Legal contracts, Legal NLP

1. Introduction

AI-driven assistance in drafting legal contracts as a tool can greatly benefit small and medium-sized enterprises. These enterprises often have limited legal support in contractual requirements compared to their large-scale counterparts that can afford sophisticated support from large legal teams. Contracts, being a type of legal document, can be characterized as being composed of a series of individual clauses or provisions, each capturing the legally binding rights, obligations, and agreements between the involved negotiating parties. As pointed out in [1], these clauses serve as the fundamental discourse units while drafting or reviewing contracts and encompass the legal essence of a contract.

The problem of clause recommendation [2] (ClauseRec) was proposed to assist contract drafting by recommending a clause to a contract. It was the first attempt at clause generation itself with the use of transformer-based techniques in NLP. For clause recom-

[1]Corresponding Author: Sagar Joshi; Email: sagar.joshi@resarch.iiit.ac.in
[2]Corresponding Author: Sumanth Balaji; Email: sumanth.balaji@resarch.iiit.ac.in. Both the corresponding authors have contributed equally to this work.

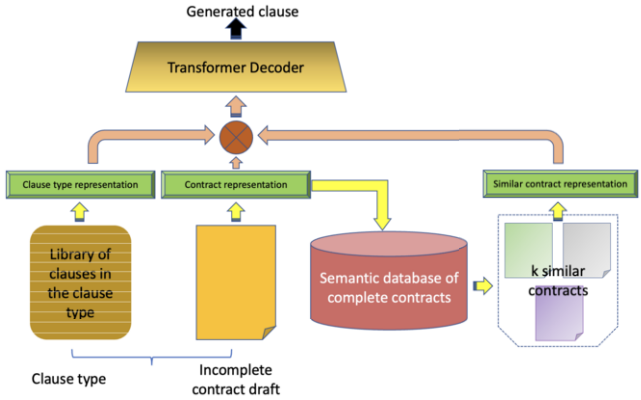


Figure 1. A pipeline for clause recommendation: We have an incomplete, in-draft contract at the input and the name of a clause type for which we need to recommend a new clause to the contract. We first compute the representation of the current contract. Since we know the clause type, we also compute a representation of the clause type based on a library of all the clauses under that type. Based on the representation of the current contract, we fetch k similar contracts and average them to obtain a similar contract representation. Now, making use of these three representations, we aggregate them before sending them to our transformer-based clause decoder to generate or recommend a new clause of the specified clause type to be added to the in-draft contract.

mendation, consider an incomplete contract in the draft to which the drafter of the contract wishes to add a new clause of a specific clause type. Clause recommendation serves to add such a clause based by taking into account the context of the current contract in the process. The work also introduced the problem of clause type relevance prediction to determine the relevance of a clause type to be added to the contract before proceeding with clause generation. However, the problem was modeled as a binary classification problem to determine whether or not a given clause type is relevant to the current contract. The pipeline for clause recommendation nevertheless makes use of the clause type information, hence serving the problem of clause type relevance as a secondary problem - extra to clause recommendation. However, it keeps opening the question of whether or not the clause type information should be taken from the user (i.e., the contract drafter).

In our work, we study the problem of clause recommendation by experimenting with several strategies for representing the context for clause generation and showing the significance of clause type information for clause generation. We improve the pipeline proposed by [2] by adding similar contract representations to the context. Figure 1 shows our best-performing pipeline for clause recommendation, which incorporates information from similar contracts in addition to the current contract and specified clause type. The generation is analyzed over 15 commonly different clause types in legal contracts. With research on legal clause generation currently in its infancy, we conclude by discussing the potential areas for improvement in clause generation. We open-source our code to ease future work on this problem[3].

[3] https://github.com/sagarsj42/strategies_for_clause_recommendation

2. Related Work

Previous work in legal contracts has focused on understanding tasks in legal NLP directed towards easier review and analysis of contracts. Identification of entities in contracts is looked upon in [3], [4], the latter introducing a dataset of 179 contracts specific to lease agreements. The dataset introduced by [4] also provides labeled data for identifying potentially unfair clauses, a task attempted prior by [5]. [6] identify critical contract clauses using a set of Context-Free Grammar (CFG) rules. The task of semantically retrieving legal clauses from a library of contracts as a span identification problem was introduced in [7]. An expert annotation dataset for contract review is provided in [8]. LEDGAR - a dataset for large-scale multilabel classification was introduced in [1]. The sheer size of this dataset and the variety of clause types covered makes its applicability much beyond the task for which it was introduced, an example being the current work for clause generation.

Since legal language is different from the general open domain language on which most of the models are pretrained, special attention has been paid to developing models to give representations specialized to the domain. LegalBERT [9] introduces a family of BERT-based models for application on downstream legal tasks. [2] train a further pretrained version of BERT for use in context building for clause recommendation and show the superiority of the trained embeddings by showing distinguishable representations based on the clause type. ALeaseBERT trained in [4] is another task-specific pretrained model on legal contracts. We make use of LegalBERT representations in our work.

3. Method

The problem of clause recommendation can be modeled as a controlled text generation problem $P(y|context)$ in which we generate a clause y given some contextual representation $context$. The $context$ representation can be modeled by making use of the inputs that can be made available from an incomplete version of the contract being drafted. Since a contract is essentially a collection of clauses, the representation of a contract can be obtained by averaging the representation of each *clause* in the contract. The average representation serves as the contract-specific contextual input. Thus for a contract having n_{clause} clauses, we can use a sequence encoder to get the representation of each clause before averaging to get the contract representation.

$$contract = \frac{\Sigma_{clause} Encoder(clause)}{n_{clause}} \tag{1}$$

The type of a legal clause is highly indicative of the clause since clauses under a single type are similar to each other. So, the clause type representation $clause\_type$ can be considered indicative in determining the characteristic text of the clause to be generated. For calculating this representation, we consider the library of all the clauses, N_{clause_type} occurring under the concerned clause type and compute an average of the individual clause embeddings.

$$clause\_type = \frac{\Sigma_{clause} Encoder(clause)}{N_{clause\_type}} \tag{2}$$

The *contract* representation can be used to retrieve k most similar contracts (*sim_contracts*) from an index of all the contract representations. Each of the retrieved k contracts can be used to provide additional context in two ways:

(1) $full\_sim\_contr$: Using the representations calculated for an entire contract and averaging over all k contracts

$$full\_sim\_contr = (\Sigma_{sim\_contract \in sim\_contracts} [\Sigma_{clause \in sim\_contract} \frac{\Sigma_{clause} Encoder(clause)}{n_{clause}}])/k \tag{3}$$

(2) *clause_sim_contr*: Using only the representation of the clauses of clause type t for which we need a clause recommendation. Here, we average the per-contract representations calculated for the clauses of the specified type before averaging over the k contracts.

$$clause\_sim\_contr = (\Sigma_{sim\_contract \in sim\_contracts} \left[\frac{\Sigma_{clause \in sim\_contract} \mathbb{1}\{clause\, \varepsilon\, t\} \cdot Encoder(clause)}{\Sigma_{clause \in sim\_contract} \mathbb{1}\{clause\, \varepsilon\, t\}}\right])/k \tag{4}$$

By making use of these representations, we compute the *context* vector in the following ways:

1. ONLY_CONTR: Using only the *contract* representation tries to predict the clause in a clause type agnostic setting, i.e., the model has no information of which particular clause type it has to recommend a clause, making the task more difficult since the model has to predict the topic as well as the underlying content, and thus results in much poorer performance as compared to clause type aware outputs.

$$context = contract \tag{5}$$

2. CONTR_TYPE: This was the methodology adopted in ClauseRec and that takes into account both - *contract* as well as $clause\_type$ representations.

$$context = (contract + clause\_type)/2 \tag{6}$$

3. CONTR_FULLSIM: In this experiment performed in a clause type agnostic setting, we complement the *contract* representation with the $full\_sim\_contr$ to see the effect of similar contract representation when the clause type is not known.

$$context = [contract\ ;\ full\_sim\_contr] \tag{7}$$

Here, $[.\,;\,.]$ indicates concatenation of the vectors.

4. CONTR_TYPE_FULLSIM: The original ClauseRec representation is augmented with full _sim _contr.

$$context = [(contract + clause\_type)/2\ ;\ full\_sim\_contr] \tag{8}$$

5. CONTR_TYPE_CLAUSESIM: Since we are aware of the clause type of clause to generate, *clause_sim_contr* is used of to see if more specific information can help generate better clauses.

$$context = [(contract + clause\_type)/2 \,;\, clause\_sim\_contr] \tag{9}$$

Once we have the strategy for computing the *context* vector, a language model with trainable parameters θ is trained to condition on this representation by minimizing the negative log-likelihood loss between the predicted and the expected output tokens.

$$l_{gen} = -log[\,p(y \,|\, context, \theta)\,] \tag{10}$$

4. Experimentation

This section explains our filtering and re-purposing of the LEDGAR dataset, experiments with encoder models, different values of k for similar contract retrieval, and metrics chosen for evaluation.

Dataset. We re-purpose the LEDGAR [1] dataset for multilabel clause type identification for our task. The dataset in its cleaned version consists of 60,540 contracts having 846,274 clauses from 12,608 different types. We filter the dataset first to eliminate all contracts with less than five clauses present, resulting in 34,442 contracts, following which we select the top 15 clause types for analyzing the generation. The selected clause types, along with their clause counts and length information, can be seen in Table 1.

Clause type	# clauses	mean length	std length
governing laws	15291	103.00	104.80
amendments	12571	127.19	119.26
entire agreements	11023	98.01	64.50
counterparts	10415	80.47	55.49
notices	9726	148.20	105.62
waivers	8945	133.93	107.43
severability	8776	107.72	61.93
expenses	8365	138.66	119.72
successors	8184	116.60	87.98
survival	6102	89.14	84.58
assigns	6099	106.94	82.61
assignments	5976	127.23	95.94
representations	5373	136.16	102.57
warranties	5320	138.06	138.13
taxes	5184	164.87	126.78

Table 1. Distribution of the selected clause types from LEDGAR.

Encoders. Considering the domain-specific nature of English in legal contracts, directly using transformer encoder models pretrained on generic corpora would not yield good results, as remarked in [2]. We experimented across three models - LegalBERT [9] trained on diverse legal texts (LegalBERT-all), LegalBERT trained only on US con-

tracts from EDGAR (LegalBERT-contracts), and a further pretrained version of BERT [10] on our task-specific data using masked language modeling objective (BERT-mlm). The further pretrained BERT model was pretrained for two epochs on the clauses in LEDGAR data. Unlike the previous work, we did not pretrain a ContractBERT model due to computational limitations and focused on using existing large pretrained models available. `base` versions of all the BERT-based models were used in the experiments performed. To determine the best encoder model for representation, we trained using the representations from these three models on CONTR_TYPE, CONTR_TYPE_FULLSIM and CONTR_TYPE _CLAUSESIM strategies, the results of which are shown in Table 2.

	Strategy					
Encoder	CONTR_TYPE		CONTR_TYPE _FULLSIM		CONTR_CLAUSESIM	
	ROUGE-L	BLEU	ROUGE-L	BLEU	ROUGE-L	BLEU
BERT-mlm	38.39	22.46	38.50	22.33	38.33	21.49
LegalBERT-contracts	38.31	21.53	38.89	22.01	37.45	20.63
LegalBERT-all	**38.75**	**22.96**	**39.28**	**22.69**	**38.85**	**23.33**

Table 2. Performance of encoder models on 3 strategies: CONTR_TYPE, CONTR_TYPE_FULLSIM, CONTR_CLAUSESIM.

LegalBERT trained on diverse legal corpora was consistently observed to outperform the other two models and was chosen as the base encoder for all subsequent experiments.

Indexing. An HNSW [11] index of the precomputed contract representations was built using FAISS [12] indexing library, and a similarity search was performed using L2 distance. Care was taken to eliminate the contract for which the recommended clause is not in retrieved contracts, and the similar contract representation was averaged on the rest. Because of this consideration, the minimum value of k was kept as 2.

Model. A transformer decoder [13] with three layers was trained from scratch across all the experiments, following the setting in ClauseRec. A wordpiece tokenizer was trained on output clauses with a vocabulary size of 8192 tokens. For the final experiments, all the decoder models were trained for 50 epochs using AdamW [14] optimization with a learning rate schedule having 25% warmup up to 6e-5 followed by linear decay. A batch size of 24 with accumulated gradients over three steps was used. Each of the experiments was performed using 2 RTX 2080 Ti GPUs.

Computing the input representations to the decoder was computationally heavy at training time, even without flowing gradients through the encoders for backpropagation. One epoch took approximately 17 hours on a single GPU, most of which was spent calculating the input representation. This prohibitively high time severely restricted the number of experiments that could be performed on our computational resources. Hence, to fasten the decoder training process, we serialized all the representations to be used prior to the experiment, which reduced the per-epoch per-GPU duration to less than 30 min.

Metrics. Commonly used text generation metrics ROUGE [15] and BLEU [16] were used to evaluate the system's performance. ROUGE, a recall-oriented metric, can be used to indicate how many of the necessary legal phrases are generated in clauses while BLEU is indicative of the precision of the phrases. Human evaluation was not performed due to practical constraints in obtaining domain-specific experts for clause comprehension.

5. Results

Strategy	Clause type / Overall	ROUGE			BLEU		
		ROUGE-1	ROUGE-2	ROUGE-L	BLEU-1	BLEU-2	BLEU
ONLY_CONTR	governing laws	38.44	18.89	30.55	14.84	11.46	10.22
	amendments	33.25	13.78	24.35	15.78	12.45	12.18
	entire agreements	34.26	13.68	25.13	10.88	10.12	9.60
	Overall	34.94	15.08	25.52	13.49	12.23	11.32
CONTR_TYPE	governing laws	52.92	37.78	46.38	47.90	33.04	31.32
	amendments	37.98	20.38	31.72	14.16	8.43	8.01
	entire agreements	48.48	28.49	40.78	26.28	15.02	13.53
	Overall	47.93	29.20	38.75	39.01	24.26	22.96
CONTR _FULLSIM	governing laws	40.57	22.11	33.48	27.45	15.24	14.70
	amendments	28.62	7.85	19.23	20.13	6.58	5.61
	entire agreements	23.78	5.25	16.12	10.97	2.41	1.25
	Overall	31.45	12.43	23.30	21.06	8.60	8.14
CONTR_TYPE _FULLSIM	governing laws	54.27	38.07	47.00	50.60	34.27	32.37
	amendments	41.35	23.86	34.74	20.72	13.19	12.78
	entire agreements	49.21	29.68	41.57	24.85	15.14	13.94
	Overall	**48.38**	**29.72**	**39.26**	**39.04**	**24.39**	**23.05**
CONTR_TYPE _CLAUSESIM	governing laws	52.66	36.62	45.89	47.84	33.24	31.66
	amendments	40.54	21.42	32.12	24.37	14.63	14.16
	entire agreements	49.79	30.17	42.00	28.21	17.58	16.26
	Overall	47.75	29.30	38.85	39.12	24.60	23.33

Table 3. Clause recommendation results across all the strategies tried out shown for top 3 clause types (based on no. of clauses) and overall metrics calculated on all the types.

To understand the number of retrieved similar contracts needed to build a suitable context, the experiments were conducted by varying values of k from 2 to 12 in steps of 2 for CONTR_TYPE_FULLSIM strategy. Not much variance in the text generations scores was observed, with the BLEU and ROUGE staying more or less around the same values. The value of k was fixed to 6 for subsequent experiments.

Table 3 shows the metrics across all the strategies for the top 3 clause types (based on the no. of clauses in the dataset) and the overall scores computed for clauses for all the 15 clause types. Metrics for individual clause types have not been added due to space constraints.

From the results, it can be easily observed that similar contracts with clause type and input contract provide the best contextual representation among the strategies tried. However, the strategy excluding similar contract representation performs competitively with the former. In a clause-type agnostic setting, the problem becomes much more difficult with modest ROUGE scores and dismal BLEU scores indicating a high amount of hallucinated, irrelevant content. Unlike clause type aware setting, the augmentation of similar contract information does not help, with an observed degradation in model output.

6. Analysis

We conduct analysis based on the generation of clauses from the best performing CONTR_TYPE_FULLSIM strategy.

The performance of clause generation across all the 15 clause types on BLEU and ROUGE-L scores can be seen in Figure 2. The clause types *successors* and *counterparts* have the highest BLEU and ROUGE-L scores, respectively. It can be observed that the clause types with generally higher BLEU (*governing laws, counterparts, successors*) scores follow suit in ROUGE-L scores, and those performing poorly in terms of BLEU (*amendments, representations, warranties, taxes*) have poor ROUGE-L scores.

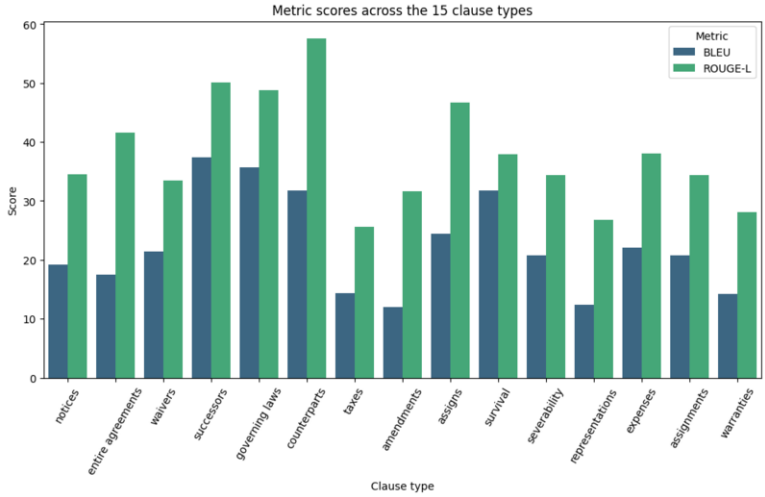


Figure 2. Variation of BLEU and ROUGE score across all 15 clause types considered using the CONTR_TYPE_FULLSIM

In Figure 3, we plot the TSNE representations of the actual and generated clauses for a subset of 1000 randomly sampled clauses from our test set. We use LegalBERT-all for encoding the clauses, followed by dimensionality reduction to obtain the plot. The plot showcases the semantic closeness of the actual and generated clauses based on the closeness of the two distributions.

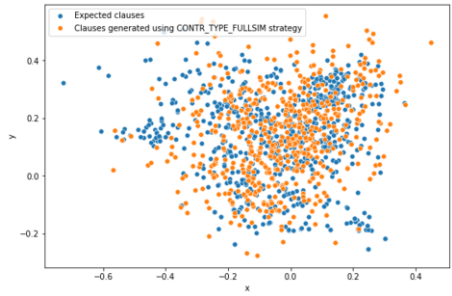


Figure 3. TSNE plot comparing the representations obtained for clause generated by the CONTR_TYPE_FULLSIM strategy against actual clauses

We show examples of a few generated clauses by the model in Figure 4. Based on the comparison with their expected counterparts, we can appreciate the ability of the model to generate characteristic clause content based on the type of clause. The exactness of the clause is not guaranteed here, as the model may generate semantically equivalent content deemed to be an equally valid clause, such as "this release" v/s "this agreement." Tailoring clauses to bear exact phrasal content can entail incorporating keyword-level information as future work.

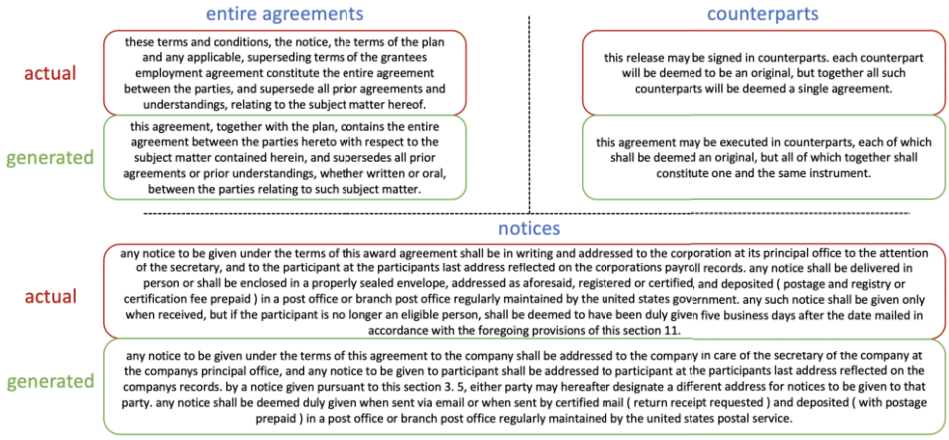


Figure 4. Examples of generated clauses in comparison to their expected (actual) counterparts using the CONTR_TYPE_FULLSIM strategy

One issue we would like to point out in the current generation is the tendency of the model to typically generate verbose clauses, which might have resulted from optimization on the training objective to include as many valid phrases as possible in a clause. As seen in Table 4, the mean and median lengths of generated clauses are far more than the actual clauses. The Pearson correlation coefficient between the two length distributions turned out to be 0.36, indicating a weak correlation. Reducing such extraneous content in generated clauses will work towards finer clause generation.

	mean	std	median
actual	88.05	88.54	54.0
generated	108.02	84.14	84.0

Table 4. Statistics of generated and actual clause lengths

7. Conclusion & Future scope

In this work, we explored the problem of clause recommendation by experimenting with several strategies for modeling the contextual input for recommendation and observed the effectiveness of similar contract representation in a clause-type aware setting. While decent results are obtained in this paradigm, the clause type agnostic setting remains a complex problem to solve and is more relevant for scaling clause recommendation to a broader set of clauses. With the considerably large size of contractual documents, fu-

ture work can explore tackling the cold start problem in clause recommendation (i.e., recommending clause without prior contract state as context) and allowing for disentangling and customization of named entities in recommended clauses. Future work can also achieve robustness by handling a more diverse range and a larger number of clause types compared to the 15 clause topics we focus on.

References

[1] Tuggener, D, Däniken, P, Peetz, T, Cieliebak, M LEDGAR: A Large-Scale Multi-label Corpus for Text Classification of Legal Provisions in Contracts. In Proceedings of the 12th Language Resources and Evaluation Conference 2020 (pp. 1235–1241). European Language Resources Association.

[2] Aggarwal, V, Garimella, A, Srinivasan, B, N, A, Jain, R ClauseRec: A Clause Recommendation Framework for AI-aided Contract Authoring. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing 2021 (pp. 8770–8776). Association for Computational Linguistics.

[3] Chalkidis, I, Androutsopoulos, I, Michos, A Extracting Contract Elements. In Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law 2017 (pp. 19–28). Association for Computing Machinery.

[4] Leivaditi, S, Rossi, J, Kanoulas, E. A Benchmark for Lease Contract Review.

[5] Marco Lippi, Przemyslaw Palka, Giuseppe Contissa, Francesca Lagioia, Hans-Wolfgang Micklitz, Giovanni Sartor, Paolo Torroni. "CLAUDETTE: an automated detector of potentially unfair clauses in online terms of service". Artificial Intelligence and Law 2019; 27(2):117–139.

[6] Kubeka, S, Ade-Ibijola, A. "Automatic Comprehension and Summarisation of Legal Contracts". Advances in Science, Technology and Engineering Systems Journal 2021; 6(2):19–28.

[7] Borchmann, L, Wisniewski, D, Gretkowski, A, Kosmala, I, Jurkiewicz, D, Szalkiewicz, , Palka, G, Kaczmarek, K, Kaliska, A, Gralinski F. "Contract Discovery: Dataset and a Few-Shot Semantic Retrieval Challenge with Competitive Baselines". In Findings of the Association for Computational Linguistics: EMNLP 2020 2020 (pp. 4254–4268). Association for Computational Linguistics.

[8] Dan Hendrycks, Collin Burns, Anya Chen, Spencer Ball. "CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review". NeurIPS 2021.

[9] Chalkidis, I, Fergadiotis, M, Malakasiotis, P, Aletras, N, Androutsopoulos, I. "LEGAL-BERT: The Muppets straight out of Law School". In Findings of the Association for Computational Linguistics: EMNLP 2020 2020 (pp. 2898–2904). Association for Computational Linguistics.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding".

[11] Malkov, Y, Yashunin, D. "Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs". IEEE Transactions on Pattern Analysis and Machine Intelligence 2020; 42(4):824-836.

[12] Johnson, J, Douze, M, Jegou, H. "Billion-scale similarity search with GPUs". IEEE Transactions on Big Data 2019; 7(3):535–547.

[13] Vaswani, A, Shazeer, N, Parmar, N, Uszkoreit, J, Jones, L, Gomez, A, Kaiser, Polosukhin, I. "Attention is All You Need". In Proceedings of the 31st International Conference on Neural Information Processing Systems 2017 (pp. 6000–6010). Curran Associates Inc..

[14] Ilya Loshchilov, Frank Hutter. "Decoupled Weight Decay Regularization". In ICLR 2019 .

[15] Lin, CY ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out 2004 (pp. 74–81). Association for Computational Linguistics.

[16] Papineni, K, Roukos, S, Ward, T, Zhu, WJ BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics 2002 (pp. 311–318). Association for Computational Linguistics.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220451

Modelling and Explaining Legal Case-Based Reasoners Through Classifiers

Xinghan LIU [a], Emiliano LORINI [a], Antonino ROTOLO [b] and Giovanni SARTOR [b]
[a] *IRIT-CNRS, University of Toulouse, France*
[b] *Alma Human AI, University of Bologna, Italy*

Abstract. This paper brings together factor-based models of case-based reasoning (CBR) and the logical specification of classifiers. Horty [8] has developed the factor-based models of precedent into a theory of precedential constraint. In this paper we combine binary-input classifier logic (BCL) to classifiers and their explanations given by Liu & Lorini [13,14] with Horty's account of factor-based CBR, since both a classifier and CBR map sets of features to decisions or classifications. We reformulate case bases in the language of BCL, and give several representation results. Furthermore, we show how notions of CBR can be analyzed by notions of classifier explanation.

Keywords. Case-based reasoning, Modal Logic for classifiers, Explainable AI

1. Introduction

This paper brings together two lines of research: factor-based models of case-based reasoning (CBR) and the logical specification of classifiers.

Logical approaches to classifiers capture the connection between features and outcomes in classifier systems. They are well-suited for modeling and computing a large variety of explanations of a classifier's decisions [18,5,12,11,4,13], e.g., prime implicants, abductive, contrastive and counterfactual explanations. Consequently, they enable detecting biases and discrimination in the classification process. They can thus contribute to provide controllability and explainability over automated decision-making (as required, e.g., by Art. 22 GDPR and by Art. 6 ECHR relative to judicial decisions).

Factor-based reasoning [2,1] is a popular approach to precedential reasoning in AI&law research. The key idea is that a case can be represented as a set of factors, where a factor is a legally relevant aspect. Factors are assumed to have a direction, i.e., to favor certain outcomes. Usually both factors and outcomes are assumed to be binary, so that each factor can be labelled with the outcome it favors (usually denoted as π, the outcome requested by the plaintiff, and δ, the outcome requested by the defendant). The party which is interested in a certain outcome in a new case can support her request by citing a past case that has the same outcome, and shares with the new case some factors supporting that outcome. The party that is interested in countering that outcome can respond with a distinction, i.e., can argue that some factors which supported that outcome in the precedent are missing in the new case or that some additional factors against that outcome are present in the new case. Horty [7,9] has developed the factor-based models

of precedent into a theory of precedential constraints, i.e., of how a new case must be decided, in order to preserve consistency in the case law. In [8,6], he takes into account the fact that judges may also provide explicit reasons for their choice of a certain outcome. This leads to the distinction between the result and the reason model of precedents. In the first model, the message conveyed by the case is only that all factors supporting the case-outcome (pro-factors) outweigh all factors against that outcome (con-factors). In the second, the message is that the factors for the case outcome indicated by the judge outweigh all factors against that outcome.

In this paper we shall combine Liu & Lorini's modal logic approach to classifiers and their explanations [13,14] with Horty's account of factor-based CBR. The combination is based on the fact that both a classifier and CBR map sets of features to decisions or classifications. In this way, our contribution is at least twofold.

First, we explore the relation between two apparently unrelated reasoning systems. While the connection between CBR and reasoning about classifier systems is of interest in itself, we believe that, through this relation, new research perspectives can be offered, since we could in the future investigate CBR by exploiting several techniques and results from modal logic. We will see that the challenge of this paper is to adapt the formal representation of a classifier to the bidirectionality of factors in the HYPO model. Once this is solved, we can provide a logical model and a semantics for factor-based CBR.

Second, we investigate the idea of normative explanation: While the literature on the concept of explanation is immense, the AI community is now paying attention to it due to the development of explainable AI (XAI) [15,3]. Our paper, by connecting CBR and reasoning about classifier systems, explores different notions of explanation in law, such as abductive and contrastive explanations for the outcome suggested by the case-based reasoner. Our model allows for building explainable case-based reasoners, which could also be deployed to reproduce and analyze the functioning of opaque predictors of the outcome of cases. We import notions such as prime implicant and contrastive explanation in the domain of XAI for classifiers to showcase how to analyze CBR in the field of XAI.

The the paper is organised as follows. Section 2 presents Horty's models of CBR. Section 3 introduces the notion of classifier model (CM) for the binary-input classifier logic BCL. Section 4 studies the connection between CBR and classifier models.Section 5 shows that notions for classifier explanation in XAI help study case base. Finally, Section 6 discusses related work and concludes. Proofs and the axiomatics are in the appendix.[1]

2. Horty's Two Models of Case-Based Reasoning

In this section we account for the two models of case-based reasoning / precedential constraint proposed by Horty. We simply say *result model* for "the factor-based result model of precedential constraint" and *reason model* for "the factor-based reason model of precedential constraint".

Let $Atm_0 = Plt \cup Dfd$, where Plt and Dfd are disjoint sets of factors favoring the plaintiff and defendant respectively. In addition, let $Val = \{1, 0, ?\}$ where elements stand for *plaintiff wins*, *defendant wins* and *indeterminacy* respectively. Let $Dec = \{\mathtt{t}(x) : x \in$

[1] The paper with appendix is available here: https://arxiv.org/abs/2210.11217.

Val} and read $\mathtt{t}(x)$ as "the actual decision/outcome (of the judge/classifier) takes value x". An outcome $\mathtt{t}(1)$ or $\mathtt{t}(0)$ means that, the judge is predicted to decide for the plaintiff or for the defendant (the classifies "forces" one of the two outcomes). The outcome $\mathtt{t}(?)$ means either outcome would be consistent: the judge may develop the law in one direction or the other. This reflects the incompleteness nature of CBR. We use *Atm* to denote $Atm_0 \cup Dec$.

We call $s \subseteq Atm_0$ a *fact situation*. A set of atoms X is called a *reason* for an outcome (decision) x if it a set of factors all favoring the same outcome: $X \subseteq Plt$ is a reason for 1 and $X \subseteq Dfd$ is a reason for 0. A (defeasible) *rule* consist of a reason and the corresponding outcome: $X \mapsto x$ is rule, if $X \subseteq Plt$ and $x = 1$, or $X \subseteq Dfd$ and $x = 0$. For readability, we make a convention that, for $x \in \{0,1\}$, let $\overline{x} = 1 - x$ and $\overline{\overline{x}} = x$. Moreover, let $Atm_0^x = Plt$ if $x = 1$, and $Atm_0^x = Dfd$ if $x = 0$.

In the reason model, a *precedent case* (precedent) is a triple $c = (s, X, x)$, where $s \subseteq Atm_0$, $X \subseteq Atm_0^x, x \in \{0,1\}$. In plain words, $s \cap Atm_0^x$ contains all *pro-factors* in s for x, while $s \cap Atm_0^{\overline{x}}$ all *con-factors* in s for x. X is the *reason of the case*, namely a subset of the pro-factors which the judge considers sufficient to support that outcome, relative to all con-factors in the case.

A *case base CB* (for reason model) is a set of precedential cases. When the reason contains all pro-factors within the situation (i.e., when $c = (s, s \cap Atm_0^x, x)$) all such factors are considered equally decisive. If a case base only contains cases of this type, we obtain what Horty calls "the result model", and note such a case base CB^{res}.[2] The class of all *CBs* and $CB^{\text{res}}s$ are noted **CB** and $\mathbf{CB}^{\text{res}}$ respectively.

Example 1 (Running example)**.** *In the paper we refer to the following running example taken from [16]. Let us assume the following six factors, each of which either favors the outcome 'misuse of trade secrets' ('the plaintiff wins') or rather favors the outcome no misuse of trade secrets ('the defendant wins'): the defendant had obtained the secret by deceiving the plaintiff (π_1) or by bribing an employee of the plaintiff (π_2), the plaintiff had taken security measures to keep the secret (π_3) , the information is obtainable elsewhere (δ_1), the product is reverse-engineerable (δ_2) and the plaintiff had voluntarily disclosed the secret to outsiders (δ_3). Hence in our running example* $Atm = \{\pi_1, \pi_2, \pi_3, \delta_1, \delta_2, \delta_3, \mathtt{t}(0), \mathtt{t}(1), \mathtt{t}(?)\}$ *Let us consider a case base* $CB^{ex} = \{c_1, c_2\}$ *where* $c_1 = (\{\pi_1, \pi_3, \delta_1, \delta_3\}, \{\pi_1\}, 1)$; $c_2 = (\{\pi_2, \delta_1, \delta_3\}, \{\delta_3\}, 0)$, *which means:*

- c_1 *has factors (fact situation)* $s_1 = \{\pi_1, \pi_3, \delta_1, \delta_3\}$, *reason* $\{\pi_1\}$ *and outcome* 1*;*
- c_2 *has outcome* δ, *factors* $s_2 = \{\pi_2, \delta_1, \delta_3\}$, *reason* $\{\delta_3\}$ *and outcome* 0

A case base can be inconsistent when two precedents map the same fact situation to different outcomes. Another scenario is that a consistent case base becomes inconsistent after *update*, namely after expanding it with some new case. Hence maintaining consistency is the crucial concern of case-based reasoning. But first of all, one need to define these notions. The following definitions, except symbolic difference, are based on [8,16].

Definition 1 (Preference relation derived from a case)**.** *Let* $c = (s, X, x)$ *be a case. Then the preference relation* $<_c$ *derived from c is s.t. for any two reasons Y, Y′ favoring x and* $\overline{x}$ *respectively,* $Y' <_c Y$ *iff* $Y' \subseteq s \cap Atm_0^{\overline{x}}$ *and* $X \subseteq Y$.

[2] So we view result model as a special kind of reason model, as [8, p. 25] also mentioned.

Definition 2 (Preference relation derived from a case base)**.** *Let CB be a case base. Then the preference relation $<_{CB}$ derived from CB is s.t. for any two reasons Y, Y' favoring x and $\bar{x}$ respectively, $Y' <_{CB} Y$ iff $\exists c \in CB$ s.t. $Y' <_c Y$.*

Definition 3 ((In)consistency)**.** *A case base CB is inconsistent, if there are two reasons Y, Y' s.t. $Y' <_{CB} Y$ and $Y <_{CB} Y'$. CB is consistent if it is not inconsistent.*

Definition 4 (Precedential constraint)**.** *Let CB be a consistent case base, X is a reason for x in CB and applicable in a new fact situation s', i.e. $X \subseteq s'$. Updating CB with the new case (s', X, x) meets the precedential constraint, iff $CB \cup \{(s', X, x)\}$ is still consistent.*

There is more than one way to satisfy the precedential constraint, depending on how the precedents in *CB* interacts with the new case. The requirement of consistency dictates the outcome when the "a fortiori" constraint applies: if reason X for x outweighs (i.e., is stronger that) reason $s \cap Atm_0^{\bar{x}}$, a fortiori any superset of X outweighs any subset of $s \cap Atm_0^{\bar{x}}$, so that only by deciding for x rather than for $\bar{x}$ consistency is maintained.[3]

Example 2 (Running example)**.** *Let us consider two fact situations according to case base CB^{ex} running example.*

- *In $s_3 = \{\pi_1, \pi_3, \delta_1\}$, only a decision for 1 in s_3 is consistent with CB^{ex}, since a decision for 0 would entail that $\{\delta_1\} >_{CB^{ex}} \{\pi_1\}$, contrary to the preference $\{\pi_1\} >_{CB^{ex}} \{\delta_1\}$, which is derivable from c_1.*
- *In $s_4 = \{\pi_2, \delta_2\}$ both $(s_4, \{\pi_2\}, 1)$ and $(s_4, \{\delta_2\}, 0)$ are consistent with CB^{ex}, since neither $\{\pi_2\} >_{CB^{ex}} \{\delta_2\}$ nor $\{\delta_2\} >_{CB^{ex}} \{\pi_2\}$.*

3. Classifier Model of Binary-input Classifier Logic

In this section we introduce the language and semantics of binary-input classifier logic BCL first appeared in [13]. Recall that $Atm = Atm_0 \cup Dec$, where $Atm_0 = Dfd \cup Plt$, and $Dec = \{\mathtt{t}(x) : x \in Val = \{0, 1, ?\}\}$. The modal language $\mathscr{L}(Atm)$ of BCL is defined as:

$$\varphi \quad ::= \quad p \mid \mathtt{t}(x) \mid \neg\varphi \mid \varphi \wedge \varphi \mid [X]\varphi,$$

where p ranges over Atm_0, $\mathtt{t}(x)$ ranges over Dec, and X is a finite subset of Atm_0.[4] Operator $\langle X \rangle$ is the dual of $[X]$ and is defined as usual: $\langle X \rangle \varphi =_{def} \neg[X]\neg\varphi$. Finally, for any $X \subseteq Y \subseteq Atm_0$, the following definition syntactically expresses a valuation on Y s.t. all variables in X are assigned as true, while all the rest in Y are false.

$$\mathsf{cn}_{X,Y} =_{def} \bigwedge_{p \in X} p \wedge \bigwedge_{p \in Y \setminus X} \neg p.$$

The language $\mathscr{L}(Atm)$ is interpreted relative to classifier models defined as follows.

[3] We generalize a fortiori constraint from only acting on result models in [8] to also on reason models in the same manner as viewing a result models as a special reason model, whose reason contains all pro-factors.

[4] Atm is finite since the factors in case-based reasoning are supposed to be finite. Notice p ranging over $Dfd \cup Plt$, i.e. p can be some δ or some π. X can denote a reason (an exclusive set of plaintiff/defendant factors), or any subset of Atm_0, which is clear from the context. Last but not least, p and $\mathtt{t}(x)$ have different statuses regarding negation: $\neg p$ means that the input variable p takes value 0, but $\neg\mathtt{t}(x)$ merely means the output does not take value x: we do not know which value it takes, since the output is trinary.

Definition 5 (Classifier model)**.** *A classifier model (CM) is a pair $C = (S, f)$ where:*

- $S \subseteq 2^{Atm_0}$ *is a set of states (or fact situations), and*
- $f : S \longrightarrow Val$ *is a decision (or classification) function.*

The class of classifier models is noted **CM**.

A pointed classifier model is a pair (C, s) with $C = (S, f)$ a classifier model and $s \in S$. Formulas in $\mathscr{L}(Atm)$ are interpreted relative to a pointed classifier model, as follows.

Definition 6 (Satisfaction relation)**.** *Let (C, s) be a pointed classifier model with $C = (S, f)$ and $s \in S$. Then:*

$$
\begin{aligned}
(C,s) &\models p \iff p \in s,\\
(C,s) &\models \mathtt{t}(x) \iff f(s) = x,\\
(C,s) &\models \neg\varphi \iff (C,s) \not\models \varphi,\\
(C,s) &\models \varphi \wedge \psi \iff (C,s) \models \varphi \text{ and } (C,s) \models \psi,\\
(C,s) &\models [X]\varphi \iff \forall s' \in S : \text{ if } (s \cap X) = (s' \cap X) \text{ then } (C,s') \models \varphi.
\end{aligned}
$$

A formula φ of $\mathscr{L}(Atm)$ is said to be satisfiable relative to the class **CM** if there exists a pointed classifier model (C, s) with $C \in \mathbf{CM}$ such that $(C, s) \models \varphi$. It is said to be valid if $\neg\varphi$ is not satisfiable relative to **CM** and noted as $\models_{\mathbf{CM}} \varphi$.

We can think of a pointed model (C, s) as a pair (s, x) in f with $f(s) = x$. The formula $[X]\varphi$ is true at a state s if φ is true at all states that are modulo-X equivalent to state s. It has the *selectis paribus* (SP) (selected things being equal) interpretation "features in X being equal, necessarily φ holds (under possible perturbation on the other features)". $[Atm_0 \setminus X]\varphi$ has the standard *ceteris paribus* (CP) interpretation "features other than X being equal, necessarily φ holds (under possible perturbation of the features in X)". Notice when $X = \emptyset$, $[\emptyset]$ is the S5 universal modality since every state is modulo-$\emptyset$ equivalent to all states, viz. $(C, s) \models [\emptyset]\varphi \iff \forall s' \in S, (C, s') \models \varphi$.

4. Representation between Consistent Case Base and CM

In this section we shall show that the language of case bases can be translated into the language $\mathscr{L}(Atm)$; hence case bases can be studied by classifier models. More precisely, a case base is consistent iff its translation, together with the following two formulas that we abbreviate as `Compl` and `2Mon`, is satisfiable in the class **CM**:

$$
\mathtt{Compl} =_{def} \bigwedge_{X \subseteq Atm_0} \langle\emptyset\rangle \mathsf{cn}_{X, Atm_0}
$$

$$
\mathtt{2Mon} =_{def} \bigwedge_{x \in \{0,1\}, X \subseteq Atm_0^x, Y \subseteq Atm_0^{\overline{x}}} \Big(\langle\emptyset\rangle(\mathsf{cn}_{X \cup Y, Atm_0} \wedge \mathtt{t}(x)) \rightarrow \bigwedge_{Atm_0^x \supseteq X' \supseteq X, Y' \subseteq Y} [\emptyset](\mathsf{cn}_{X' \cup Y', Atm_0} \rightarrow \mathtt{t}(x)) \Big)
$$

According to Compl, *every possible* situation description must be satisfied by the classifier, where a situation description is a conjunction of factors (those being present X) and negations of factors (those being absent, $Atm_0 \setminus X$).

2Mon introduces a *two-way monotonicity*, which is meant to implement the *a fortiori* constraint: if the classifier associates a situation s to an outcome x, then it must assign the same outcome to every situation s' such that both (a) s' includes all factors for x that are in s and (b) s' does *not include* factors for $\bar{x}$ that are *outside of* s. This formula is meant to maintain consistency with respect to the preference relation, as Definition 1 indicates: if a case including reason X for x and factors Y for $\bar{x}$, has outcome x, it means that $X > Y$. Thus it cannot be that outcome $\bar{x}$ is assigned to a situation s' including both a superset $X' \supseteq X$ of factors for x and only a subset $Y' \subseteq Y$ of factors for $\bar{x}$. In fact, if $X > Y$, then is must be the case that also $X' > Y'$, while a decision for $\bar{x}$ would entail that $X' < Y'$.

Let $\mathbf{CM}^{prec} = \{C = (S, f) \in \mathbf{CM} : \forall s \in S, (C, s) \models \mathtt{Compl} \wedge \mathtt{2Mon}\}$, where $\mathbf{CM}^{prec}$ means the class of CMs for precedent theory. Satisfiability and validity relative to $\mathbf{CM}^{prec}$ are defined in an analogous way as $\mathbf{CM}$.

To translate a result-model case-base CB^{res} into a classifier model (C, f), we need to ensure that all precedents in the case-base are satisfied by the classifier, with regard to both their factors and their outcome.

Definition 7 (Translation of case base for result model)**.** *The translation function tr_1 maps each case from a case base CB^{res} to a corresponding formula in the language $\mathscr{L}(Atm)$. It is defined as follows:*

$$tr_1(s, s \cap Atm_0^x, x) =_{def} \langle \emptyset \rangle (\mathtt{cn}_{s, Atm_0} \wedge \mathtt{t}(x)).$$

We generalize it to the entire case base CB^{res} as follows:

$$tr_1(CB^{\text{res}}) =_{def} \bigwedge_{(s, s \cap Atm_0^x, x) \in CB} tr(s, s \cap Atm_0^x, x).$$

Therefore, in the result model a precedent $(s, s \cap Atm_0^x, x)$ is viewed as a situation s being classified by f as x.

Example 3 (Running example)**.** *The case $(\{\pi_1, \pi_2, \delta_1\}, \{\pi_1, \pi_2\}, 1\})$ is translated as $\langle \emptyset \rangle (\pi_1 \wedge \pi_2 \wedge \delta_1 \wedge \neg \pi_3 \wedge \neg \delta_2 \wedge \neg \delta_3 \wedge \mathtt{t}(1))$, which means that $f(\pi_1, \pi_2, \delta_1) = 1$*

In translations for the reason model we need to capture the role of reasons. This is obtained by ensuring that for every case (s, X, x), not the fact situation s directly, but the one consisting only of reason X and all $\bar{x}$-factors in s (i.e. $s \cap Atm_0^{\bar{x}}$) is classified as x. It reflects that the precedent finds x-factors in s outside of X dispensable for the outcome.

Definition 8 (Translation of case base for reason model)**.** *The translation function tr_2 maps each case from a case base CB to a corresponding formula in the language $\mathscr{L}(Atm)$. It is defined as follows:*

$$tr_2(s, X, x) =_{def} \langle \emptyset \rangle (\mathtt{cn}_{X \cup (s \cap Atm_0^{\bar{x}}), Atm_0} \wedge \mathtt{t}(x)).$$

We generalize it to the entire case base CB as follows:

$$tr_2(CB) =_{def} \bigwedge_{(s,s\cap Atm_0^x,x)\in CB} tr_2(s,s\cap Atm_0^x,x).$$

Notice that the function tr_1 for the result model is a special case of the function tr_2 for the reason model, since $((s\cap Atm_0^x)\cup(s\cap Atm_0^{\overline{x}}) = s$.

Fact 1. $tr_1(s,s\cap Atm_0^x,x) = tr_2(s,s\cap Atm_0^x,x)$.

The formulas `2Mon` and `Compl` require that the outcome x supported by reason X in a precedent is assigned to all possible cases including X that do not contain additional factors against x. If both formulas are satisfiable then the case base is consistent, as stated by the following theorem.

Theorem 1. *Let $CB \in$ **CB** be a case base. Then, CB is consistent iff $tr_2(CB)$ is satisfiable in* $\mathbf{CM}^{prec}$.

In light of the theorem and the fact above, the representation of a case base for result model turns to be a corollary.

Corollary 1. *Let $CB^{\text{res}} \in \mathbf{CB}^{\text{res}}$ be a case base for the result model. Then, CB^{res} is consistent iff $tr_1(CB^{\text{res}})$ is satisfiable in* $\mathbf{CM}^{prec}$.

Similarly, the precedential constraint can also be represented as a corollary.

Corollary 2. *Let $CB \in$ **CB** be a consistent case base and (s',X,x) a case. Updating CB with (s',X,x) meets the precedential constraint, iff $tr_2(CB)\wedge tr_2(s',X,x)$ is satisfiable in* $\mathbf{CM}^{prec}$.

Example 4 (Running example)**.** *Case $c_3 = (\{\pi_1,\pi_2,\delta_2\},\{\delta_2\},0)$ is incompatible with the CB^{ex}. According to $tr_2(CB^{ex}\cup\{c_3\})$, `2Mon` and `Compl`, the fact situation $\{\pi_1,\pi_2,\delta_1\}$ should be classified both as 1, based on CB^{ex}, and 0, based on c_3.*

5. Explanations

The representation results above pave the way to providing explanations for the outcomes of cases. For this purpose it is necessary to introduce the following notations. Let λ denote a conjunction of finitely many literals, where a literal is an atom p (positive literal) or its negation $\neg p$ (negative literal). We write $\lambda\subseteq\lambda'$, call λ a part (subset) of λ', if all literals in λ also occur in λ'; and $\lambda\subset\lambda'$ if $\lambda\subseteq\lambda'$ but not $\lambda'\subseteq\lambda$. We write $Lit(\lambda), Lit^+(\lambda), Lit^-(\lambda)$ to mean all literals, all positive literals and all negative literals in λ respectively. By convention $\top$ is a term of zero conjuncts. In the glossary of Boolean classifier (function), λ is called a *term* or *property* (of the instance s). The set of terms is noted *Term*. A key role in our analysis is played by the notion of a (prime) implicant, i.e., a (subset-minimal) term which makes a classification necessarily true.

Definition 9 (Implicant (Imp) and prime implicant (PImp))**.** *We write $\mathtt{Imp}(\lambda,x)$ to mean that λ is an* implicant *for x and define it as $\mathtt{Imp}(\lambda,x) =_{def} [\emptyset](\lambda\rightarrow\mathtt{t}(x))$. We write $\mathtt{PImp}(\lambda,x)$ to mean that λ is a* prime implicant *for x and define it as*

$$\mathtt{PImp}(\lambda,x) =_{def} [\emptyset]\Big(\lambda\rightarrow\big(\mathtt{t}(x)\wedge\bigwedge_{p\in Atm(\lambda)}\langle Atm(\lambda)\setminus\{p\}\rangle\neg\mathtt{t}(x)\big)\Big).$$

According to the definition, λ being an implicant for x means that any state s verifying λ is necessarily classified as x (necessity); and λ being a prime implicant for x means that any proper subset of λ is not an implicant for x (minimality).[5] Implicants explain the classifier in the sense that to know an implicant satisfied at a state is to know the classification of the state.

Intuitively, for a case base containing precedent (s,X,x) to be consistent, s must be incompatible with every prime implicant λ for $\bar{x}$. To guarantee that, either λ must have some literal $\neg p$, where p is in X and hence is true at s; or λ must have some literal p, where $p \notin s \cap Atm_0^{\bar{x}}$ and hence is false at s.

Proposition 1. *Let CB be a consistent case base and $(s,X,x) \in CB$, and $C \in \mathbf{CM}^{prec}$ s.t. $(C,s) \models tr_2(CB)$. Then, $\forall \lambda \in Term_{,}$ if $(C,s) \models \mathtt{PImp}(\lambda,\bar{x})$, then either $X \cap Atm(Lit^-(\lambda)) \neq \emptyset$ or $s \cap Atm_0^{\bar{x}} \not\supseteq Atm(Lit^+(\lambda))$.*

Example 5. *Let $C = (S,f) \in \mathbf{CM}^{prec}$ and $tr_2(CB^{ex})$ is satisfiable in C. Obviously π_1 cannot be PImp for 0, otherwise $f(s_1) = 0$, contrary to c_1. Also $\neg\delta_2 \wedge \pi_2$ cannot be PImp for 1, otherwise $f(\{\pi_2,\delta_1,\delta_3\}) = 1$, contrary to c_2.*

In XAI, people [18,5,12] also focus on "local" (prime) implicants, namely (prime) implicants true at a given state. We adopt the definitions of abductive explanations in [12,10], and express these notions in $\mathscr{L}(Atm)$ as follows:

Definition 10 (Abductive explanation (AXp) and weak abductive explanation (wAXp))**.** *We write $\mathtt{AXp}(\lambda,x)$ to mean that λ* abductively explains *the decision x and define it as $\mathtt{AXp}(\lambda,x) =_{def} \lambda \wedge \mathtt{PImp}(\lambda,x)$. We write $\mathtt{wAXp}\ (\lambda,x)$ to mean that λ* weak-abductively explains *the decision x and define it as $\mathtt{wAXp}\ (\lambda,x) =_{def} \lambda \wedge \mathtt{Imp}(\lambda,x)$.*

The proposition below states that to be the reason (of a fact situation) is to be the positive part of some weak AXp of that situation. Notice a reason is not always the positive part of some AXp, since reasons in precedent do not in general respect minimality.

Proposition 2. *Let CB be a consistent case base, $(s,X,x) \in CB$, and $C \in \mathbf{CM}^{prec}$ s.t. $(C,s) \models tr_2(CB)$. Then $\exists \lambda \in Term$ s.t. $Atm(Lit^+(\lambda)) = X$ and $(C,s) \models \mathtt{wAXp}\ (\lambda,x)$.*

In fact, we always know one weak AXp for a precedent (s,X,x) in a consistent case base, i.e., the conjunction of all factors in X and negations of all $\bar{x}$-factors that are in s.

Proposition 3. *Let CB be a consistent case base, $(s,X,x) \in CB$, and $C \in \mathbf{CM}^{prec}$ s.t. $(C,s) \models tr_2(CB)$. Then we have $(C,s) \models \mathtt{wAXp}\ (\mathtt{cn}_{X,X\cup(s\cap Atm_0^{\bar{x}})},x)$.*

Example 6. *Let $C \in \mathbf{CM}^{prec}$ be a model of $tr_2(CB^{ex})$. Then we have $(C,s_1) \models \mathtt{wAXp}\ (\pi_1 \wedge \neg\delta_2,1)$ and $(C,s_2) \models \mathtt{wAXp}\ (\delta_2 \wedge \neg\pi_1 \wedge \neg\pi_2,0)$. Notice that $(C,s_2) \models \neg\mathtt{wAXp}\ (\delta_2,0)$, because e.g. $(C,s_1) \models \delta_2 \wedge \neg\mathtt{t}(0)$.*

The idea of contrastive explanation is dual with abductive explanation, since it points to a minimal part of a situation whose change would falsify the current decision, and

[5] Notice that we have not fully used the expressive power of $[X]\varphi$ and $\langle X\rangle\varphi$ until now for minimality. The intuitive meaning of $\langle Atm(\lambda)\setminus\{p\}\rangle\neg\mathtt{t}(x)$ in the formula is that even if we just perturb one variable p in λ from its actual value, the classification will possibly no longer be x.

the duality between their weak versions is similar [10]. A conjunction of literals λ is a contrastive explanation for outcome x in situation s, if the following conditions are satisfied: (a) λ is true at s, and s has outcome x, (b) if all literals in λ were false then the outcome would be different, (c) λ is the subset-minimal literals satisfiying (a) and (b). A weak contrastive explanation is only based on conditions (a) and (b).

Definition 11 (Contrastive explanation (CXp) and weak contrastive explanation (wCXp))**.** *We write* $\mathtt{CXp}(\lambda,x)$ *to mean that* λ constrastively explains *the decision x and define it as*

$$\mathtt{CXp}(\lambda,x) =_{def} \lambda \wedge \langle Atm_0 \setminus Atm(\lambda)\rangle \neg \mathtt{t}(x) \wedge \bigwedge_{p \in Atm(\lambda)} [(Atm_0 \setminus Atm(\lambda)) \cup \{p\}]\mathtt{t}(x).$$

We write $\mathtt{wCXp}\ (\lambda,x)$ *to mean that* λ weak-contrastively explains *the decision x and define it as* $\mathtt{wCXp}\ (\lambda,x) =_{def} \lambda \wedge \mathtt{t}(x) \wedge \langle Atm_0 \setminus Atm(\lambda)\rangle \neg \mathtt{t}(x)$.

Intuitively speaking, we can test whether λ is a wCXp of situation s having outcome x by "flipping" its positive literals to negative, and negative to positive, and observe if the resulting state is classified differently from x. CXp is the subset-minimal wCXp.

Weak CXps can be used to study the preferences between reasons in a case base. The next proposition indicates that given a precedent (s,X,x), if the absence of Y at s, by itself *alone* can weakly contrastively explain x, then Y is "no weaker than" X in CB.

Proposition 4. *Let CB be a consistent case base and* $(s,X,x) \in CB$*, and* $C \in \mathbf{CM}^{prec}$ *s.t.* $(C,s) \models tr_2(CB)$*. If* $(C,s) \models \mathtt{wCXp}\ (\mathtt{cn}_{\emptyset,Y},x)$*, then it is not the case that* $Y <_{CB} X$.

Example 7. *Let* $C \in \mathbf{CM}^{prec}$ *be a model of* $tr_2(CB^{ex})$*. Since* $\{\delta_3\} <_{CB^{ex}} \{\pi_1\}$*, we have* $(C,s_2) \models \mathtt{wCXp}\ (\mathtt{cn}_{\emptyset,\{\pi_1\}},0)$*. Indeed* $f(\pi_1,\delta_1,\delta_3) = 0$ *by* `2Mon` *according to* s_1.

6. Related Work and Conclusion

In this paper, we have shown that through the concept of classifier a novel logical model of factor-based case-based reasoning can be provided, which allows for a rigorous analysis of case bases and of the inferences they support.

As noted in the introduction, our work is based upon the case-based reasoning models of HYPO and CATO [2,1] and upon the analysis of precedential constraint by Jeff Horty [8,9]. Further approaches exist that make use of logic in reasoning with cases. For instance, [17] provided a factor-based model based on formal defeasible argumentation. More recently [19,20] represent precedents as propositional formulas and compare precedents by (propositional) logical entailment.

However, this propositional representation does not fully use the power of logic, in the sense that it does not provide a proof theory (axiomatics) for reasoning with precedents. By contrast, besides the semantic framework presented here, we can make syntactic derivations of properties of CBR using the axiomatics of BCL (see in Appendix).

Moreover, our representation results allow for exploring different notions of explanation, such as abductive and contrastive explanations. We can accordingly explain why a case-based reasoning suggests a particular outcome (rather then a different one) in a new case. Thus, out model could be used to build explainable case-based reasoners,

which could also be deployed to reproduce and analyze the functioning of opaque predictors of the outcome of cases. Thus, by bringing CBR into the broader context of classifier systems, we connect three lines of research: legal case-based reasoning, AI&Law approaches on to explanation [3], techniques and results developed in the context of XAI.

In future work we will examine more deeply the relation between classifiers, explanations, and reasoning with legal precedents. Interesting developments pertain to addressing analogical reasoning beyond the a fortiori constraint considered here and to deploying ideas of explanation to extract knowledge out of cases (e.g., to determine the direction of factors and the way in which they interact).

References

[1] Vincent Aleven. Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment. *Artificial Intelligence*, 150(1-2):183–237, 2003.
[2] Kevin D. Ashley. *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. MIT, 1990.
[3] Katie Atkinson, Trevor Bench-Capon, and Danushka Bollegala. Explanation in ai and law: Past, present and future. *Artificial Intelligence*, 289:103387, 2020.
[4] Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez, and Pierre Marquis. On the computational intelligibility of boolean classifiers. In *Proceedings of KR 2021*, number 1, pages 74–86, 2021.
[5] Adnan Darwiche and Auguste Hirth. On the reasons behind decisions. In *Proceedings of ECAI 2020*, pages 712–720. IOS Press, 2020.
[6] John Horty. Reasoning with dimensions and magnitudes. In *International Conference on Artificial Intelligence and Law, ICAIL2017*. ACM, 2017.
[7] John F. Horty. The result model of precedent. *Legal Theory*, 10:19–31, 2004.
[8] John F. Horty. Rules and reasons in the theory of precedent. *Legal theory*, 17:1–33, 2011.
[9] John F. Horty and Trevor J. M. Bench-Capon. A factor-based definition of precedential constraint. *Artificial intelligence and Law*, 20:181–214, 2012.
[10] Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, Martin Cooper, Nicholas Asher, and Joao Marques-Silva. Tractable explanations for d-dnnf classifiers. In *Proceedings of AAAI 2022*, pages 5719–5728, 2022.
[11] Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and Joao Marques-Silva. From contrastive to abductive explanations and back again. In *International Conference of the Italian Association for Artificial Intelligence*, pages 335–355. Springer, 2020.
[12] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. Abduction-based explanations for machine learning models. In *Proceedings of AAAI 2019*, pages 1511–1519, 2019.
[13] Xinghan Liu and Emiliano Lorini. A logic for binary classifiers and their explanation. In *Proceedings of the 4th International Conference on Logic and Argumentation (CLAR 2021)*, pages 302–321. Springer-Verlag, 2021.
[14] Xinghan Liu and Emiliano Lorini. A unified logical framework for explanations in classifier systems. *Journal of Logic and Computation*, forthcoming.
[15] Tim Miller, Robert Hoffman, Ofra Amir, and Andreas Holzinger, editors. *Artificial Intelligence journal: Special issue on Explainable Artificial Intelligence (XAI)*, volume 307, 2022.
[16] Henry Prakken. A formal analysis of some factor and precedentbased accounts of precedential constraint. *Artificial Intelligence and Law*, 2021.
[17] Henry Prakken and Giovanni Sartor. Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law*, 6:231–87, 1998.
[18] Andy Shih, Arthur Choi, and Adnan Darwiche. Formal verification of bayesian network classifiers. In *International Conference on Probabilistic Graphical Models*, pages 427–438. PMLR, 2018.
[19] Heng Zheng, Davide Grossi, and Bart Verheij. Case-based reasoning with precedent models: Preliminary report. In *Computational Models of Argument*, pages 443–450. IOS Press, 2020.
[20] Heng Zheng, Davide Grossi, and Bart Verheij. Precedent comparison in the precedent model formalism: theory and application to legal cases. In *Proceedings of the EXplainable and Responsible AI in Law (XAILA) Workshop at JURIX*, 2020.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220452

Reasoning with Legal Cases: A Hybrid ADF-ML Approach

Jack Mumford , Katie Atkinson , and Trevor Bench-Capon
Department of Computer Science, University of Liverpool, UK

Abstract. Reasoning with legal cases has long been modelled using symbolic methods. In recent years, the increased availability of legal data together with improved machine learning techniques has led to an explosion of interest in data-driven methods being applied to the problem of predicting outcomes of legal cases. Although encouraging results have been reported, they are unable to justify the outcomes produced in satisfactory legal terms and do not exploit the structure inherent within legal domains; in particular, with respect to the *issues* and *factors* relevant to the decision. In this paper we present the technical foundations of a novel hybrid approach to reasoning with legal cases, using Abstract Dialectical Frameworks (ADFs) in conjunction with hierarchical BERT. ADFs are used to represent the legal knowledge of a domain in a structured way to enable justifications and improve performance. The machine learning is targeted at the task of factor ascription; once factors present in a case are ascribed, the outcome follows from reasoning over the ADF. To realise this hybrid approach, we present a new hybrid system to enable factor ascription, envisioned for use in legal domains, such as the European Convention on Human Rights that is used frequently in modelling experiments.

Keywords. Abstract Dialectical Frameworks, Argumentation Frameworks, Reasoning with legal cases, Hybrid machine learning-argumentation

1. Introduction

Modelling legal case-based reasoning has been a central concern within the field of AI and Law from its early days e.g. [1]. Many of the approaches that have been proposed and developed over the past three decades have used symbolic techniques since, recognising that when humans make decisions on legal cases they call upon their domain expertise, it was natural to aim to model this expertise in legal AI systems. However, in the past decade there has been an increase in the application of data-driven methods aimed at predicting legal cases, aligning with developments in the general field of AI where machine learning (ML) applications have become increasingly prominent. The concerns about lack of interpretability and explainability of ML systems that are widespread within the general field of AI very much apply to tools that are built to perform legal reasoning; if such tools are to be trusted for deployment in real world scenarios, serious attention must be paid to the their ability to accurately capture laws and legal reasoning in the manner currently required of human experts. Moreover, in law, explanation is essential: natural justice requires that decisions are explained to the parties.

In this paper we present new work, extending on the outline presented in [2], that enables an important step within a wider programme that aims to demonstrate how hybrid

approaches to modelling legal case-based reasoning can be developed to enable learning from large volumes of data, whilst retaining crucial domain knowledge that is needed to reason about and explain automated decisions on legal cases. In section 2 we provide a brief summary review of some of the key contributions to date within the field of AI and law that provide computational models of legal reasoning. In section 3 we provide an overview of how Boolean Abstract Dialectical Frameworks (ADFs) [3] can be used to model knowledge of a legal domain. In section 4 we set out the details of our new hybrid system in which an ADF provides the legal domain knowledge representation, and the H-BERT model is used to learn how to ascribe the base-level factor input for the ADF. Section 5 reports the results of experiments evaluating the performance of our hybrid system. We conclude in section 6 with reflections on our study and its results, along with the next steps for building upon the new foundational hybrid approach that we have presented.

2. Approaches to Modelling Legal Reasoning in AI and Law

The broad process of reasoning with legal cases in common law jurisdictions is undertaken within the context of a body of case law, comprising previous decisions, whereby a new case must be decided in the light of these precedents. At a hearing each side presents arguments as to why they should win and these arguments will typically be based on the precedent decisions and legislation.

Early work on modelling legal reasoning demonstrated how production rules [4] and logic programming [5] could be used to model and explain legal case-based reasoning. A shift of focus to identify the arguments involved in case based reasoning was brought through the development of the HYPO system [6]. This work was built upon in the CATO system [7], which was developed to assist law school students in forming better case-based arguments. A key idea within CATO was to describe cases in terms of *factors*, which are legally significant abstractions of patterns of facts found in the cases of a domain. These factors were then organised within a hierarchy of increasing abstraction, with factors labelled with a polarity to show whether they support or oppose the presence of their parent. Moving upwards through abstract factors within the hierarchy leads to the legal *issues* that have to be resolved to reach a decision in the particular legal domain (US Trade Secrets in the case of HYPO and CATO).

In later work based on CATO, Issue-Based Prediction (IBP) [8] was developed with the aim of not only discovering and presenting arguments, but also predicting the outcomes of cases. Evaluation results showed a good level of accuracy (over 90%) where the domain model relied upon manual analysis, but when machine learning was used for ascribing factors, the accuracy level decreased (to around 70%) [9].

In a further development in the spirit of CATO-style approaches, [10] set out a methodology, called ANGELIC, for capturing case law and explaining conclusions drawn through reasoning over the model. The methodology makes use of a well established knowledge representation technique, Abstract Dialectical Frameworks (ADFs) [3], to capture the factors and relationships between them within a domain of case law. Once defined for a domain, an ADF can easily be transformed into a logic [10] or JAVA [11] program that, when supplied with the facts of a case, can determine an outcome for the case and provide acceptable arguments leading to this decision. A success rate of over

96% accuracy in replicating past decisions was reported in [10], reflecting the high level of domain expertise captured within the ADFs through manual knowledge acquisition tasks undertaken to build the domain model.

In recent years, there has been an increase in work aimed at performing the task of case prediction through the use of data-driven methods that learn from the large datasets now available. Key representative examples are the work presented in [12], [13] and [14]. Problems with these approaches include lack of accuracy (typically around 70-80%), degradation of performance as the training set ages, and lack of explanations. State-of-the-art transformer-based models [15] for NLP tasks are unsuitable for long document classification, and proposed solutions [16] such as hierarchical transformer methods [17] rely on very large data sets and significant pre-training in order to produce decent results. None of these problems apply to symbolic models.

Our motivation for the work set out in this paper is to bring together into a hybrid system benefits found within symbolic and data-driven approaches to legal case-based reasoning. The domain expertise captured within ADFs is vital for grounding and explaining reasoning within the law, yet we wish to make use of machine learning approaches where they can be usefully deployed for the relatively expensive task of factor ascription. In the next section we show how ADFs are used to capture domains, prior to presenting, in Section 4, the use of ML for enabling the ascription of factors as an integral part of our new hybrid method for case classification.

3. Representing Legal Knowledge with ADFs

We represent the Legal Knowledge following the ANGELIC Methodology [10]. The methodology results in an instantiated Abstract Dialectical Framework (ADF) [3] and a set of questions. The instantiated ADF can be represented as a Table in which each node is associated with an ID, an informative label describing the node, a list of the children of the node, and a set of acceptance conditions. Sample nodes from the ADF modelling the European Convention on Human Rights (ECHR) in [18] are shown in Table 1. The acceptance conditions all take the form of "ACCEPT" or "REJECT" followed by a body expression containing only children of the node and the appropriate logical operators. The acceptance conditions are prioritised so that they will be tested in order and when one succeeds, the others will be ignored. The final acceptance condition for a node is always a default, so no node is undecided. The use of disjunctions means that the accept and reject conditions are interleaved. The ADF is accompanied by a set of questions which are posed to the user and which determine which leaf notes are accepted. Thus the leaf nodes have acceptance conditions in terms of the responses to one or more questions, such as "ACCEPT IF $Q7 \geq Q9$".

All the nodes in the ADF are Boolean, accepted or rejected, as in CATO [7]. There has been some work to model the degrees of acceptance (e.g. [19], [20]) but we see this as a matter of factor ascription [21]. The base level factors forming the leaf nodes all represent reasons to decide for one side or the other, and may represent a judgement. Suppose we have a factor *SignificantDelay*. This factor would have an associated question, such as *Q6: Enter the delay in months*. Precedents will have indicated the length of delay considered significant and so, if 18 months was considered significant, we have as the acceptance condition for this leaf node "ACCEPT IF $Q6 \geq 18$". Thus once the ADF is reached, matters of extent are settled and we can deal only with Booleans.

Table 1.: Sample nodes from ADF in [18]

ID	Description	Children	Acceptance Conditions
1	Violation of Article 6	2,3,8,20,21	REJECT IF NOT is a victim OR NOT case is admissible ACCEPT IF the case was not fair or public OR victim was presumed guilty OR the victim did not have the minimum rights REJECT otherwise
4	The case is admissible	4,5	REJECT IF NOT the case is well-founded OR there was no significant disadvantage ACCEPT otherwise

As noted in Section 2, legal domain knowledge can be seen as exhibiting a hierarchical structure, as represented by the factor hierarchy in CATO [7]. At the top is a statement expressing the decision (e.g. *The Plaintiff should win*) that is determined by resolution of the relevant issues (e.g. *There was a confidential relationship*). The issues provide necessary and sufficient conditions for the decision. The issues are resolved by considering the balance between pro and con factors. The factors themselves (e.g. *KnewInfoConfidential*) are ascribed on the basis of the facts of the case (e.g. *The defendant acquired the information while employed by the plaintiff*). These reasoning steps are based upon precedents. There are three different kinds of precedent, as explained in [21], each appropriate to a different statement type. *Framework* precedents, supplement legislation to identify the issues used to resolve the root node. *Preference* precedents determine which way the balance of factors should fall when resolving issues. *Ascription* precedents provide sufficient conditions for assigning factors to a case on the basis of its facts.

This hierarchy can be straightforwardly modelled as an ANGELIC ADF. For the node representing the decision, the necessary and sufficient conditions from legislation and framework precedents are stated in the required form. For the issue nodes, which require a balance of factors in accordance with precedents, the set of preference precedents are translated into rules in the manner devised in [22]. The pro reasons in a precedent form a condition with ACCEPT as head, the con reasons form a condition with REJECT as head and the outcome in the precedent determines the priority between them. The default here reflects the burden of proof for the particular issue. Finally the conditions for the leaf nodes, which have question answers as children, use the ascription precedents to determine whether the factor is present given the facts represented by these answers.

This method of representation has been applied in a number of domains, both academic [10] and practical e.g. [23]. Article 6 of the European Convention on Human Rights (ECHR), which concerns the right to a fair trial, used by the machine learning systems described in [12] and [14], was modelled as an ADF in [18] and [11], on which the work in this paper is based.

4. Hybrid ADF/H-BERT Method

Our approach is to leverage domain knowledge in conjunction with a state-of-the-art H-BERT architecture [24]. We use a Python implementation of the ADF constructed specifically for Article 6 of the ECHR [18] to provide intermediate classifications of base-level factors, and train an independent H-BERT model for each base-level factor.

The ADF provides domain knowledge representation of the legal reasoning process, and the H-BERT models are responsible for learning how to ascribe the base-level factor input for the ADFs from the case documentation. Intuitively, the ADF is interpreted as a final set of fixed weights that is added on top of the H-BERT models.

As we are interested in producing case classifications that can be understood and justified in terms of genuine domain knowledge, it is important that the H-BERT models ascribe base-level factors in a sensible and appropriate manner. In the legal reasoning hierarchy, base-level factors are ascribed according to the facts of the case. ECHR case summaries neatly present such facts as bullets within the document, which enables their extraction via the use of regular expressions whilst processing the document in HTML format. Each document is thus segmented in accordance with the bullets (taken only from the section of the document entitled "THE FACTS", discussed in more detail below) and a pre-trained RoBERTa [25] model is used to produce an intermediate output representation for each bullet. The RoBERTa outputs for each bullet are then used as the inputs for training a subsequent BERT model that classifies the whole document for a particular base-level factor. Each document is encoded according to the RoBERTa model one time only, since the model is pre-trained. However, we must train a unique BERT model for each base-level factor in order to capture the interactions between the segmented fact encodings that are relevant for that specific factor.

The pipeline can be visualised in Figure 1, and can be understood as progressing through six stages:

***Stage 1*:** A corpus is obtained by scraping cases from the HUDOC website[1] – the principal repository for ECHR legal case documentation. Cases are scraped in HTML format in order to facilitate effective processing.

***Stage 2*:** Each case in the corpus is processed and segmented to the fact level of representation, understood to correspond to the bulleted points in the document, which can be extracted from the HTML format. The text is further processed to remove irrelevant bullets, such as headings and enumeration markers. Only text from THE FACTS section is taken as pertaining to the fact level; other sections provide information either irrelevant to justifiable legal reasoning (e.g. the identities of the judges) or contain the reasoning itself, which would defeat the purpose if included.

***Stage 3*:** For each case in the corpus, each fact segment is tokenized and then encoded by a pre-trained RoBERTa model, where the same model is used for all segments, without fine-tuning.

***Stage 4*:** The RoBERTa encodings of a case document's fact segments are used as input for the BERT models, where we have thirty-two BERT models – one for each base-level factor of the ADF (excluding admissibility related factors that are irrelevant for our corpus which consists solely of admissible cases). Each BERT model further encodes each RoBERTa input to provide overall document context.

***Stage 5*:** The BERT encodings are used as input in a feed-forward neural network (FFNN) that outputs a Boolean classification which is used to determine ascription or non-ascription of a particular base-level factor in the ADF.

***Stage 6*:** The Python program implementing the ADF then produces the outcome classification that follows from the base-level factor ascriptions.

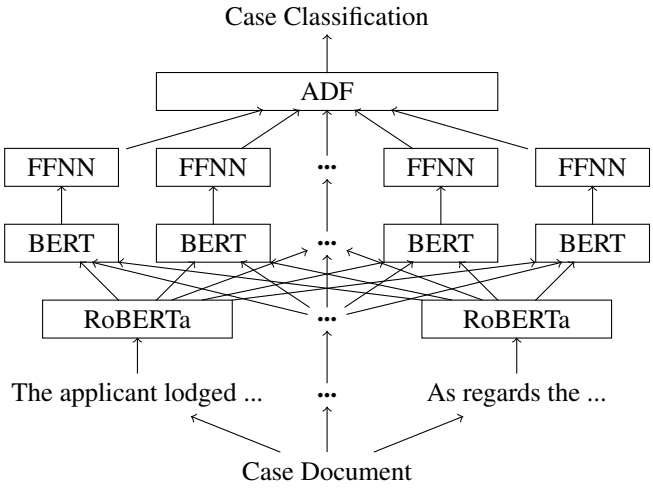


Figure 1. Feed-forward hybrid H-BERT/ADF model for ECHR Article 6 case classification.

During training, learning applies to adjusting weights only in the BERT models according to classifications provided from the ADF, which are in turn derived from processing the correct case outcome classification. The process extends through four steps:

***Step 1*:** Each case in the corpus is labelled according to its correct outcome classification: violation, or no-violation.

***Step 2*:** This classification is fed backwards through the ADF to produce probabilistic weights for the base-level factors, where the weight for ascription of any given base-level factor is the proportion of instances in which the factor is ascribed in correct outcome classification over all possible combinations of ascription.

***Step 3*:** Each base-level factor weight is used as the probability for ascribing the factor for classification with the relevant FFNN. For example if a base-level factor was assigned a probabilistic weight from the ADF of 0.2, then there would be a 20% chance that the FFNN would be provided a True Boolean classification, and 80% chance of a False Boolean classification.

***Step 4*:** The FFNN passes weight adjustments back to the appropriate BERT model, which in turn adjusts its weights accordingly. Learning does not propagate down to the RoBERTa models, which are fixed to their pre-training settings.

5. Experiments

In this section we assess the performance of our hybrid system on the legal case classification task. We first outline the details of the case corpus data set, and give the implementation details for our system[2]. We also compare the performance of the hybrid system against a H-BERT benchmark[3].

[1] hudoc.echr.coe.int/

[2] Undertaken on Barkla – High Performance Computing facilities, at the University of Liverpool, UK.

[3] Code available at: https://github.com/jamumford/LADF-HBERT.git

5.1. Data Set and Implementation Details

To build a relevant data set for experimentation necessitated the use of cases from January 2015 onwards, as the ADF was developed from official documentation and expert opinion relevant from this particular time point. As the law is subject to change over time, we decided to omit cases prior to January 2015 from analysis, due to the risk of incompatibility with the ADF model. We also restricted analysis to cases available only in English (note that when using the HUDOC site's filter for cases in English, roughly one in four of the filtered cases were incorrectly formatted and not available in English), which resulted in 575 cases between January 2015 and January 2022: 150 non-violation verdicts, and 425 violation verdicts.

The scarcity of the data is in stark contrast to the relatively vast data sets that are usually employed for NLP tasks. We focus on two classification approaches, a state-of-the-art hierarchical BERT approach developed specifically for small data sets which we refer to as H-BERT [24], and our hybrid system which uses the aforementioned H-BERT architecture in conjunction with the ADF layer as outlined in section 4. Both approaches use the same fact-level pre-trained RoBERTa model encodings using 512 tokens, and both use 256 tokens for document BERT model encoding.

Three metrics were selected for analysis as appropriate for the binary classification task: accuracy, macro F1 score, and MCC (Matthews correlation coefficient) score. Since the data set is unbalanced between non-violation and violation verdicts, the macro F1 and the MCC scores are of clear relevance, with the MCC score particularly suited to providing a representative score for performance over the full distribution.

In total forty experiments were conducted, twenty for each approach. Each experiment was randomly split 80% into training and 20% into test data, and underwent 30 epochs of training. Four Nvidia Tesla P100 GPUs were used for fine-tuning.

For the hybrid H-BERT/ADF system, in each epoch the training data was further split such that 10% was randomly assigned to a validation data set. At the end of an epoch the validation data set's MCC score was used to scale the factor ascription weights from the ADF layer. Hence a higher validation set MCC score would increase the likelihood of preserving the previous epoch's factor ascriptions for the subsequent epoch, whereas a lower MCC score would decrease the likelihood.

5.2. Results and Evaluation

The experimental results are summarised in Table 2 and provide a comparison of the performance of the H-BERT model against our hybrid H-BERT/ADF system. For all metrics, the hybrid system demonstrably outperforms the benchmark H-BERT model. These results can be reliably inferred as statistically significant, with negligible Mann-Whitney p values returned for each metric. The values $p < 0.001$ indicate that the hypothesis that the distributions for the two classification approaches (across all experiments) belong to the same population can be rejected at the 99.9% confidence level (and indeed the p values are so low that the hypothesis can be rejected at even higher confidence levels).

There is a greater degree of variance of results for the hybrid system in comparison with the H-BERT results, as indicated by the negative and positive range values. This greater variance is likely due to the inherent uncertainty of the thirty-two classification targets derived from the factor ascription weights produced by the ADF layer. Since the

Table 2.: Comparison of standard H-BERT model performance against Hybrid H-BERT/ADF system on ECHR Article 6 case outcome classification task. Results reported to 2d.p.

	Accuracy	Macro F1	MCC
H-BERT [24]	$66.78^{+0.17}_{-0.70}$	$60.16^{+0.23}_{-0.04}$	$17.90^{+0.03}_{-0.10}$
Hybrid H-BERT/ADF	$\mathbf{72.00}^{+8.87}_{-7.65}$	$\mathbf{67.47}^{+8.72}_{-5.58}$	$\mathbf{33.98}^{+18.66}_{-11.48}$
Mann-Whitley p values	< 0.001	< 0.001	< 0.001

factor ascription weights are treated as probabilities to ascribe a factor for a given case, each of the classification targets for each BERT model in the hybrid system will likely change over epochs, whereas they remain constant in the benchmark H-BERT approach.

Each H-BERT experiment required roughly five minutes of training for our given data set. Our hybrid system required roughly four hours, due mainly to the thirty-two H-BERT models (one for each base-level factor) that were necessary to train. However, it is worth noting that training only needs to be done once, and the resulting model could be used without the need for frequent updates, since the law changes relatively slowly in comparison to these processing times. Once training is completed, running a single case for classification requires negligible time (less than one second) when considered against practical timescales for evaluation.

6. Discussion and Summary

In our approach to argument-based modelling of legal reasoning, we have identified a particular role for machine learning: factor ascription rather than predicting cases as a whole, with ascription done by Hierarchical BERT applied to natural language descriptions of the case facts. Prediction is still possible, since the factors in a case determine the outcome. Law changes over time; models that cover a wide period of time are unlikely to be of great relevance in terms of producing justifiable outcomes. However, most ML approaches to case classification have focused on large data sets, which is understandable since state-of-the-art performance in NLP typically requires an abundance of data. But this is likely to set an upper limit on the usefulness of such models.

Factors are vital for acceptable explanations, since the justification of a case outcome is given in terms of how the issues were resolved considering the balance of factors present in the case. Use of factors for explaining the output of machine learning systems has been advocated in [26] and [27]. Exploiting the domain structure and learning factor ascription rather than case prediction should also improve performance. While formal approaches summarised in [28] treat cases as collections of factors, empirical approaches such as [8] and [29] decompose cases into issues. As shown in [21] applying precedential constraint at the level of cases rather than issues results in cases which should be constrained being not constrained, because they can be distinguished on factors unrelated to the issue in dispute in the case. This is also borne out by empirical work. In [8], while issue based prediction abstained on only one of the 186 test cases enabling accuracy of 91.4%, a similar system which treated the outcome as a single issue abstained on 50, reducing accuracy to 68.3%, a level in line with the legal machine learning approaches reported in [30]. A further reason for providing explanation in terms of factors is provided

by [31], in which it was shown that a good level of performance by a machine learning system was no guarantee that it was applying the correct rationale when making its predictions. In the light of this, avoiding injustices and securing trust in the predictions demands an explanation, couched in legal terms, of the outcome. This in turn demands that the factors in a case be identified. The approach taken in this paper is broadly aligned with the observations made in [32], that cognitive computing for legal application will involve the interaction between an expert-derived factor-based model for higher reasoning and ML for lower-level processing of the raw document text. In future research we would like to assess our hybrid system in terms of explainability and justifiability of any given classification/prediction, via the fact-level attention weights produced by the individual H-BERT models. These attention weights might be useful for improving performance, presenting a feedback loop where experts approve or reject the facts selected by the system to justify factor ascription.

It should be noted that the effort involved in constructing the domain model has been found to be not disproportionate when applied to real world problems working with legal practitioners (e.g. [23]). An obstacle to practical deployment, however, is the effort required for manual analysis of cases into factors, both for constructing the case base and representing the new cases. Thus using machine learning to address the labour intensive task of ascribing factors to cases, while reserving the construction of the domain for experts with the appropriate knowledge, seems the sensible way to allocate resources. The time taken to train and use our hybrid system is fully in keeping with these expectations.

Our new H-BERT/ADF system lays the foundation for a hybrid approach to automating reasoning about legal cases, using both symbolic and data-driven techniques. Structured legal domain expertise is captured using the ANGELIC ADFs described in Section 3, which enable explanations of the outcomes of reasoning. Our ultimate aim is to use machine learning for the task of factor ascription, since factors must be ascribed for every case. We consider the results outlined in this paper as encouraging in moving towards this aim. Our next steps will be to incorporate higher levels of domain knowledge via the use of annotated data sets consisting of cases labelled by domain experts with respect to factor ascription. These annotations will be used to remove some of the uncertainty of the classification labels presented to the H-BERT models, both directly for the labelled cases, and indirectly by suggesting better priors to guide the ADF layer's weight propagation for factor ascription in accordance with the background distribution of factor ascriptions from the annotated data set. Comparison to alternative ML approaches in the literature was not conducted in the analysis due to divergent data sets and scope of focus. However, future work would benefit from direct comparison against a wider array of benchmarks. We also want to explore other means of capturing domain knowledge in the learning process, which we argue is likely to be essential for deriving effective and justifiable models, such as via the incorporation of semantic-search methods.

References

[1] McCarty LT. Reflections on TAXMAN: An experiment in Artificial Intelligence and legal reasoning. Harvard Law Review. 1976;90:837.

[2] Mumford J, Atkinson K, Bench-Capon T. Explaining Factor Ascription. In: Proceedings of JURIX 2021. IOS Press; 2021. p. 191-6.

[3] Brewka G, Ellmauthaler S, Strass H, Wallner J, Woltran P. Abstract Dialectical Frameworks revisited. In: Proceedings of the 23rd IJCAI. AAAI Press; 2013. p. 803-9.

[4] Gardner A. An Artificial Intelligence Approach to Legal Reasoning. MIT Press; 1987.
[5] Sergot M, Sadri F, Kowalski R, Kriwaczek F, Hammond P, Cory H. The British Nationality Act as a logic program. Communications of the ACM. 1986;29(5):370-86.
[6] Ashley KD. Modeling legal arguments: Reasoning with cases and hypotheticals. MIT press; 1990.
[7] Aleven V. Teaching case-based argumentation through a model and examples. Univ. of Pittsburgh; 1997.
[8] Bruninghaus S, Ashley K. Predicting outcomes of case based legal arguments. In: Proceedings of the 9th ICAIL; 2003. p. 233-42.
[9] Ashley KD, Brüninghaus S. Automatically classifying case texts and predicting outcomes. AI and Law. 2009;17(2):125-65.
[10] Al-Abdulkarim L, Atkinson K, Bench-Capon T. A methodology for designing systems to reason with legal cases using ADFs. AI and Law. 2016;24(1):1-49.
[11] Atkinson K, Collenette J, Bench-Capon T, Dzehtsiarou K. Practical tools from formal models: the ECHR as a case study. In: Proceedings of the 18th ICAIL; 2021. p. 170-4.
[12] Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. Predicting judicial decisions of the ECHR: A natural language processing perspective. PeerJ Computer Science. 2016;2:e93.
[13] Chalkidis I, Fergadiotis M, Malakasiotis P, Aletras N, Androutsopoulos I. LEGAL-BERT: The muppets straight out of law school. arXiv preprint arXiv:201002559. 2020.
[14] Medvedeva M, Vols M, Wieling M. Using machine learning to predict decisions of the European Court of Human Rights. AI and Law. 2019:1-30.
[15] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.
[16] Dai X, Chalkidis I, Darkner S, Elliott D. Revisiting Transformer-based Models for Long Document Classification. arXiv preprint arXiv:220406683. 2022.
[17] Zhang X, Wei F, Zhou M. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. arXiv preprint arXiv:190506566. 2019.
[18] Collenette J, Atkinson K, Bench-Capon T. An Explainable Approach to Deducing Outcomes in European Court of Human Rights Cases Using ADFs. In: Proc. of COMMA 2020; 2020. p. 21-32.
[19] Horty JF. Reasoning with Dimensions and Magnitudes. In: Proceedings of the 16th ICAIL; 2017. p. 109-18.
[20] Bench-Capon T, Atkinson K. Lessons from Implementing Factors with Magnitude. In: Proceedings of JURIX 2018; 2018. p. 11-20.
[21] Bench-Capon T, Atkinson K. Precedential constraint: The role of issues. In: Proceedings of the 18th ICAIL; 2021. p. 12-21.
[22] Prakken H, Sartor G. Modelling reasoning with precedents in a formal dialogue game. AI and Law. 1998;6(3-4):87-231.
[23] Al-Abdulkarim L, Atkinson K, Bench-Capon T, Whittle S, Williams R, Wolfenden C. Noise induced hearing loss: Building an application using the ANGELIC methodology. Argument & Computation. 2019;10(1):5-22.
[24] Lu J, Henchion M, Bacher I, Namee BM. A sentence-level hierarchical bert model for document classification with limited labelled data. In: Proceedings of DS 2021. Springer; 2021. p. 231-41.
[25] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:190711692. 2019.
[26] Branting LK, Pfeifer C, Brown B, Ferro L, Aberdeen J, Weiss B, et al. Scalable and explainable legal prediction. AI and Law. 2021;29(2):213-38.
[27] Prakken H, Ratsma R. A top-level model of case-based argumentation for explanation: Formalisation and experiments. Argument & Computation. 2021:1-36.
[28] Prakken H. A formal analysis of some factor-and precedent-based accounts of precedential constraint. AI and Law. 2021:1-27.
[29] Grabmair M. Predicting trade secret case outcomes using argument schemes and learned quantitative value effect tradeoffs. In: Proceedings of the 16th ICAIL; 2017. p. 89-98.
[30] Chalkidis I, Jana A, Hartung D, Bommarito M, Androutsopoulos I, Katz D, et al. LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. arXiv preprint:211000976. 2021.
[31] Steging C, Renooij S, Verheij B. Discovering the Rationale of Decisions: Experiments on Aligning Learning and Reasoning. arXiv preprint arXiv:210506758. 2021.
[32] Ashley KD. Artificial intelligence and legal analytics: new tools for law practice in the digital age. Cambridge University Press; 2017.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220453

A Multi-Step Approach in Translating Natural Language into Logical Formula

Ha-Thanh Nguyen [a] Fungwacharakorn Wachara [a] Fumihito Nishino [a] Ken Satoh [a]
[a] *National Institute of Informatics, Japan*
nguyenhathanh@nii.ac.jp

Abstract. Translating often has the meaning of converting from one human language to another. However, in a broader sense, it means transforming a message from one form of communication to another form. Logic is an important form of communication and the ability to translate natural language into logic is important in many different fields, in which logical reasoning and logical arguments are used. In the legal field, for example, judges must often reason from facts and arguments presented in natural language to logical conclusions. In this paper, toward the goal of support for this kind of reasoning with machines, we propose a method for translating natural language into logical representations using a combination of deep learning methods. Our approach contributes methodologies and insights to the development of computational methods for converting natural language into logical representations.

Keywords. deep translation, natural language, logical representation

1. Introduction

Although the law is written in natural language, this form is probably not the best option. First, natural language is ambiguous. This means that there can be more than one interpretation of what a sentence in law means [1]. This can lead to confusion and can make it hard to enforce the law. Second, natural language is constantly changing [2]. Words can change meaning over time, which can become a challenge in understanding old laws. Finally, natural language is not the same in every country. This can make it hard to write laws that can be enforced internationally. Representing the law in a logical form, such as first-order logic, can help to solve these problems. First-order logic is a formal language that can be used to write down laws in a way that is clear and unambiguous. First-order logic is also not subject to the same problems as natural language. This is because first-order logic is a fixed and well-defined language. This means that the meaning of the formula in first-order logic is the same in every country.

Using logic programming to support judges and attorneys has many benefits. Perhaps the most significant benefit is that it can help to ensure that the law is applied consistently across cases. By representing the law using formal logic, it becomes possible to develop an automated system that can check for inconsistencies and errors. As a result, the quality of legal decision-making can be improved. Another benefit of using logic programming to support judges and attorneys is that it can help to make the law more

accessible to the general public. By representing the law using formal logic, it becomes possible to develop systems that can generate plain-language explanations of the law. This can help to increase public understanding of the law and improve compliance with the law. Finally, using logic programming to support judges and attorneys can help to make the law more efficient. By representing the law using formal logic, it becomes possible to develop systems that can automatically generate legal documents. This can save significant time and effort for both judges and attorneys.

Despite the above benefits, writing logical formula is a huge challenge. Writing correct logical expressions for law sentences requires expertise and experience. No matter how powerful the system is, without correct inputs, the result can be incorrect. In addition, the result of the reasoning system might be overwhelming to users which are not familiar with logical reasoning. Therefore, using law expert systems that are based on logic programming is not a trivial task. The most difficult requirement is to design the system so that law experts, usually those who are not specialized in logic, can give input to the system. Efforts to date such as controlled natural language or existing translation systems have not met this requirement.

The ideal system should translate both legal rules and case descriptions written in natural language sentences into logical formulas so that judgement could be fully automatically made. However, as far as legal rule translation is concerned, it is usually very difficult to translate legal rules written in natural language into logical formulas. It is because there are some hidden conditions only derived from legal interpretations of the whole set of legal rules and this information only can be obtained from legal literature, not from legal rules themselves. We, therefore, take the intermediate approach for producing judgment as follows.

1. We manually translate legal rules into logical formulas with the consultation of legal scholars and legal literature.
2. We translate case descriptions into logical formulas (fact formulas) automatically.
3. We reason about judgement by applying logical rule formulas to logical fact formulas.

For step 2, the problem is how to extract relevant information from case descriptions. One approach would be pattern-based information retrieval from case descriptions [3]. However, their approach is not robust, that is, if sentences do not follow pre-designed patterns, we cannot extract information, and preparing for various patterns manually will be very time-consuming.

In this paper, we take a robust approach to retrieve relevant fact formulas from a case description written in natural language using large pre-trained language models. To do so, we need to answer the following two questions: First, we want to see whether a large language model trained on a large amount of natural language text can be used to automatically generate logical formulas that are consistent with the text after some fine-tuning. Second, we want to see the weakness of this approach by looking at the types of errors the model makes. To answer the first question, we use large pretrained language models to translate natural language into logical formulas. To answer the second question, we examine the types of errors the model makes. We find that large pretrained language models can be used to automatically generate logical formulas that are consistent with the text after some finetuning. However, with the translation approach, a few slight mistakes in the prediction results can cause syntax errors in logical formulas, which is

tolerable in natural language translation. This information can be used to prepare necessary resources and improvements for the model.

Our main contribution is a novel approach to reason about judgments using manually translated logical rule formulas and automatically translated logical fact formulas. In particular, for automatic translation from case description into logical facts, we combine translation and correction of natural language into a logical formula using deep learning. To this end, we develop an effective deep learning framework with an appropriate training strategy to perform the translation and correction with PROLEG syntax[4]. We conduct experiments to verify the effectiveness of our approach and discuss the insights we gain from the experiments. The structure of this paper is as follows. In Section 2, we have preliminaries covering the basics of the existing logical form in English and the current status of machine translation methods. Section 3 provides a description of the proposed multi-step translation approach. In Section 4, we show the experiments and evaluation of the proposed model. Finally, we conclude in Section 5 and discuss some future works.

2. Backgrounds

2.1. Controlled Natural Languages

Controlled natural languages have been long invented for reducing ambiguity in natural languages. Basically, they are natural languages with restrictions of vocabulary and grammar so that sentences in such languages can be translated into unique logical expressions.

Attempto Controlled English (ACE) [5] is one notable controlled natural language which has been used for representing laws and regulations [6]. Each sentence in ACE can be translated into a first-order expression. For instance, the sentence *Every household creates some garbage* can be translated into $\forall x[household(x) \rightarrow \exists y[garbage(y) \wedge create(x,y)]]$. To achieve this, ACE restricts its vocabulary and grammar, for example, every common noun must occur in a noun phrase with a quantifier. Hence, some acceptable sentences in natural English are unacceptable in ACE. For instance, the sentence *Every household should pay tax for garbage* is unacceptable in ACE because there are no qualifiers for *tax* and *garbage*. Furthermore, strict interpretation rules of ACE may lead to unintended interpretations. For instance, since a pronoun in ACE refers to the most recent noun, *its garbage* in the sentence *Every household should pay some tax for its garbage* is interpreted to *the tax's garbage*, which is counterintuitive. In addition, it is hard to link the complex forms of words to their simple form. For instance, it is hard to link *taxation* to *tax*. Hence, simple forms are preferred to complex forms in ACE.

Logical English [7] is one more recent controlled natural language which has been used for representing laws and regulations [8]. Each sentence in Logical English can be translated into a Prolog rule. In the same manner to ACE, Logical English restricts its vocabulary and grammar, for example, the first occurrence of a common noun phrase must begin with *a/an* and the repeated noun phrase in the same sentence must begin with *the*. It is reported in [8] that Logical English attempts to serve as a syntactic sugar to make logic programs understandable for lay persons; however, it is still hard to write readable Logical English sentences. Assistive tools for writing in Logical English are being developed.

2.2. Existing Machine Translation Methods

A dictionary-based approach to automated translation relies on a set of lexical items, each mapping a concept to a set of alternative meanings. These dictionary entries are written by human experts and are often based on the structure of the source language [9]. In many cases, however, the mapping of concepts to a set of alternative meanings is not one-to-one. In these cases, the dictionary-based approach may require a human expert to choose the appropriate meaning for each concept. Another weakness of this approach is that it can not deal with the variance in wording that is characteristic of natural language. It is impossible to write a dictionary that covers all of the possible ways that a concept can be expressed.

Template methods generate translated sentences following a linear, rule-based approach [10]. This approach is also known as a finite state machine. The template method consists of a set of rules for different sentence templates, which are instantiated to input a sentence and generate a sentence. The simplest form of sentence generation is a sequence of replacement rules. This simple form is not sufficient for real-world applications and data-driven approaches are proposed. A data-driven approach is built using a corpus of parallel sentences, which can be used to build either a translation rule or a translation model.

The statistical approach to machine translation is based on the hypothesis that the translation of a text is a function of the source text and its context. This approach overcomes the problem of dictionary-based and rule-based machine translation by using a statistical model that can deal with the complexity of natural languages. Learning from data, the statistical approach can automatically discover the latent rules that govern the relationship between the source text and the target text. As a result, this approach does not require the knowledge of linguistics or the development of dictionaries and rules. In the era of big data, the statistical approach has become the most popular approach to machine translation with the development of neural machine translation. Large language models like BERT [11], BART [12], T5 [13] and GPT-3 [14] have achieved state-of-the-art results in various natural language processing tasks, including machine translation.

Although the problem of machine translation has been studied for a long time, the studies that directly solve the problem of converting from natural language to logical expressions are quite limited. The main reason is the lack of a large-scale dataset and a method for end-to-end learning from input natural language to output logical expression. In the software industry, converting natural language to another form of software resource like database schema receives more attention than logical expressions. Therefore, it becomes easier to find a large-scale dataset for database schema [15]. The second reason is that to be meaningful, the logical formula should be well defined in a formal language containing definitions and rules. Building such a language consumes a lot of effort in both research and development. Fortunately, PROLEG [4] provides a language for representing logical expression in the legal domain. With that condition, this work can be considered the first step to solving the problem of converting natural language to logical expressions automatically. In the domain of law, current studies about translation are not about machine translation but about human translation. In their paper, Witt et al. [16] provide a detailed analysis of the translation problem in the context of legal texts and propose approaches for improving human translation of legal texts into logical formulas.

3. Method

3.1. General Framework

Solving the problem of translating natural language into a logical formula, we need to deal with two challenges. The first one is the limitation of training data. There is no available dataset that can be used to directly train a model for this task. Furthermore, the professional annotators who can provide high-quality data are also limited. The second challenge is that the logical syntax is much more rigid than the natural language. The translation should be very precise in order to get the correct logical formula. We find that this is an interesting and challenging problem that has not been well explored in the field of natural language processing and deep learning.

We propose the general framework as in Figure 1. There are two main modules: translation and correction. Both of them are pretrained language models. By the universal approximation theorem, all the functions can be approximated by a very deep neural network. Theoretically, we only need one neural network for translation. However, in the context of limited training data, we need to have an appropriate approach to make the system more robust. The translation module is responsible for the translation of the source sentence into the target sentence. The correction module is responsible for the final correction of the translation result. The training data for the translation module is required more expert knowledge and human effort to annotate, but the training data for the correction module is much easier to obtain. This is the key idea that makes our method more effective and efficient, making the goal feasible with limited expert annotations. We present in detail how to prepare the training data for translation and correction modules in Section 3.2.

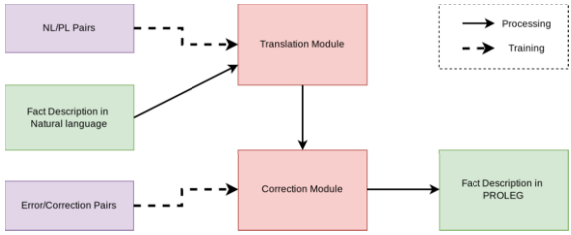


Figure 1. The general framework of the system.

3.2. Data Preparation

To prepare the data for the translation module, we have experts construct the pairs of fact descriptions in natural language and in logical representation (i.e. PROLEG). The logical representations need to be syntactically correct and follow the PROLEG grammar; however, the natural language descriptions can have multiple variations. As an initial investigation, we ask the experts to create logical representations for 15 different scenarios and their variance. Finally, we have 150 logical representations and their corresponding natural language descriptions. An example of data is shown in Table 1.

With a relatively small set of 150 samples, we do not expect that the translation module will be able to cover all the possible variations of the natural language descriptions.

Input
This Room Rental Agreement(this “Agreement”) is made and executed on the next business day, by and between Sanna Mirella Marin(hereinafter referred to as “Landlord”); and LEITCH Michael(hereinafter referred to as “Tenant”). In consideration whereby the Landlord leases the leased premises with the address located at Sunrise Village, Arkansas, 72207 United States. In consideration that the lease term shall commence on October 15, 2018 with the agreed payment term by which the tenant shall make the payment on a monthly basis. The Tenant shall pay the amount of $5 per month to the Landlord on an agreed payment basis.
Output
- Lessor(“Sanna Mirella Marin”). - Lessee(“LEITCH Michael”). - agreement_of_lease_contract(“Sanna Mirella Marin”,“LEITCH Michael”,“Room”,“Sunrise Village, Arkansas, 72207 United States”,“$5 per month”,“2018 year 10 month 15 day”,_, “the next business day”,“This Room Rental Agreement”).

Table 1. Sample of data for translation module.

Strategy	Original	Error
Random capitalization	This is a sample sentence	this is a sAMple sentence
Random split	This is a sample sentence	Th is is a sample sent ence
Random removing	This is a sample sentence	This is a smple setence
Random adding	This is a sample sentence	This is a saample senjtence
Random replacing	This is a sample sentence	This is ant sample sentence

Table 2. Example of error generation strategies.

Interestingly, its errors follow some patterns, which can be learned by another neural network module (the correction module). Data preparation for the correction module can be done in several automatic strategies with examples described in Table 2. We generate a dataset of 20,000 samples, which is much larger than the training set of the translation module.

3.3. Training

In the framework described in Figure 1, the translation module and correction module are trained in a sequential manner:

- Train the translation module with the 150 samples constructed in Section 3.2.
- Train the correction module with the generated incorrect logical representations.
- Use the trained translation module and correction module to translate new natural language descriptions into logical representations.

We do not train the models from scratch but make use of large version of BART to initialize the parameters of our models. We train the translation module and the correction module with early stopped by validation set.

4. Experiments

4.1. Evaluation Metrics

We implement two metrics for this typical translation task. The first metric is exact match accuracy and the second is the longest common non-continuous subsequence (LCNS). We do not use BLEU [17] or ROUGE [18] or other standard translation metrics because these are not suitable for the problem of translating natural language to logical formulas. With these metrics, there need to be several reference translations from which the translation can be judged. In our case, we have only one reference translation. The syntax of logical formulas is very rigid and they do not accept variance in the translation. The exact match accuracy can let us know the percentage of translation units that are exactly matched with the reference translation (ready for the application) and the LCNS can tell how far the current performance is from the ideal case. The exact match evaluation is calculated by Formula 1.

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}_{pred_i=gold_i} \tag{1}$$

To compute the LCNS evaluation, we implement a dynamic algorithm to find the longest common non-continuous substring of two strings *pred* and *gold* as in Algorithm 1:

Algorithm 1 LCNS's algorithm

procedure LCNS($gold, pred$) ▷ lcns of gold and pred
  $m \leftarrow$ `length of` $gold$
  $n \leftarrow$ `length of` $pred$
  L `size of` mxn ▷ dynamic programming matrix
  for $i \leftarrow 1,2,...,m$ `and` $j \leftarrow 1,2,...,n$ **, do**
    if $pred_i = gold_j$ **then**
      $L_{i,j} \leftarrow L_{i-1,j-1} + 1$
    else
      $L_{i,j} \leftarrow \max\{L_{i-1,j}, L_{i,j-1}\}$.
    end if
  end for
  return $L_{m,n}$ ▷ The lcns is $L_{m,n}$
end procedure

The LCNS evaluation value is the division of LCNS($gold, pred$) by the length of $gold$.

4.2. Experimental Results

Table 3 shows the experimental result of the translation system with and without the correction module. With a relatively small dataset, the translation module itself can produce a translation that has about 45% of exact match accuracy with the reference translation

Table 3. The experimental result of the translation system with and without correction module

System	Exact Match	LCNS
Translation Module w.o. Correction Module	0.45	0.94
Translation Module w. Correction Module	0.58	0.97

and 94% of LCNS. The correction module can further improve the translation quality to 58% of exact match accuracy and 97% of LCNS. We can have three observations from these experimental results:

- The translation module alone can produce a translation with fairly high quality.
- The correction module can further improve the translation quality. The improvement is not that significant given the small training dataset, but it is significant enough to show that the correction module has the potential to improve the translation quality.
- The translation quality can be further improved if we have a larger training dataset.

While the Exact Match and LCNS metrics show us the overall performance and improvement of the system, it's hard to imagine how good the system really is and what errors the correction module can detect and correct. Therefore, in Section 4.3, we analyze the specific cases that we observed in our experiment.

4.3. Error Analysis

Table 4. Translation examples for the translation system with and without correction module

	Expected/Generated Formula	LCNS
Gold Translation	fact_of_duress(personC,personB,rescission(contract0),1998 year 5 month 25 day).	-
w.o Correction Module	fact_of_duress(personC,personB,resCission(Contract0),1998 year 5 month 25 day).	0.97
Full System	fact_of_duress(personC,personB,rescission(contract0),1998 year 5 month 25 day).	1.00
Gold Translation	manifestation_fact(rescission(contract0),personA,personB,2030 year 12 month 1 day).	-
w.o Correction Module	manifeStation_fact(resCission(Contract0),personA,personB,2030 year 12 month 1 day).	0.96
Full System	manifestation_fact(rescission(contract0),personA,personB,2030 year 12 month 1 day).	1.00
Gold Translation	contract(personB,personA,this_real_estate,300,1995 year 11 month 13 day,contract0).	-
w.o Correction Module	contract(personB,personA,this_real_ estate,300,2000 year 11 month 13 day	0.81
Full System	contract(personB,personA,this_real_estate,300,2000 year 11 month 13 day	0.81

Table 4 shows some translation examples for the translation system with and without the correction module. The reference translation is provided in the first column. The translation generated by the translation module is in the second column and the translation generated by the translation module with the correction module is in the third column. We can see that the system can extract the key information from the input sentence and generate a translation with fairly high quality. Although the predicates in the target logical form do not appear in the input sentence, the system can produce a correct translation by reusing the existing predicates in the training dataset. In the version with solely the translation module, the system sometimes misspells the predicates and variable names in the logical formula. The correction module can correct some of these errors. In some cases, the correction module can also correct the case of the predicate names and variable names, which is important for the correctness of the logical formula. In the case of wrong date translation, the correction module based on noised generation can only reformat the date but can not assure the correctness of the date.

Looking at Table 4, we see that in the first two examples, the translation module produces pretty good but not perfect outputs. Words like "rescission", "contract" are miscapitalized as "resCission", "Contract". These errors are insignificant to humans, and may not be detected at a glance. The LCNS metric also shows that these translations are almost identical to the original (0.96-0.97). However, they are serious errors for the reasoning engine, which is very strict in syntax. Correction module is effective in cases like this.

However, in the last example, where the translation module completely distorts the content of the argument, the correction module loses its effect. As we see, it can only remove an redundant space in the variable "this_real_ estate" and this is of no value in terms of formula correction. The logical expression is still both syntactically and semantically wrong. These analyzes give us suggestions for the design of models focusing on recognizing the arguments, which will be implemented and introduced in future work.

4.4. Discussions

From the experimental results, we can see that the translation system can achieve fairly high quality with a small amount of training data. However, the system still has some limitations. First, in our current dataset, the number of predicates and variables is limited. Therefore, the translation system can not generate a logical formula containing predicates and variables that are very different from the predicates and variables in the training dataset. Second, the translation system is not robust in recognizing the dates. In the third output in Table 4, the translation system translates the date "1995 year 11 month 13 day" to "2000 year 11 month 13 day". This result should be due to the lack of date recognition skill in the model and overfit to the training dataset. The correction module can not deal with this kind of error. Third, the system sometimes returns an output that is not complete. Also in the third output, the function "contract" does not have all required arguments and the closing ")" is missing. In future work, the combination of template-based fill-in-slot training is a promising method to improve the completeness of the output. In addition, a larger dataset and more types of predicates and variables is required to improve the robustness of the translation system.

5. Conclusions

In this work, we propose a translation system that translates natural language text into logical formulas. We use PROLEG syntax as our target language. As an investigation step, we hire experts to create a small translation dataset from natural language to logical formulas. To improve the translation performance, we propose the multi-step framework by appending the correction module to deal with translationś errors. The correction module is constructed by learning a reverse function from the error generator. The experimental results show that this design can improve the translation quality. The translation system with the correction module can achieve 58% of exact match accuracy and 97% of LCNS in the current small dataset. This work is a promising first step toward automatically translating natural language into logical formulas. To make the system more robust, we need to either improve the quality of the data used to train the model or use a more sophisticated approach to handle the errors. In future works, we want to experiment with other approaches inspired by other work on neural machine translation.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103, JP19H05470 and JP22H00543 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4, Japan.

References

[1] Allen LE, Lysaght LJ. Modern logic as a tool for remedying ambiguities in legal documents and analyzing the structure of legal documents' contained definitions. In: Logic in the Theory and Practice of Lawmaking. Springer; 2015. p. 383-407.
[2] Christiansen MH, Kirby S. Language evolution. OUP Oxford; 2003.
[3] Navas-Loro M, Satoh K, Rodríguez-Doncel V. ContractFrames: bridging the gap between natural language and logics in contract law. In: JSAI International Symposium on Artificial Intelligence. Springer; 2018. p. 101-14.
[4] Satoh K, Asai K, Kogawa T, Kubota M, Nakamura M, Nishigai Y, et al. PROLEG: an implementation of the presupposed ultimate fact theory of Japanese civil code by PROLOG technology. In: JSAI international symposium on artificial intelligence. Springer; 2010. p. 153-64.
[5] Fuchs NE, Kaljurand K, Kuhn T. Attempto controlled english for knowledge representation. In: Reasoning web. Springer; 2008. p. 104-24.
[6] Wyner A. From the language of legislation to executable logic programs. In: Logic in the theory and practice of lawmaking. Springer; 2015. p. 409-34.
[7] Kowalski R. Logical english. Proceedings of Logic and Practice of Programming (LPOP). 2020.
[8] Kowalski B, Dávila J, CA CL, Calejo M. Logical English as a Programming Language for the Law. In: Programming Languages and the Law 2022; 2022. .
[9] Neff MS, McCord MC. Acquiring lexical data from machine-readable dictionary resources for machine translation. Citeseer; 1990.
[10] Och FJ, Ney H. The alignment template approach to statistical machine translation. Computational linguistics. 2004;30(4):417-49.
[11] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.
[12] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:191013461. 2019.
[13] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21(140):1-67.
[14] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Advances in neural information processing systems. 2020;33:1877-901.
[15] Ahkouk K, Machkour M, Majhadi K, Mama R. SQLSketch: Generating SQL Queries using a sketch-based approach. Journal of Intelligent & Fuzzy Systems. 2021;40.
[16] Witt A, Huggins A, Governatori G, Buckley J. Converting copyright legislation into machine-executable code: interpretation, coding validation and legal alignment. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law; 2021. p. 139-48.
[17] Papineni K, Roukos S, Ward T, Zhu WJ. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics; 2002. p. 311-8.
[18] Lin CY. Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out; 2004. p. 74-81.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220454

Why Do Tenants Sue Their Landlords? Answers from a Topic Model

Olivier SALAÜN [a,1], Fabrizio GOTTI [a], Philippe LANGLAIS [a] and Karim BENYEKHLEF [b]
[a] *RALI, DIRO, Université de Montréal*
[b] *Cyberjustice Laboratory, Faculty of Law, Université de Montréal*

Abstract. Topic modeling is widely used in various domains for extracting latent topics underlying large corpora, including judicial texts. In the latter, topics tend to be made by and for domain experts, but remain unintelligible for laymen. In the framework of housing law court decisions in French which mixes abstract legal terminology with real-life situations described in common language, similarly to [1], we aim at identifying different situations that can cause a tenant to prosecute their landlord in court with the application of topic models. Upon quantitative evaluation, LDA and BERTopic deliver the best results, but a closer manual analysis reveals that the second embedding-based approach is much better at producing and even uncovering topics that describe a tenant's real-life issues and situations.

Keywords. topic modeling, court decisions, French language, housing law, knowledge extraction

1. Context

Topic modeling is an application of natural language processing (NLP) widely used for summarizing large corpora into different clusters of terms. These terms describe latent topics present in the text but not immediately visible to the reader. In this work, we aim at applying and evaluating topic models to a corpus of court decisions in French from the Tribunal administratif du logement (TAL, Housing law tribunal) in Canada. This court deals exclusively with all disputes involving landlords and tenants bound by an accommodation lease contract. These litigations are mostly motivated by rent arrears or substandard housing. An analysis was performed by [1] over cases in which a tenant was claiming damages from their landlord before the TAL. The objective was to find which were the concrete factors (e.g. water/electricity access issue, bedbugs, lack of maintenance) that caused the judge to accept the tenant's claims and award them damages. The facts and evidence of tenant claims are judged in the light of articles 1854, 1864 and 1910 from Civil Code of Québec[2] that define landlord contractual obligations to provide an accommodation *in a good state of repair* and to ensure both *peaceable enjoyment* and *good state of habitability*. These articles provide general abstract legal concepts without making an exhaustive list of concrete criteria. This is left to case-by-case interpretation

[1] Corresponding Author: Olivier Salaün, salaunol@iro.umontreal.ca
[2] Articles 1854, 1864 and 1910 can be read at https://canlii.ca/t/55g4j

by the judge, hence [1]'s initiative of manually annotating 149 cases which ultimately yielded 44 *factors* (some are shown in the top left blue box of Figure 2). We shall call these factors **reference topics** (RTs) here. [1]'s manual topic extraction is costly (a legal expertise is required, on long documents) and only covers a tiny portion of the several hundred thousand cases. In this work, we intend to apply topic models as an attempt to automate such examination of cases and to design methods able to isolate relevant topics with respect to the housing law domain.

2. Related Work

Topic modeling is used in a wide variety of domains, such as social networks analysis [2], scientific papers [3] or medical data [4] for instance. Some may even call it *distant reading* [5] as it consists in applying approaches, such as LDA (explained later), to extract thematic representations of large corpora.

2.1. Topic Modeling for the Legal Domain

For legal practitioners, topic modeling also provides useful unsupervised methods for soft clustering/categorization of legal documents, without having a prior classification scheme [6]. Such methods extract **topics** that can be described as collections of words clustered together based on their distribution across the documents. In the legal field, topic models were applied for instance to UK legislative documents [7], Latvian legal acts [8] and court decisions from Australia [9], the Netherlands [10], Brazil [11,12] and the United States [13]. We must emphasize that these topic modeling experiments usually yielded topics meant for experts in the field. However, a certain language gap exists between the specialized legal terminology used by judges, and laymen's generic language as shown by [14]: They describe the same reality in different terms. This discrepancy exists for housing law, and we posit that extracted topics can help bridge this gap by illuminating a *taxonomy of practice* [9] i.e. real-world situations cues discussed in the scope of legal abstract concepts. Such a resource would be useful for laymen seeking legal information, for instance when trying to connect concrete problems (e.g. *water leakage*) to relevant legal concepts (e.g. *good state of habitability*).

2.2. Challenges about Topic Evaluation

Despite readily available toolkits facilitating topic modeling, evaluating topics is still an open question, in part because the assessment differs depending on the data and the legal area. A possible strategy is an extrinsic evaluation of the topics, for instance measuring the improvement topics bring when carrying out text classification [11,12]. Such a protocol obviously requires the cases to be manually classified beforehand (this is not our case). Intrinsic evaluation of topics can also be manually assessed by comparing topics automatically assigned to a document with its original ground truth category [13]. When no such categories are available, typical manual methods include ordinal three-point Likert-scales and intrusion tests [15,10]. Finally, a common automated metric for topic evaluation is topic coherence [16,8] based on word co-occurrences from an external reference corpus.

3. Data Description and Preprocessing

The dataset manually examined by [1] consisted only of 149 cases from 2017. We extended this to 12,102 cases spanning 2001 to 2018 that included clearly separated sections described by [17]: facts (rich with real-life situations described by laymen) and legal analysis (rich with legal terminology). Clear section boundaries are necessary because, unlike works from Section 2 and following [10], we do not carry out topic modeling on the entire document but only on the facts section so that our topics contain as little legal terminology as possible and more real-life situations. As in [1]'s work, we retained cases citing articles 1854, 1864 or 1910 from the Civil Code of Québec in which the tenant is the applicant, thus amounting to 1,381 cases. Since each case can contain several litigation factors, topic modeling is done at the paragraph level. Our resulting dataset of 34,685 paragraphs is processed in a standardized fashion with a SpaCy tokenizer: We remove dates, monies, digits, symbols, French- and law-specific stopwords, then filter tokens with specific part-of-speech tags, lemmatize and lowercase them before merging bigram collocations (e.g. *hot_water*). We remove paragraphs with less than 5 terms, yielding 26,815 paragraphs.

4. Models

We selected two traditional and one neural topic models, respectively: Latent Semantic Indexing (LSI, also known as Latent Semantic Analysis) [18], Latent Dirichlet Allocation (LDA) [19] and BERTopic [20]. When conducting training for each of these models, the number of topics (a hyperparameter) is set beforehand to 50, 100, and 200.

4.1. Latent Semantic Indexing (LSI)

LSI relies on singular value decomposition (SVD) of a sparse $paragraphs \times words$ ($P \times W$) matrix M in which each value $v_{p,w}$ represents the term frequency-inverse paragraph frequency (TF-I*D*F) weight of word w for paragraph p.$v_{p,w}$ increases with the frequency of w in p but decreases if w is widespread among paragraphs. The SVD of M produces 3 matrices M', M'' and M''' of respective shapes : $paragraphs \times topics$ ($P \times T$), $T \times T$ and $T \times W$. The last matrix provides word distribution to each topic. We use Gensim library [21] for training LSI models and set the number of power iteration steps to 100 for improving the accuracy of the SVD approximation with large sparse matrices.

4.2. Latent Dirichlet Allocation (LDA)

LDA is a generative stochastic model that aims at recreating the original corpus through a pseudo-corpus generation. For a preset number of topics, it generates a collection of pseudo-paragraphs whose word and topic distributions approximate as closely as possible those of the real dataset. Paragraphs are considered as random mixtures over latent topics, and topics as distributions over words. $Dir(\alpha)$ is the Dirichlet distribution of topics over paragraphs while $Dir(\beta)$ is the Dirichlet one of words over topics. $Dir(\alpha)$ is the prior for multinomial topic distribution θ_p for paragraph p while $Dir(\beta)$ is the prior for multinomial word distribution ϕ_k for topic k. As shown on Figure 1, at position i of pseudo-paragraph p of pseudo-corpus C, word $w_{p,i}$ is defined by both θ_p that defines the

Table 1. Top 10 terms from a randomly chosen topic for each model translated from French (predefined number of topics: 100)

LSI : door window room last place repair landlord day problem floor
LDA : party owner infiltration estimate dwelling concern heat finish list receive_notification
BERTopic : building manager management company caretaker witness son occupies responsible viner

topic $z_{p,i}$ at i in p, and by ϕ_k. After random initialization of these distributions and several passes over the documents, LDA is able to identify a steady collection of salient words for each topic k. We again use the Gensim library and set the number of passes at 100.

4.3. BERTopic

BERTopic relies on context-based representation derived from transformer [22,23] embeddings, thus allowing the model to access semantic information. First, the paragraphs are encoded with sentence-BERT embeddings [24] that are suitable for paraphrase detection and clustering. In our case, we chose a multilingual (over 50 languages) model[3] [25] for embedding French paragraphs. Next, these representations are dimensionally reduced with UMAP [26] before being passed to HDBSCAN [27] for soft and hierarchical clustering. A cluster contains paragraphs that are assumed to relate to the same topic. Said topic is then represented by a collection of the most salient words contained in its paragraphs through TF-IDF measures. After observing topics such as those in Table 1, we decided to retain only the top 5 words for each candidate topic during evaluation as the remaining terms are less topic-representative and noisier.

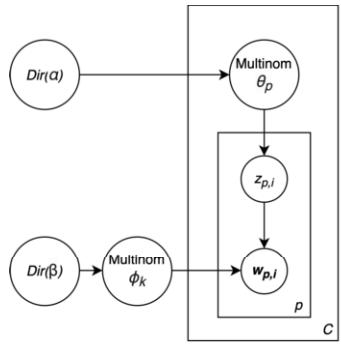


Figure 1. Graphical model representation of LDA generative process of word $w_{p,i}$ at position i of pseudo-paragraph p in pseudo-corpus C.

5. Quantitative Automated Evaluation

As discussed in Section 2.2, topic evaluation is challenging. In our case, despite the lack of text classification labels that would allow an extrinsic evaluation, we focus on two possible automatic approaches. The first one consists in comparing the **candidate topics** (CT) with the 44 **reference topics** (RT) manually identified by [1]. The second one relies on commonly used automatic topic coherence metrics.

5.1. Automated Evaluation with Respect to Domain-Specific Reference Topics

Comparing candidate topics (CTs) with [1]'s reference topics (RTs) addresses the question of whether a topic model can identify these RTs. An automated pairwise comparison

[3]The pretrained sentence-transformer model we used is available at: https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

approach is illustrated in Figure 2. For each pair (CT_i, RT_j), a score S is computed yielding a $|CT| \times |RT|$ score matrix from which we retain the top $|CT|$ scores and the corresponding (CT_i, RT_j) pairs. From these pairs, we count the number d of **d**istinct matched RTs. The recall and precision are obtained by dividing d by $|RT|$ and $|CT|$, respectively. Although these metrics are not perfect, they are a necessary compromise, since the RTs identified by [1] only cover a tiny portion of all decisions.

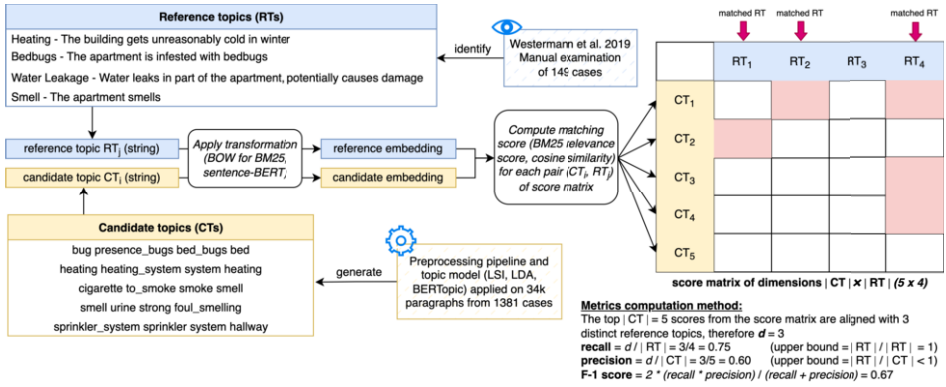


Figure 2. A toy example where 5 CTs are compared with 4 RTs. All topics are taken from actual [1]'s reference topics and candidate topics translated from French for illustration purposes.

Computing the similarity score S is delicate because RTs are short sentence-like descriptions while a CT is a sequence containing the 5 terms most representative of the topic. Two approaches are used for computing similarity: **1)** For each pair (CT_i, RT_j), each string is encoded with a multilingual sentence-transformer embedder[4] and the cosine similarity between the two is used as a matching score. **2)** We use an Okapi BM25 [28] approach in which RT and CT are queries and documents, respectively. Scores correspond to BM25 bag-of-words-based proximity scores assigned to CT w.r.t. to RT.

5.2. Topic Coherence: Evaluation with Respect to an External Reference Corpus

Topic quality is commonly measured with topic **c**oherence metrics, in particular normalized pointwise information (NPMI) shown by [29] to be positively correlated with human judgment. A topic CT_t gets a high **c**_NPMI score, shown in Eq. 1, if its N top terms have high pairwise joint probabilities.

$$c\_NPMI(CT_t) = \sum_{j=2}^{N} \sum_{i=1}^{j-1} \frac{log \frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{-logP(w_i,w_j)} \tag{1}$$

Joint probabilities $P(w_i, w_j)$ are computed on the basis of a large corpus, usually Wikipedia [16], with a 10-word co-occurrence window. We extracted and preprocessed (see Section 3) a French Wikipedia snapshot dated 1 September 2022 (2.4 million articles) and a corpus of housing law decisions (531k cases from Tribunal administratif du logement shown in Table 2 as TAL) as generic and domain-specific reference corpora,

[4] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

Table 2. Scores per model and per number of topics in terms of CTs-RTs similarity and topic coherence NPMI. Highest and second highest values are in bold and underlined respectively for each metric.

Model and number of topics		BM25 Proximity score			SBERT Cosine Similarity			c_NPMI	
		R	P	F1	R	P	F1	Wiki	TAL
LSI	50	0.273	0.240	0.255	0.409	0.360	0.383	0.0090	0.0504
	100	0.409	0.180	0.250	0.591	0.260	0.361	-0.0008	0.0512
	200	0.545	0.120	0.197	0.659	0.145	0.238	-0.0058	0.0361
LDA	50	0.568	**0.500**	**0.532**	0.523	0.460	0.489	0.0215	0.0780
	100	0.614	0.270	0.375	0.705	0.310	0.431	0.0035	0.0607
	200	**0.636**	0.140	0.230	0.818	0.180	0.295	-0.0181	0.0412
BERTopic	50	0.545	0.480	0.511	0.614	**0.540**	**0.574**	**0.1462**	**0.3087**
	100	0.591	0.260	0.361	0.818	0.360	0.500	0.1266	0.2368
	200	0.614	0.135	0.221	**0.909**	0.200	0.328	0.0830	0.1852

respectively. The resulting c_NPMI scores range from -1 to 1. -1 implies the complete absence of co-occurrence of a topic pair of words within the reference corpus while 1 means complete co-occurrence.

5.3. Results

As shown in Table 2, LDA and BERTopic overall outperform LSI across all metrics. For a given model, raising the preset number of generated topics improves BM25 and SBERT recalls, which is expected. Overall, for a fixed number of topics, considering precisions and F1 measures, LDA outperforms BERTopic in terms of BM25 proximity scores while BERTopic achieves the highest performance with SBERT Cosine Similarities. This could be explained by the fact that LDA and BERTopic are respectively bag-of-words and embedding-based models. Concerning c_NPMI scores and regardless of the reference corpus, LSI and LDA get scores close to 0, the latter slightly outperforming the former, while BERTopic achieves the highest scores. This suggests that terms in LSI and LDA topics co-occur by chance under independent distributions while terms clustered by BERTopic co-occur beyond chance [30]. Given that both BM25 and SBERT recall scores for LDA and BERTopic increase with the number of topics, and since we want to assess to what extent topic models can help in identifying housing law use cases, we decided to manually evaluate the set of 200 topics generated by each of these two models.

6. Qualitative Human Evaluation

6.1. Intrinsic Evaluation of Candidate Topics Relevance

In order to assess the quality and intelligibility of CTs for laymen, two non-legal experts (co-authors of this work) were asked to evaluate the top 5 terms of a total of 400 topics from LDA and BERTopic models (each yielded 200 topics). These topics were shown in random order to evaluators who had no information on the models that produced them. For each topic, evaluators were asked whether they were able to identify an issue or a situation that would concern a tenant. When this was the case, the annotators were further asked to succinctly describe the theme they detected (e.g. *mold*, *disagreement on rent*).

Table 3. Selected candidate topics qualified as relevant by evaluators and that match reference topics from [1] (translated from French). CQ2s come from BERTopic except when in italics (generated by LDA).

Reference topic	Number of matching CQ2s	CQ1s	Examples of CQ2s
Water Leakage	9	10	water hot_water water_damage water_pressure occur water_infiltration infiltration occur roof roofing *water_infiltration garage occur complete use* *let water_damage believe place material*
Noise	8	10	noise jackhammer excessive_noise saw noise_arise music loud_music excessive_neighbourhood party play noise child child_run disturb play subject_soundproofing complaint_unbelievable unit_verify
Bedbugs	6	6	exterminator treatment extermination bedbug proceed_treatment insomnia stress sleep bug_bite phobia *bug mattress infestation deliver kitchen_floor*
Heating	5	8	heating temperature thermostat degree furnace cold temperature winter october heating dismantling_heater close_start outdated_problem heater heating_system
Exterior Issues	5	9	balcony rear_balcony antenna rot banister staircase staircase_lead hand_rail stair_step solidly access_terrace access rear give_access lock level_elevator refreshment_put sprinkler_system corridor_need

Relevant CTs are reported in Tables 3 and 4. Overall, Cohen's kappa score[5] for inter-annotator agreement amounts to 0.562 for all topics, 0.386 for the 200 LDA topics and 0.649 for the 200 BERTtopic ones. The main difference between evaluators resides in the fact that one only considered material issues as relevant topics but dismissed several issues that could be induced by people (e.g. harassment, violence, intrusion), though these actually account for housing law litigation factors. If we put aside the 14 CTs related to interpersonal issues, the aforementioned kappa scores increase to respectively 0.619, 0.440 and 0.706. Such a difference in inter-annotator agreement between LDA and BERTopic can be explained by the fact that LDA bag-of-words topics were harder to interpret as they gather terms that make less sense together in comparison to BERTopic. Such a low score for LDA is consistent with its c_NPMI scores from Table 2 and with the fact that CTs defined by both annotators as relevant topics amount to 10.5% and 33.0% for LDA and BERTopic, respectively.

6.2. Qualitative Analysis of Candidate Topics Evaluated as Relevant

After the identification of actually relevant CTs by non-experts, a domain expert (co-author of this work) manually paired these CTs to [1]'s RTs. For convenience, CTs **q**ualified as relevant by at least one and by exactly both evaluators are named C**Q**1s and C**Q**2s, respectively. The number of distinct RTs that could be paired to CQ1s and CQ2s amount respectively to 28 and 22 out of 44. The top 5 RTs with the more matching CQ1s

[5] According to [31], Cohen suggested kappa scores to be interpreted as fair, moderate and substantial agreement for values in the respective ranges $0.21 - 0.40$, $0.41 - 0.60$, and $0.61 - 0.80$

Table 4. Examples of candidate topics qualified as relevant by non-expert evaluators that correspond to topics not included in [1]'s reference (translated from French). CQ2s come from BERTopic except when in italics (generated by LDA).

Uncovered topic	Number of matching CQ2s	Number of matching CQ1s	Examples of CQ2s
Plumbing	5	7	*affect plumbing hot lacking finishing* plumber plumbing drain valve batur kitchen repair_faucet washbasin_room water meal faucet noise water trickle_water adjust_definitely
Air quality	4	4	asthma symptom suffer doctor nose ventilation ventilation_system air exhaust duct allergy allergic mélabo test respiratory_problem
Internet access	3	3	telephone phone_number internet_service call telephone_line cable_origin optic_upgrade get_fibre bell_videotron hole_made videotron cable panel technician cable_television
Lighting	1	1	light lighting break_height burnt_pole lighting_deficient
Disabled accessibility	1	1	person_with_disability redo_june intercom_ramp hall_entrance autumn

and CQ2s are shown in Table 3. When pairing RTs and CQ2s, we noticed that RTs could be abstract and vague while CQ2s helped in bringing more nuance and precision by pinpointing precise themes. For instance, topic modeling allows extracting different noise-related issues such as construction (*jackhammer*), *loud_music*, *child*[ren]*_run*[ning] and *soundproofing*. The benefits of topic modeling are even more noticeable for *Exterior issues* by naming precise elements: *rear_balcony*, *staircase*, *hand_rail*, *access_terrace*, *level_elevator*. For relevant topics that could not be paired with RTs, the domain expert created new labels: 33 for CQ1s and 12 for CQ2s. This allowed uncovering new *litigation factors* that were not included in [1]'s RTs and that are shown in Table 4. For instance, several CQ2s relate to plumbing issues without involving water leakage. Other CQ2s, despite their small number, pinpoint sensitive issues such as air quality and accessibility for the disabled.

7. Discussion

Overall, relevant CTs are more likely to be obtained with BERTopic rather than with LDA. One explanation is that unlike most works described in Section 2 that dealt with documents from different legal areas [11,12,13], our corpus of paragraphs is much more homogeneous as it is only related to housing law, hence making topic modeling more difficult. Consequently, the bag-of-words approach of LDA gives less relevant topics compared to BERTopic, which has access to word semantic information. We also noticed that, when increasing the number of output topics, LDA was more likely to produce repeated noisy meaningless topics such as *berat blood applicances best pilule* (sic) reported by evaluators. A tentative explanation is that setting a very high number of topics can cause the LDA model to manufacture topics from noisy words. Such an issue was not observed with topics obtained from BERTopic.

Furthermore, the ratio of relevant CQ2s and CQ1s only covers a minority of all CTs, respectively 21.7% and 41.2%. CQ2s cover 10.5% and 33.0% of CTs by LDA and

BERTopic. These figures amount to 32.0% and 50.5% for CQ1s. A tentative explanation is that although input paragraphs are extracted from the facts section of court decisions, some legal jargon phrases still persist in them, yielding topics that do not refer to real-life situations but rather to formal legal procedures. The lack of domain knowledge and familiarity with housing law may also hinder evaluators from identifying relevant topics. Despite this issue, we must also emphasize that our topic modeling approach revealed new topics not included in [1]'s RTs. On the basis of CQ1s, 11% and 18% of LDA and BERTopic CTs referred to such uncovered situations.

8. Conclusion

In this paper, we applied topic modeling methods to a corpus of housing law decisions with the goal of automatically extracting topics similar to [1]'s factors. A quantitative analysis showed that LDA and BERTopic seemed to provide the best results, although a further manual analysis revealed that the latter method yielded more relevant topics thanks to its access to semantic information while the former was limited by its bag-of-words approach. As a guideline, we recommend using embedding-based rather than bag-of-words-based topic modeling approaches when dealing with a corpus focused on a single legal area. As future work, we intend to repeat the experiment with a larger corpus by adding claims from landlords, to include experts and non-experts for a broader manual evaluation of topics, and to improve the robustness of automatic metrics for filtering out relevant topics from noisy ones. So far, our results show that we are on a promising track to assist laymen in navigating through technical legal documents by connecting abstract legal concepts with concrete, real-life situations described in everyday language.

Acknowledgements We would like to thank the Cyberjustice Laboratory at the Université de Montréal, the LexUM Chair on Legal Information and the Autonomy through Cyberjustice Technologies project for supporting this research.

References

[1] Westermann H, Walker VR, Ashley KD, Benyekhlef K. Using Factors to Predict and Analyze Landlord-Tenant Decisions to Increase Access to Justice. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law; 2019. p. 133-42.
[2] Tan S, Li Y, Sun H, Guan Z, Yan X, Bu J, et al. Interpreting the public sentiment variations on twitter. IEEE transactions on knowledge and data engineering. 2013;26(5):1158-70.
[3] Griffiths TL, Steyvers M. Finding scientific topics. Proceedings of the National Academy of Sciences. 2004;101(suppl_1):5228-35.
[4] Song CW, Jung H, Chung K. Development of a medical big-data mining process using topic modeling. Cluster Computing. 2019;22(1):1949-58.
[5] Moretti F. Distant reading. Verso Books; 2013.
[6] Dyevre A. Text-mining for lawyers: how machine learning techniques can advance our understanding of legal discourse. Erasmus L Rev. 2021;14:7.
[7] O'Neill J, Robin C, O'Brien L, Buitelaar P. An Analysis of Topic Modelling for Legislative Texts. In: ASAIL@ICAIL; 2017. .
[8] Vīksna R, Kirikova M, Kiopa D. Exploring the Use of Topic Analysis in Latvian Legal Documents. In: Tagarelli A, Zumpano E, Latific AK, Calì A, editors. Proceedings of the First International Workshop

"CAiSE for Legal Documents" (COUrT 2020) co-located with the 32nd International Conference on Advanced Information Systems Engineering (CAiSE 2020), Grenoble, France, June 9, 2020. vol. 2690 of CEUR Workshop Proceedings. CEUR-WS.org; 2020. p. 39-47.

[9] Carter DJ, Brown J, Rahmani A. Reading the High Court at a distance: topic modelling the legal subject matter and judicial activity of the High Court of Australia, 1903-2015. University of New South Wales Law Journal. 2016;39(4):1300-54.

[10] Remmits Y. Finding the topics of case law: Latent dirichlet allocation on supreme court decisions. 2017.

[11] Luz De Araujo PH, De Campos T. Topic Modelling Brazilian Supreme Court Lawsuits. In: Legal Knowledge and Information Systems. IOS Press; 2020. p. 113-22.

[12] Aguiar A, Silveira R, Furtado V, Pinheiro V, Neto JAM. Using Topic Modeling in Classification of Brazilian Lawsuits. In: International Conference on Computational Processing of the Portuguese Language. Springer; 2022. p. 233-42.

[13] Silveira R, Fernandes C, Neto JAM, Furtado V, Pimentel Filho JE. Topic Modelling of Legal Documents via LEGAL-BERT. Proceedings http://ceur-ws org ISSN. 2021;1613:0073.

[14] Branting K, Balhana C, Pfeifer C, Aberdeen JS, Brown B. Judges Are from Mars, Pro Se Litigants Are from Venus: Predicting Decisions from Lay Text. In: JURIX; 2020. p. 215-8.

[15] Chang J, Gerrish S, Wang C, Boyd-Graber J, Blei D. Reading tea leaves: How humans interpret topic models. Advances in neural information processing systems. 2009;22.

[16] Röder M, Both A, Hinneburg A. Exploring the space of topic coherence measures. In: Proceedings of the eighth ACM international conference on Web search and data mining; 2015. p. 399-408.

[17] Lou A, Salaün O, Westermann H, Kosseim L. Extracting Facts from Case Rulings Through Paragraph Segmentation of Judicial Decisions. In: International Conference on Applications of Natural Language to Information Systems. Springer; 2021. p. 187-98.

[18] Dumais ST, Furnas GW, Landauer TK, Deerwester S, Harshman R. Using latent semantic analysis to improve access to textual information. In: Proceedings of the SIGCHI conference on Human factors in computing systems; 1988. p. 281-5.

[19] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Journal of machine Learning research. 2003;3(Jan):993-1022.

[20] Grootendorst M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv preprint arXiv:220305794. 2022.

[21] Řehůřek R, Sojka P. Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Valletta, Malta: ELRA; 2010. p. 45-50.

[22] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998-6008.

[23] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT. Minneapolis, Minnesota: Association for Computational Linguistics; 2019. p. 4171-86.

[24] Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019. p. 3982-92.

[25] Reimers N, Gurevych I. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics; 2020. p. 4512-25.

[26] McInnes L, Healy J, Saul N, Großberger L. UMAP: Uniform Manifold Approximation and Projection. Journal of Open Source Software. 2018;3(29):861.

[27] McInnes L, Healy J, Astels S. hdbscan: Hierarchical density based clustering. The Journal of Open Source Software. 2017;2(11):205.

[28] Trotman A, Puurula A, Burgess B. Improvements to BM25 and language models examined. In: Proceedings of the 2014 Australasian Document Computing Symposium; 2014. p. 58-65.

[29] Lau JH, Newman D, Baldwin T. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics; 2014. p. 530-9.

[30] Bouma G. Normalized (pointwise) mutual information in collocation extraction. Proceedings of GSCL. 2009;30:31-40.

[31] McHugh ML. Interrater reliability: the kappa statistic. Biochemia medica. 2012;22(3):276-82.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220455

Conditional Abstractive Summarization of Court Decisions for Laymen and Insights from Human Evaluation

Olivier SALAÜN [a,1], Aurore TROUSSEL [b], Sylvain LONGHAIS [b], Hannes WESTERMANN [b], Philippe LANGLAIS [a] and Karim BENYEKHLEF [b]
[a] *RALI, DIRO, Université de Montréal*
[b] *Cyberjustice Laboratory, Faculty of Law, Université de Montréal*

Abstract. Legal text summarization is generally formalized as an extractive text summarization task applied to court decisions from which the most relevant sentences are identified and returned as a gist meant to be read by legal experts. However, such summaries are not suitable for laymen seeking intelligible legal information. In the scope of the JusticeBot, a question-answering system in French that provides information about housing law, we intend to generate summaries of court decisions that are, on the one hand, conditioned by a question-answer-decision triplet, and on the other hand, intelligible for ordinary citizens not familiar with legal documents. So far, our best model, a further pre-trained BARThez, achieves an average ROUGE-1 score of 37.7 and a deepened manual evaluation of summaries reveals that there is still room for improvement.

Keywords. text summarization, court decisions, French texts, housing law, access to legal information

1. Introduction

In the province of Quebec in Canada, the Tribunal administratif du logement (TAL, Housing Law Tribunal) is a court with an exclusive jurisdiction within the framework of provincial housing law for all legal disputes involving a lease contract among landlords and tenants. Such litigations are generally motivated by late payment of the rent or substandard housing. Since the TAL has to deal with a massive number of cases every year (i.e. over 51.7k introduced cases and 55.8k audiences held in the year 2020-2021 [1]) and as the parties involved, especially tenants, are usually unfamiliar with or even intimidated by formal legal procedures [2], the Cyberjustice laboratory built a tool to facilitate access to legal information by landlords and tenants.

Such tool, whose preliminary foundations were laid by [3], was released in July 2021 as the JusticeBot[2], a decision-tree-like system in which laymen are guided through paths of questions. For each question, such as the one shown in Figure 1, users are given binary answers they have to choose from, so that they can be given more refined questions and

[1] Corresponding Author: Olivier Salaün, salaunol@iro.umontreal.ca
[2] JusticeBot, https://justicebot.ca/

more relevant information as they continue on a pathway that corresponds the most to their case. To ensure that each question and the implications of each optional answer are well understood, the user is given past decisions for illustration purposes. For instance, for the question in Figure 1, the concept of "being often late in paying the rent" has no straightforward definition. Therefore, the user is provided in the bottom light blue panel ten different decisions from the TAL, with one half in which the judge determined that the tenant was "often late" and another half in which it was not the case (shown in the bottom unwrapped panel).

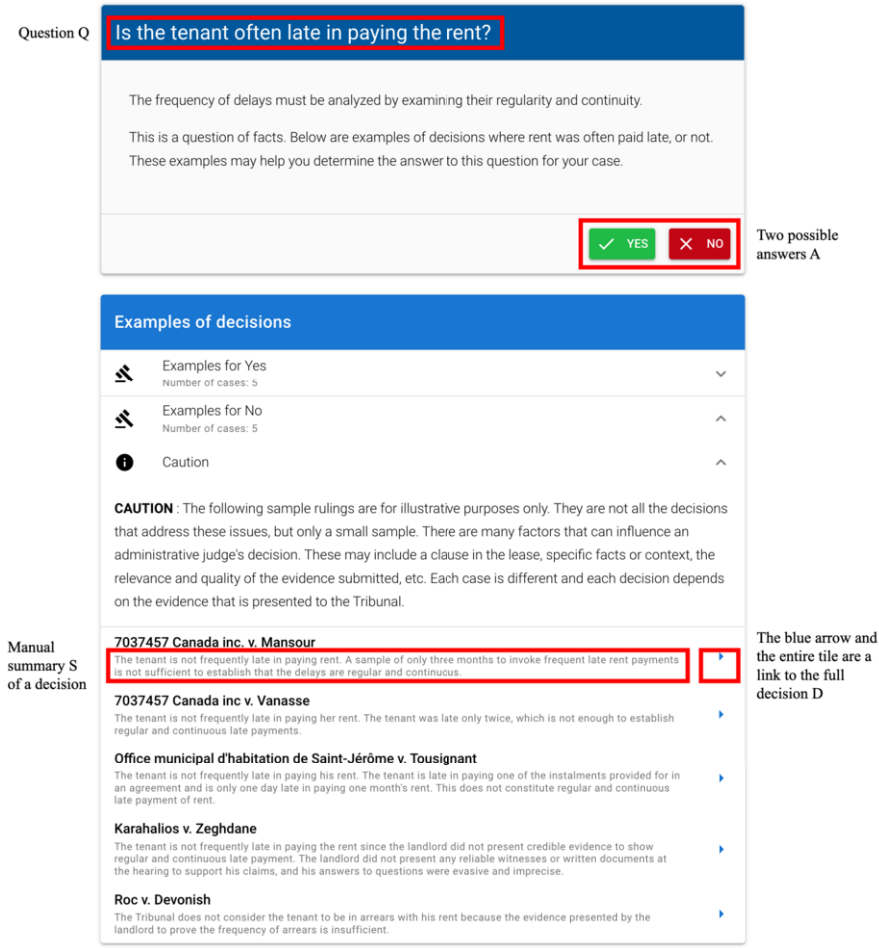


Figure 1. JusticeBot interface (translated from French) with a binary question: "Is the tenant often late in paying the rent?" The user can answer "Yes" or "No" as well as consult past court decisions that correspond to each answer as shown in the unwrapped bottom panel.

Although users have the possibility to read the full original case on CanLII[3] by clicking on it, we observed that 80.5% of them limit themselves to reading the short grey

[3]Canadian Legal Information Institute, https://www.canlii.org/

summary below the bold decisions titles. For each question-answer pair, relevant cases were manually collected and summarized by law graduate students following drafting instructions. Our goal is to investigate to what extent we can automate the summarization of court decisions for laymen within the scope of Quebec housing law.

2. Related Work

Automatic summarization applied to legal texts is generally framed as an extractive task. One of the earliest works was made by [4,5] who relied on rule-based thematic segmentation for selecting the most salient sentences for each section of Canadian judgments. A similar approach, based on entities extracted from text, was used by [6] for Australian court cases. In most experiments, in order to find the sentences that best summarize the entire judgment, authors generally use a scoring pipeline for deciding which sentences to add to the output summary. Such pipelines rely not only on the content of the sentence itself but also on its rhetorical role [7] (i.e. whether the sentence belongs to the Fact, the Issue or the Conclusion section of the case) whose importance was highlighted by [8], [9] and [10] for British, Taiwanese and Indian courts cases, respectively.

Besides these works based on sentence-extraction, more and more sophisticated models were developed for abstractive summarization by [11,12,13], but such approaches were mostly applied to news datasets. To the best of our knowledge, the only benchmark available for abstractive summarization of legal documents is BigPatent [14], though this corpus consists of patents and not court judgments as in the aforementioned extractive summarization tasks. Several reasons may explain the lack of accessible benchmarks for legal judgment abstractive summarization:

- Summaries written by legal experts are prohibitively costly to obtain, due to the length and complexity of decisions, plus the scarcity of legal practitioners who must be familiar with the target legal area ;
- Since decisions are significantly longer than documents in generic NLP corpora, they can be hard to process for models whose maximum input length is limited despite the emergence of transformer-based [15] models such as [16,17] that can process longer documents but at a high computing cost. Therefore, an extractive approach is more widespread for this type of document.

Unlike the aforementioned tasks, our summarization goal is slightly different as we aim at generating summaries intended for laymen with no prior legal knowledge instead of legal practitioners. Hence, an abstractive approach is preferred over an extractive one. Moreover, our task implies that the model makes a summary conditioned by a predefined triplet of question-answer-decision such as those shown on the interface. Finally, we must emphasize that our corpus is in Canadian French, a language not as widespread as English in the legal NLP field.

Moreover, our work does not fit the usual scope of legal summarization for domain experts as it is closer to that of other NLP experiments trying to make justice more accessible to laymen. For instance, [18] made a retrieval task in which the most relevant articles of Belgian law must be retrieved given a question asked by ordinary citizens. Similarly, [19] pursued the same goal with *plumitifs*, court dockets with complex abbreviations and legal jargon, that they tried to convert into intelligible texts for laymen through a text generation approach.

3. Data and Task Description

We extracted from the JusticeBot database and from CanLII a total of 156 instances. Although the dataset is small, it reflects diverse real-life issues faced by users. Each instance has 4 pieces of text:

- the question `Q` and the answer `A` from the JusticeBot interface ;
- the main text of a decision `D` that we extracted with heuristics from the HTML page from CanLII (metadata are removed) ;
- the summary `S` about `D` provided by an annotator to the JusticeBot user.

The task consists in mapping `Q`, `A` and `D` to the target summary `S`. The mean/median number of tokens for `Q`, `A`, `D` and `S` amount to 13/13, 1/1, 1030/745 and 44/41, respectively. Because of the dataset size, we will run our experiments with 10 folds, each fold having a train/validation/test split ratio of approximately 80:10:10. Performance results are averaged scores across folds and standard deviations are reported.

4. Models and Summary Generation Experiments

We decided to use the transformer-based encoder-decoder model BARThez [20] (same architecture as BART [13]) as it delivered state-of-the-art performance in French for news and dialogue summarization [21]. It is also a suitable starting point for making a legal-French-oriented language model [22].

4.1. Further Pretraining through Unsupervised Denoising Task

We used in our experiments two versions of BARThez: one with default pretrained parameters[4] called **Van**BART (**Van**illa BARThez), another called **FPT**BART in which default parameters are **F**urther **P**re**T**rained through the unsupervised denoising task with the FairSeq library [23]. In such a task, as described by [13], the model is given as input a corrupted version of a text segment from which it must generate the original one. The further pretraining corpus is made of 531,564 TAL decisions which we split into train and validation sets with an 80:20 ratio. Two resources frequently cited by TAL magistrates, 3.5k articles from Civil Code of Québec[5] (C.c.Q.) and TAL law[6], were also added to the train set. The denoising task is performed during 2 million steps with a 10^{-5} learning rate and 10^{-2} weight decay with Adam [24] optimization of cross-entropy loss. After roughly 12 days of pretraining on a single NVIDIA GeForce RTX 3090, the perplexity decreased from 1.78 to 1.33 on the validation set.

4.2. Supervised Text Summarization and Combinations of Text Inputs

Given text inputs `Q`, `A` and `D`, and target summary `S`, we tried several combinations of text inputs such as:

1. **D**: all paragraphs of all sections of **d**ecision `D` ;

[4]Pretrained checkpoint available at https://huggingface.co/moussaKam/barthez
[5]Code civil du Québec, RLRQ c CCQ-1991, http://canlii.ca/t/6b4rq
[6]Loi sur la régie du logement, RLRQ c R-8.1, http://canlii.ca/t/69m68

2. **QD**: a concatenation of **q**uestion `Q` and **d**ecision `D`'s paragraphs ;
3. **QDr**: same as above, but the paragraphs of decision `D` are in **r**everse order ;
4. **QAD**: a concatenation of **q**uestion `Q`, **a**nswer `A` and **d**ecision `D` ;
5. **QADr**: same as above, but the paragraphs of decision `D` are in **r**everse order.

Several reasons explain the reverting of paragraphs order in inputs 3 and 5:

- the annotators who drafted the summaries emphasized that the most relevant information was usually located towards the end of the decision, just before the verdict section. Therefore, they would tend to spend more time reading the pre-verdict part of the document as it gives the gist of the case instead of reading it from top to bottom. Such observations are consistent with those drawn by domain experts in the summarization task conducted by [6] ;
- although BART architecture has a larger maximum input sequence length (1024 tokens) with respect to commonly used transformer models (512 tokens for BERT [25]), reverting the paragraphs order of a court case allows minimizing the risk that important information located towards the end of the decision is not included in the input sequence of the model.

Table 1. Average scores in terms of automatic text generation metrics for each model and combination of text inputs (standard deviations are shown between parentheses and best scores are in bold font).

	Model and input combination	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
VanBART	decision	7.4 (6.2)	31.0 (7.4)	14.4 (7.5)	25.0 (7.4)
	question + decision	8.2 (3.7)	34.5 (4.3)	16.5 (4.6)	27.5 (5.3)
	question + decision (reversed order)	9.9 (5.7)	35.1 (5.1)	18.4 (5.5)	28.6 (5.5)
	question + answer + decision	8.4 (2.8)	33.9 (3.5)	17.0 (3.5)	27.5 (3.5)
	question + answer + decision (reversed order)	9.1 (4.0)	34.3 (3.2)	17.0 (3.2)	27.8 (3.0)
FPTBART	decision	10.0 (5.2)	32.8 (5.9)	16.2 (6.1)	26.0 (5.9)
	question + decision	13.0 (5.8)	**37.7 (6.1)**	20.1 (6.1)	29.9 (6.5)
	question + decision (reversed order)	12.0 (4.9)	37.3 (4.5)	19.8 (5.8)	29.6 (4.8)
	question + answer + decision	**14.0 (6.7)**	37.1 (6.7)	20.5 (7.5)	29.9 (6.7)
	question + answer + decision (reversed order)	13.3 (6.4)	**37.7 (6.9)**	**20.8 (7.4)**	**30.1 (7.0)**

Each aforementioned combination of text is provided as input to both VanBART and FPTBART. For each fold, the model is fine-tuned with the Adam optimizer. Given the small dataset size, the batch size is 1. In order to smooth out the optimization of cross-entropy loss, we apply an initial learning rate of 10^{-4} with a scheduler that halves it at the end of each training epoch if the ROUGE-1 score does not improve on the validation set. The training is stopped if this score does not improve after 10 consecutive epochs. The model whose parameter setting achieves the highest ROUGE-1 score on the validation set is used for summary generation, in which the maximum number of output tokens and the number of beams for beam search are set to 200 and 3, respectively.

5. Results and Discussion

For each model and each combination of text inputs, Table 1 gives the average scores across 10 folds for BLEU [26], ROUGE-1, ROUGE-2, ROUGE-L [27] along with stan-

dard deviation. Overall, for a given input combination, FPTBART delivers a higher performance with respect to VanBART. Such improvement is consistent with the fact that unsupervised pretraining of a transformer model helps for tasks in specialized domain as shown by [28]. Considering FPTBART results, the combination of question, answer and decision (QAD and QADr) seems to perform best in terms of ROUGE and BLEU as they contain more information and slightly outperform the combination of question and decision only (QD). Still, as it is hard to appreciate such syntax-based measures given the dataset size and the nature of our experiment, we retained output summaries from settings QD, QAD and QADr for manual evaluation.

5.1. Manual Evaluation

For a given fold, we took 16 test instances for which we considered a total of 48 output summaries generated by FPTBART with inputs QD, QAD and QADr. Three experts (co-authors of this paper), including one NLP specialist and two law graduate students, evaluated these summaries with an online form[7]. On the basis of guidelines provided by [29,30], we designed an intrinsic evaluation framework with two parts. The first one is related to the form (fluency) of candidate summaries:

- **1.0 grammar**: does the candidate summary contain any grammar or spelling mistakes?
- **1.1 readability**: does the summary contain repeated words ("hallucinations")? Does it make intelligible sense?
- **1.2 style**: is the choice of words appropriate for a JusticeBot layman user?

The latter part is related to the summary usefulness (adequacy) with respect to the JusticeBot's objective to ease access to legal information:

- **2.0 adequacy with respect to the decision**: does the candidate summary accurately reflect the use case and the relevant elements described in the decision?
- **2.1 adequacy with respect to the question**: does the summary address the issue described in the question shown to the JusticeBot user?
- **2.2 linking the decision and the question**: does the summary explain how the decision illustrates the answer suggested to the question? Is the summary meaning consistent with the answer?
- **2.3 consistency with manual summary**: is the generated summary consistent with the elements provided in the manual one already displayed in the JusticeBot?

All criteria are assessed by evaluators on a 4-point ordinal scale. The scores are shown in Table 2 along with Krippendorff's alphas (KA), a measure of inter-evaluator agreement[8]. Average instance-wise and question-wise KA across evaluators amounts to 0.525, but such a value hides disparities. As shown in Table 2, the KA for fluency questions are close to 0 and even negative, denoting a lack of agreement among evaluators despite efforts made to make each criterion as clear as possible. This could also be due to overly specific fluency questions. On the other hand, the agreements are more noticeable for adequacy questions, especially for questions 2.2 and 2.3. On the basis of unweighted

[7]The evaluation form is available at https://forms.gle/uX8n4LuQ5sddxsfD8

[8]Krippendorff's alpha ranges from −1 to 1. 1 denotes perfect agreement, 0 denotes absence of agreement beyond chance, and negative alpha indicates disagreement [31].

Table 2. Average of manual evaluations scores for each criterion (scale from 1 to 4 included) along with Krippendorff's alphas. For each criterion, the highest average of evaluators' scores is in bold font.

	Manual evaluation criterion	Krippendorff's alpha	Average of evaluators' scores		
			QD	QAD	QADr
Fluency	1.0 grammar	-0.093	3.73	3.69	**3.90**
	1.1 readbility	0.059	3.56	3.54	**3.62**
	1.2 style	-0.200	3.56	**3.62**	**3.62**
	Unweighted average of fluency scores		*3.62*	*3.62*	***3.72***
Adequacy	2.0 adequacy with respect to decision	0.490	2.90	3.21	**3.00**
	2.1 adequacy with respect to question	0.621	2.96	**3.17**	2.94
	2.2 linking decision and question	0.736	2.69	**2.83**	2.62
	2.3 consistency with manual summary	0.776	2.54	**2.67**	2.50
	Unweighted average of adequacy scores		*2.77*	***2.97***	*2.77*
	Unweighted average of all scores		*3.13*	***3.25***	*3.17*

average of all manual scores, the FPTBART achieves the best performance with QAD (3.25) input followed by QADr (3.17) and QD (3.13). The fact that QAD outperforms QADr for adequacy criteria but underperforms for fluency criteria suggests that reversing decision paragraphs order within the text input has little influence on output summaries.

5.2. Correlation among Automatic and Manual Metrics

Given the important manual evaluation cost, in particular in the specialized domain of housing law, we tried to find whether some automatic metrics can be used as proxies for the different manual evaluation criteria. We took the scores obtained in the previous subsection 5.1 for summaries generated by FPTBART with input combinations QD, QAD and QADr. For each pair of candidate and reference summaries, we compute automatic metrics available for evaluation of text generation: ROUGE-1, ROUGE-2, ROUGE-L, BERTscore [32], BLEU, chrF [33]. Once we have the manual criteria (MC) scores on one hand and the automatic metric (AM) ones on the other hand, we compute a correlation matrix (Kendall's τ coefficients) among these two sets of metrics that is shown as a heatmap in Figure 2. Overall, correlations between AM and fluency-related MC struggle to exceed 0.5 as shown in the region surrounded by green dashed lines. On the contrary in the region surrounded by solid blue edges, adequacy-related scores MC and AM have higher correlations. Computing the average correlation of each AM with these four MC suggests that ROUGE-1 (0.664) and ROUGE-2 (0.633) are the best proxies for adequacy-related metrics, despite them being merely syntax-based.

5.3. Qualitative Analysis

Upon manual examination, some summaries appeared easier to generate than others, especially those that consisted in a single sentence and/or are related to rent arrears (the most frequent issue in our dataset and in real life), although models also struggle with time duration. This is shown in Example (a) in Figure 3 where all candidate summaries accurately describe the tenant as being late in payment, with QADr adding 3 months of delay. For more complex cases, on the contrary, output summaries are less convincing as models tend to repeat phrases used in somewhat similar cases but without properly

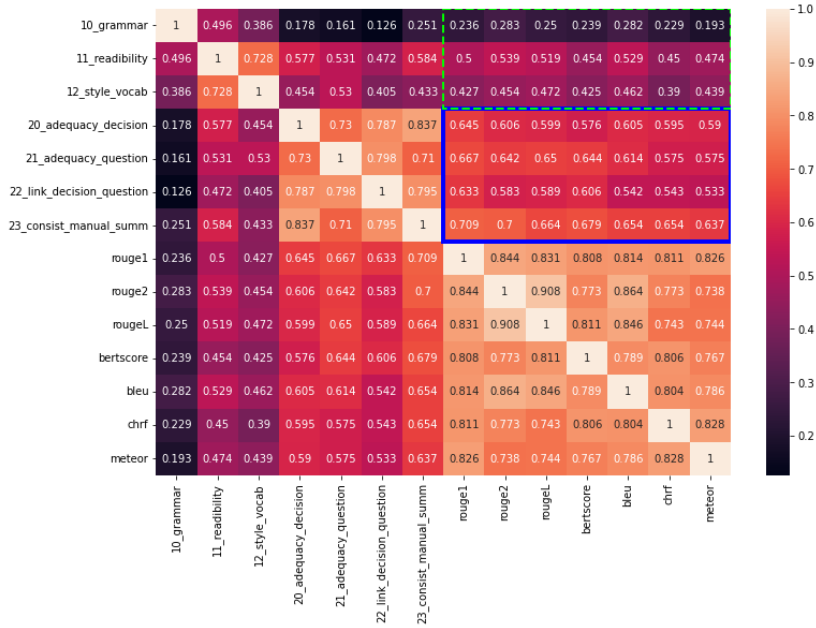


Figure 2. Heatmap of a correlation matrix among automatic and manual metric scores for summaries generated by FPTBART with input combinations QD, QAD and QADr. Green and blue rectangles highlight correlation values between automatic metrics with fluency criteria and adequacy criteria, respectively.

addressing the instance at hand. Example (b) provides a good illustration with the blue highlighted segments that contradicts the reference summary. Moreover, the last candidate (QADr), although being consistent with the input triplet and target summary, is making an extrapolation about article 1943 C.c.Q. that indeed makes the notice invalid but specifies nothing regarding what language should be used. Therefore, that language issue was handled by the court itself. As shown by these examples, automated summarization may generate summaries with correct wording at first sight. However, the possibility of them containing legal errors still makes a manual evaluation by experts necessary.

6. Conclusion

We implemented a novel conditional text generation task aimed at making summaries of court decisions for laymen by employing a state-of-the-art generative model. In addition to using automatic metrics (AM), we designed domain-specific manual criteria (MC) for sensible human evaluation and to assess to what extent generated summaries meet JusticeBot's user requirements. Overall, the best performance in terms of AM and MC was achieved by a further-pretrained BARThez that gets a question-answer-decision triplet as input, but a closer examination of output summaries reveals large room for improvement. As future work, we plan to: obtain a much larger dataset that covers more cases ; find methods to guide and control summary generation at a finer grain in order to incorporate legal reasoning ; examine how transformer attention weights handle each component of triplet input.

(a)	**Input question:** Is the tenant more than 3 weeks late in paying rent? **Input answer:** Yes **Reference summary:** The tenant is more than 3 weeks late in paying the rent as he has not paid his rent for 2 months. **QD, QAD:** The tenant is more than 3 weeks late in paying rent as he has not paid his rent for 2 consecutive months. **QADr:** The tenant is more than 3 weeks late with his rent as he has not paid his rent for 5 consecutive months.
(b)	**Input question**: Does the notice sent by your landlord comply with the law? **Input answer:** No **Reference summary:** The notice is not valid because it is written in French while the lease is in English and this error causes prejudice to the tenant. **QD:** The notice is wrong because it is not written in French while the lease is in English. However, despite this error, the tenant understood the content of the notice very well and was not prejudiced in any way by this situation. **QAD:** The notice is written in French whereas the lease is in English. However, despite this error, the tenant understood the content of the notice very well and was not prejudiced in any way. **QADr:** The notice of modification of the lease does not comply with the requirements of article 1943 C.c.Q. since it is written in French whereas the lease is in English.

Figure 3. Examples of summaries (translated from French) generated by FPTBART. QD, QAD and QADr correspond to the input used.

Acknowledgements We would like to thank the Cyberjustice Laboratory at the Université de Montréal, the LexUM Chair on Legal Information and the Autonomy through Cyberjustice Technologies project for supporting this research.

References

[1] Tribunal administratif du logement. Rapport annuel de gestion 2020-2021; 2021. Report retrieved on 11th of August 2022 from https://www.tal.gouv.qc.ca/sites/default/files/Rapport_annuel_2020-2021.pdf.

[2] Gallié M, Brunet J, Laniel RA. Les expulsions pour arriérés de loyer au Québec: un contentieux de masse. McGill Law Journal/Revue de droit de McGill. 2016;61(3):611-66.

[3] Westermann H, Walker VR, Ashley KD, Benyekhlef K. Using Factors to Predict and Analyze Landlord-Tenant Decisions to Increase Access to Justice. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law; 2019. p. 133-42.

[4] Farzindar A, Lapalme G. LetSum, an automatic Legal Text Summarizing. In: Legal knowledge and information systems: JURIX 2004, the seventeenth annual conference. vol. 120. IOS Press; 2004. p. 11.

[5] Farzindar A, Lapalme G. Legal Text Summarization by Exploration of the Thematic Structure and Argumentative Roles. In: Text Summarization Branches Out. Barcelona, Spain: Association for Computational Linguistics; 2004. p. 27-34.

[6] Polsley S, Jhunjhunwala P, Huang R. CaseSummarizer: A System for Automated Summarization of Legal Texts. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations. Osaka, Japan: The COLING 2016 Organizing Committee; 2016. p. 258-62.

[7] Saravanan M, Ravindran B. Identification of rhetorical roles for segmentation and summarization of a legal judgment. Artificial Intelligence and Law. 2010;18(1):45-76.

[8] Hachey B, Grover C. Extractive summarisation of legal texts. Artificial Intelligence and Law. 2006;14(4):305-45.

[9] Liu CL, Chen KC. Extracting the gist of Chinese judgments of the supreme court. In: proceedings of the seventeenth international conference on artificial intelligence and law; 2019. p. 73-82.

[10] Bhattacharya P, Poddar S, Rudra K, Ghosh K, Ghosh S. Incorporating Domain Knowledge for Extractive Summarization of Legal Case Documents. In: Proceedings of the 18th International Conference on Artificial Intelligence and Law (ICAIL); 2021. p. 22-31.

[11] Nallapati R, Zhou B, dos Santos C, Gulçehre Ç, Xiang B. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning; 2016. p. 280-90.

[12] See A, Liu PJ, Manning CD. Get To The Point: Summarization with Pointer-Generator Networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics; 2017. p. 1073-83.
[13] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020. p. 7871-80.
[14] Sharma E, Li C, Wang L. BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; 2019. p. 2204-13.
[15] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998-6008.
[16] Zhang J, Zhao Y, Saleh M, Liu P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning. PMLR; 2020. p. 11328-39.
[17] Zaheer M, Guruganesh G, Dubey KA, Ainslie J, Alberti C, Ontanon S, et al. Big bird: Transformers for longer sequences. Advances in Neural Information Processing Systems. 2020;33:17283-97.
[18] Louis A, Spanakis G. A Statutory Article Retrieval Dataset in French. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics; 2022. p. 6789-803.
[19] Beauchemin D, Garneau N, Gaumond E, Déziel PL, Khoury R, Lamontagne L. Generating Intelligible Plumitifs Descriptions: Use Case Application with Ethical Considerations. In: Proceedings of the 13th International Conference on Natural Language Generation. Dublin, Ireland: Association for Computational Linguistics; 2020. p. 15-21.
[20] Kamal Eddine M, Tixier A, Vazirgiannis M. BARThez: a Skilled Pretrained French Sequence-to-Sequence Model. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021. p. 9369-90.
[21] Zhou Y, Portet F, Ringeval F. Effectiveness of French Language Models on Abstractive Dialogue Summarization Task. In: LREC 2022; 2022. p. 3571-81.
[22] Garneau N, Gaumond E, Lamontagne L, Déziel PL. CriminelBART: a French Canadian legal language model specialized in criminal law. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law; 2021. p. 256-7.
[23] Ott M, Edunov S, Baevski A, Fan A, Gross S, Ng N, et al. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In: Proceedings of NAACL-HLT 2019: Demonstrations; 2019. p. 48-53.
[24] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. In: ICLR (Poster); 2015. .
[25] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT; 2019. p. 4171-86.
[26] Papineni K, Roukos S, Ward T, jing Zhu W. BLEU: a Method for Automatic Evaluation of Machine Translation; 2002. p. 311-8.
[27] Lin CY. ROUGE: A Package for Automatic Evaluation of Summaries. In: Text Summarization Branches Out. Barcelona, Spain: Association for Computational Linguistics; 2004. p. 74-81.
[28] Zheng L, Guha N, Anderson BR, Henderson P, Ho DE. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law; 2021. p. 159-68.
[29] Celikyilmaz A, Clark E, Gao J. Evaluation of text generation: A survey. arXiv preprint arXiv:200614799. 2020.
[30] Howcroft DM, Belz A, Clinciu MA, Gkatzia D, Hasan SA, Mahamood S, et al. Twenty Years of Confusion in Human Evaluation: NLG Needs Evaluation Sheets and Standardised Definitions. In: Proceedings of the 13th International Conference on Natural Language Generation; 2020. p. 169-82.
[31] Zapf A, Castell S, Morawietz L, Karch A. Measuring inter-rater reliability for nominal data–which coefficients and confidence intervals are appropriate? BMC medical research methodology. 2016;16(1):1-10.
[32] Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y. BERTScore: Evaluating Text Generation with BERT. In: International Conference on Learning Representations; 2020. .
[33] Popović M. chrF: character n-gram F-score for automatic MT evaluation. In: Proceedings of the Tenth Workshop on Statistical Machine Translation. Lisbon, Portugal: Association for Computational Linguistics; 2015. p. 392-5.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220456

Toward an Intelligent Tutoring System for Argument Mining in Legal Texts

Hannes WESTERMANN [a,1], Jaromír ŠAVELKA [b], Vern R. WALKER [c], Kevin D. ASHLEY [d] and Karim BENYEKHLEF [a]

[a] *Cyberjustice Laboratory, Faculté de droit, Université de Montréal*
[b] *School of Computer Science, Carnegie Mellon University*
[c] *LLT Lab, Maurice A. Deane School of Law, Hofstra University*
[d] *School of Computing and Information, University of Pittsburgh*

Abstract. We propose an adaptive environment (CABINET) to support caselaw analysis (identifying key argument elements) based on a novel cognitive computing framework that carefully matches various machine learning (ML) capabilities to the proficiency of a user. CABINET supports law students in their learning as well as professionals in their work. The results of our experiments focused on the feasibility of the proposed framework are promising. We show that the system is capable of identifying a potential error in the analysis with very low false positives rate (2.0-3.5%), as well as of predicting the key argument element type (e.g., an issue or a holding) with a reasonably high F_1-score (0.74).

Keywords. Intelligent tutoring system, caselaw analysis, case brief, legal education, legal annotation, legal text classification, argument mining, human-computer interaction.

1. Introduction

In this paper we examine the application of cognitive computing [16] to support both a law student learning how to extract key arguments from a court opinion and a legal expert performing the same. We propose an adaptive environment that evolves from a tutoring system to a production annotation tool, as a user transitions from a learner to an expert. The concept is based on a novel cognitive computing framework where (1) the involvement of machine learning (ML) based components is carefully matched to the proficiency level of a human user; and (2) the involvement respects the limitations of the state-of-the-art of automated argument mining in legal cases. We experimentally confirm feasibility of the key ML components by testing the following two hypotheses: Given a sentence in a case brief, it is possible (H1) to detect if the sentence is placed in an *incorrect* section, and (H2) to predict the *correct* section for the sentence.

[1] Corresponding Author: Hannes Westermann, E-mail: hannes.westermann@umontreal.ca

2. Background

Lawyers routinely analyze case decisions (i.e., court opinions) to gain insight into what is a persuasive or binding precedent (typically common law countries) and/or what is the established decision-making practice in a given matter (typically civil law countries). As the list of relevant cases may be long and the opinions might be sizeable, a principled approach to the analysis is necessary to make the task feasible and as efficient/effective as possible. Such an approach requires knowing how to read an opinion, which parts to focus on, and which information to identify as crucial for understanding the case.

In U.S. law schools, case briefs are widely employed to teach law students how to analyze a case and how to use prior decisions to create new arguments or analyses [6]. Writing a case brief involves reading and understanding a case, and identifying text passages that contain the key aspects of the decision. These are then extracted and arranged in a structured format that often includes the following sections:

- **Facts** - Events and actions relevant to the dispute.
- **Issue** - Main questions (points of contention) the court must address.
- **Holding** - Legal rulings when the law is applied to a particular set of facts.
- **Procedural History** - The treatment the dispute has received from the courts.
- **Reasoning** - The analysis of the court leading to the outcome.
- **Rule** - The official rules the court must adhere to (e.g., statutory provisions).

Interestingly, many professors never ask students to turn in their briefs and, hence, do not provide a learner with much needed feedback. [18] However, practice and feedback are essential for learning. When it comes to practice, the research clearly shows that it should be focused and deliberate [8], at the appropriate level of challenge [8], and in sufficient quantity [14]. Such practice should be coordinated with targeted feedback on specific aspects of students' performance in order to promote the greatest learning gains. [2,5] Feedback should also be timely, i.e., immediate and frequent [13]. These elements do not seem present when it comes to learning to brief cases. As a result, while law students tend to start out by dutifully briefing cases, they usually switch to a less detailed approach after a few weeks, focused on color-coding sentences or taking notes in the margins of the case texts. Due to the lack of feedback and practice, it is thus unclear whether the crucial skill of briefing cases has been acquired.

To address the issue we propose CABINET, an intelligent tutoring system that gradually evolves from a platform aimed at learners to a powerful annotation environment to support an expert. In a nutshell, CABINET allows a user to select a sentence and assign it to one of the case brief's sections. More importantly, the system provides varying levels of scaffolding (i.e., varying levels of challenge) and timely feedback appropriate for the learner's level of proficiency to maximize the learning outcomes. The tool thus adapts with the user, teaching them how to brief cases at first and later supporting them in briefing and understanding cases more efficiently.

3. Related Work

Numerous researchers have proposed frameworks where a human and a computer complement each other in performing tasks in the legal domain. For example, human-aided

computer cognition framework has been proposed and evaluated in the context of eDiscovery. [15] Active learning has been explored in various contexts, such as classification of German [35] or United States [25] statutory provisions, or relevance assessment in eDiscovery [7]. The annotation tool proposed in [37] supports human annotators by enabling them to view similar sentences together. The environment described in [36] provides statistical insights into a data set assisting a human expert in creating text classification rules. The work presented in this paper is to our best knowledge the first study that explicitly maps multiple ML components to different levels of user's proficiency.

Multiple research studies have explored applications of intelligent tutoring systems in teaching legal argumentation and case analysis skills. These include supporting law students in graphically representing legal arguments [22], assessing case relevance and distinguishing cases [1], performing case-based and rule-based reasoning [4], and selecting applicable legal rules from statutes [23] and precedents [21]. An adaptive legal textbook based on knowledge graphs has been proposed in [31]. The framework presented in this paper is the first intelligent tutoring system for legal domain that can be adapted to the proficiency level of a user to eventually support a legal professional in the task of analyzing cases by extracting their key arguments.

A key component of the proposed framework is the automatic recognition of key argument elements in case texts. This task has been studied extensively in AI & Law and it is often referred to as automatic identification of rhetorical roles that sentences play in the text of courts' opinions. Rhetorical role classification focuses on segmenting cases into functional parts [39,12,28] which can, e.g., improve legal information retrieval and enable legal argument retrieval [10,33,34]. Information about a sentence's rhetorical role can also be utilized in summarization [9,11,19,3,40]. The roles often include categories such as Facts, Issue, or Conclusion that are related to the ones used in this work.

A variety of ML/NLP techniques have been employed to predict sentence role labels. These span from rule-based approaches [36] to applying ML models such as Support Vector Machines [33]. The problem has also been treated as tagging of sequences that consist of multiple sentences instead of simpler single sentence classification. Here, models such as Conditional Random Fields have often been used [24,26]. A deep learning system based on Bi-LSTM was shown to perform well in [3]. Systems based on a multilingual embeddings, Bi-LSTM, or pre-trained language models demonstrated strong transfer learning capabilities in this task [29,30]. The work presented in this paper is the first attempt to use the sentence rhetorical role identification models in intelligent tutoring to support law students in learning how to analyze legal cases.

4. Proposed Framework

We propose CABINET (CAse Brief INteractive EnvironmenT) which is a cognitive computing framework that adapts to the proficiency level of a user. An overall design of CABINET's user interface is shown in Figure 1. We adopt a fairly standard layout where an analyzed document is displayed beside a template to be populated with the extracted key argument elements. The goal is to identify the key argument spans of text (*argument element identification task*) and to categorize them in terms of case brief sections (*argument element categorization task*).

CABINET provides scaffolds and supports that allow it to evolve from an intelligent tutoring system for learners to a tool to support the more efficient work for professionals.

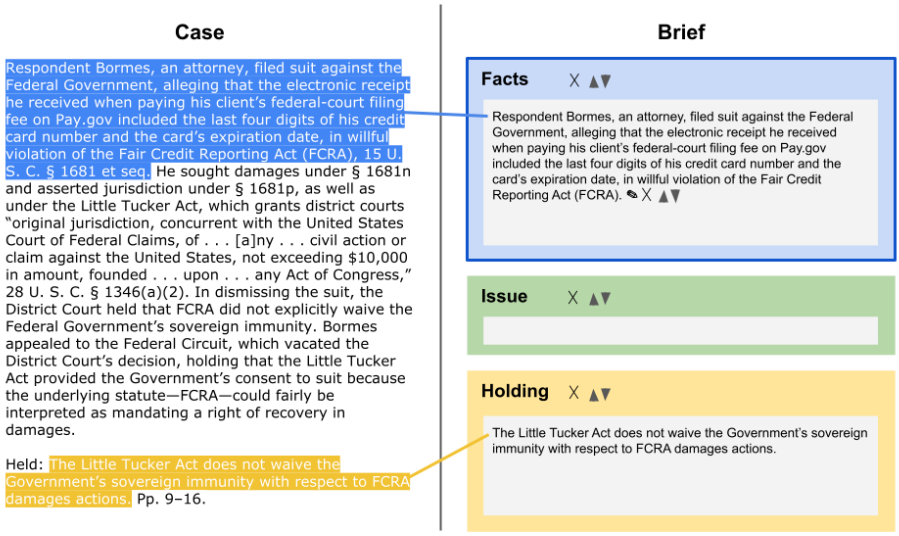


Figure 1. An overall design of CABINET's user interface: The analyzed document is displayed on the left. The case brief sections are populated and edited on the right. The system preserves the mapping between the original text and the resulting case brief sections.

To this end we adopt the National Institutes of Health's competencies proficiency scale[2] (NIH proficiency scale), a highly-regarded instrument used to measure one's ability to demonstrate competency in a task.

Figure 2 shows how the system adapts and adjusts to the proficiency level of the user as indicated by the NIH scale, from level 1 (Fundamental Awareness) to level 5 (Expert). Initially, the system provides the user with reference answers and explanations to provide an adequate level of challenge and timely feedback (blue in Figure 2). As the user learns, they are able to perform more tasks themselves (green). The system takes on the role of a safe-guard against apparent mistakes relying on one of its ML components. In the latter stages the system's ML components take over the initial steps in performing the work (orange) and a user (now an expert) reviews the results and corrects mistakes.

Fundamental Awareness (NIH level 1) - An individual at this level has common knowledge for understanding basic techniques and concepts. The learner is aware of the concept of briefing a case, its purpose and value, and is superficially aware of case brief sections. At this level, CABINET leverages the so-called *worked example effect*: studying worked examples appears to be more effective than learning by solving the equivalent problems. [32] This effect has also been confirmed in the related area of reasoning about legal cases. [20] As shown in Figure 2, the learner's task is to use the CABINET interface to inspect and reflect on cases annotated by legal education experts. The interface also provides explanations justifying the choices of the expert. This stage relies on the in-depth annotation of a small number of cases (tentatively 10-20 cases).

Novice (NIH Level 2) - At this level, an individual has the level of experience gained in a classroom and/or experimental scenarios or as a trainee on the job. They are expected to need help when performing a skill. Learners at this level are ready to attempt to categorize key argument elements with respect to case brief sections. As shown in the

[2] https://hr.nih.gov/working-nih/competencies/competencies-proficiency-scale

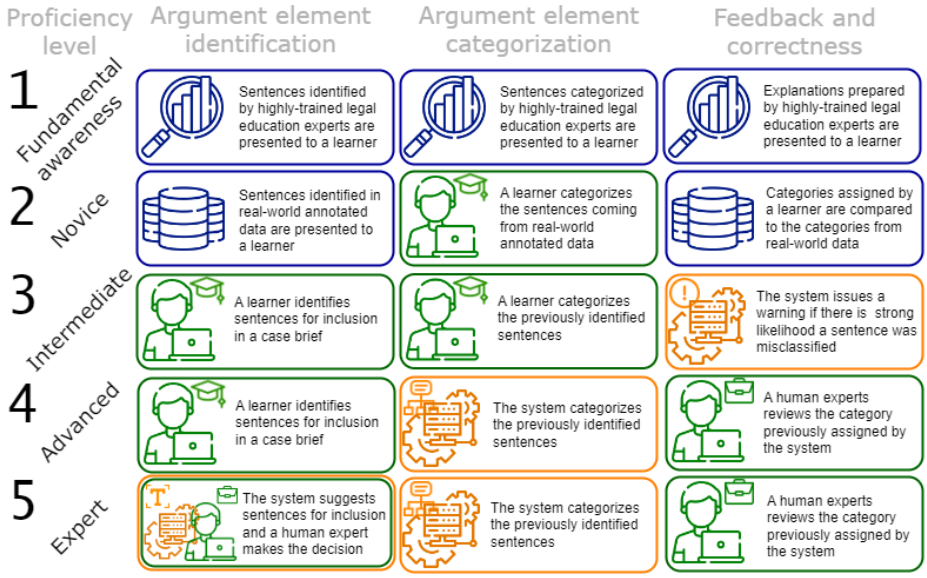


Figure 2. The rows of the diagrams correspond to the five NIH proficiency levels. The columns represent tasks performed as cooperation between a user and a computer. The performance of the tasks is either based on static expert data that have been pre-annotated (blue), user's work (green), or ML component (orange).

second row of Figure 2, learners are presented with texts where the key argument elements have already been identified by legal experts. The learner performs the argument element categorization task. Their choices are compared to those of the experts and the learner is notified about a mismatch and the category assigned by an expert is revealed.

Intermediate (NIH Level 3) - An individual at this level is able to successfully complete tasks as requested with occasional expert help. At this level, the learner attempts to identify key argument elements in a text and to categorize them. As shown in the third row of Figure 2, CABINET assumes the role of a safe-guard which evaluates the work of the advanced learner. Here, the feedback comes from a ML component that identifies the sentences that have a very high likelihood of being placed in the wrong section. The feasibility of such an ML model is evaluated in Section 5 (Experiment 1).

Advanced (NIH Level 4) - At this level, an individual can perform the actions associated with the skill without assistance. It is assumed that a trained professional can independently identify the key argument elements in a text as well as classify them with the correct case brief section. At this stage, CABINET employs a classification model to predict the case brief section of an argument element that a user identified for inclusion in the case brief. The user is expected to evaluate the predictions and correct potential errors. Some errors are tolerable at this stage, since the user is proficient enough to efficiently correct them. The feasibility of such a model is evaluated in Section 5 (Experiment 2).

Expert (NIH Level 5) - At this level, an individual is an expert in a given area. They can provide guidance, troubleshoot and answer questions related to this area of expertise and the field where the skill is used. CABINET provides the same kind of assistance as in the previous stage, but uses more active ways of supporting the professional at this level. Specifically, CABINET subtly highlights passages in a text with colors corresponding to predicted case brief sections and intensity corresponding to system's confidence in the

passage being a key argument element. This is achieved by a ML component applied to a full text. Since such predictions cannot be performed with a high degree of reliability (see Section 3), the highlighted text passages are only meant to augment the expert's review of a case text, allowing them to identify the key argument elements more efficiently.

5. Experiments

To examine the framework's feasibility, we assess two hypotheses that correspond to the system's key capabilities described in Section 4. Given a sentence in a case brief:

(H1) ... it is possible to detect if the sentence is in an *incorrect* section.
(H2) ... it is possible to predict the *correct* section for the sentence.

The capability assessed by H1 is deployed at the Intermediate proficiency level (NIH Level 3), to warn a user when a sentence is likely assigned to an incorrect case brief section. The capability assessed by H2 is utilized at the Advanced and Expert proficiency levels (NIH Levels 4 and 5), to predict the correct section for a text passage identified by a user as a key argument element.

5.1. Dataset

We obtained a dataset of 715 unique case briefs by scraping a publicly available Case Brief Summary database.[3] We used an extensive battery of regular expressions to segment the retrieved briefs into individual sections corresponding to the key argument element types. While there were over 100 unique section names we identified the six main types (see Section 2) to which we could map many of the different variations (e.g., all of "Legal Issue", "Issues", and "Issue" map to a single category). We applied a specialized legal case sentence boundary detection system to segment the sections into 9,924 sentences. [27] Figure 3 shows the distribution of the sentences in terms of the key argument element types from the perspective of their overall counts as well as their distribution over the individual case briefs. We divide the dataset into random splits on a document basis. The splits are used for training (70%), validation (15%) and testing (15%).

5.2. Model and Training

As a **baseline** we use a simple model that randomly predicts the labels based on their frequency in the dataset. We use the standard implementation provided by the sklearn framework, with "stratified" sampling.[4] As the main model we employ **RoBERTa** (a robustly optimized BERT pretraining approach) developed by Liu. [17] The model was chosen due to its high performance and simplicity to train. While higher performance might be achieved by more recent models, the RoBERTa model suffices for the purpose of proving the feasibility of the key components of the framework. Out of the available

[3] Accessible at: http://www.casebriefsummary.com/. Currently, the website appears to be offline. However, it has been archived by the Web Archive project at https://web.archive.org/web/20200927234341/http://www.casebriefsummary.com/

[4] https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html

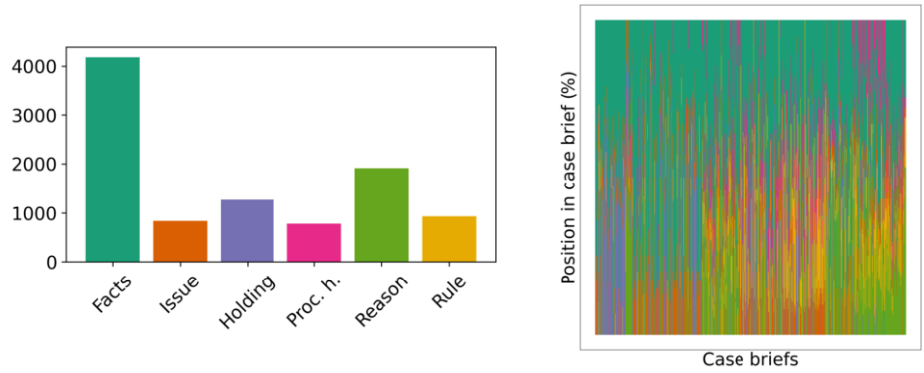


Figure 3. Left: Distribution of sentences in the dataset by section. Right: Location of sentence classes (y-axis) across the different case briefs (x-axis).

Table 1. Statistics about warning and abstentions for section assignments, for thresholds 0.05, 0.1 and 0.2.

	Warn	Abstain
Warn	5952	122
Abstain	1248	1318

	Warn	Abstain
Warn	6282	169
Abstain	918	1271

	Warn	Abstain
Warn	6549	256
Abstain	651	1184

models we chose to work with the smaller *roberta.base* that has 125 million parameters, for faster iteration times. Due to the sentences being short, we did not have to address the model's sequence length limitation of 512 tokens. The training set is used to train the model for 4 epochs. At each epoch, we evaluate the performance of the model against the evaluation set and pick the best-performing model for our experiments.

5.3. Testing H1: Warning about an Incorrectly Categorized Argument Element

When interacting with an Intermediate user (NIH Level 3) CABINET issues a warning if a piece of text identified by the user as a key argument element has been (highly likely) assigned with to an incorrect section (see Section 4). Hence, the input is a short text together with a label assigned by the user. The system either issues a warning about the assignment being likely incorrect or abstains.

We transform the dataset by creating text-label pairs between each sentence and all the labels. Since there are 1,440 sentences in the test set and 6 unique labels, there are 8,640 such pairs. For each pair, we retrieve the probability distribution the model assigns over the possible labels. If the value for a given pair is below a static threshold (we experiment with 0.05, 0.1 and 0.2), the system issues a warning. It is crucial to minimize the number of false positives (i.e., issuing a warning when the user-assigned label is in fact correct). It is comparatively less important to treat false negatives (i.e. missing out on an incorrect assignment). Since the user is still in the process of learning, abstaining in case of a mistake is preferable to providing the user erroneous (confusing) feedback.

The results for the three thresholds are reported in Table 1. The columns correspond to whether the warning should be raised, whereas the rows correspond to whether the model would raise the warning or abstain. The diagonal (cells shaded in green) reports the number of pairs for which the model behaves as desired. The cells outside of the diagonal (shaded in red) report the disagreements.

Table 2. Performance (left) and confusion matrix (right) for class predictions.

	Baseline			RoBERTa		
Argument Type	P	R	F_1	*P*	*R*	F_1
Facts	.46	.47	.47	.90	.80	.85
Issue	.05	.05	.05	.96	.91	.93
Holding	.13	.17	.15	.42	.53	.47
Procedural History	.06	.06	.06	.66	.81	.73
Reasoning	.23	.17	.20	.56	.67	.61
Rule	.06	.07	.07	.65	.48	.55
Weighted Avg	.28	.27	.27	.76	.73	.74

	Facts	Issue	Holding	Proc. H.	Reason	Rule
Facts	524	0	9	14	31	3
Issue	2	122	0	0	0	3
Holding	25	4	72	4	46	21
Proc. H.	35	3	6	95	3	1
Reason	59	4	36	5	181	38
Rule	8	1	13	0	11	61

5.4. Testing H2: Categorizing Key Argument Elements Automatically

When interacting with the Advanced and Expert users (NIH Levels 4 and 5) CABINET automatically categorizes the key argument elements identified in the text by the user (see Section 4). This is a straightforward sentence classification task. For this component, a certain amount of error is tolerable since (a) the user's proficiency is relatively high and (b) the user is actively involved in selecting the sentences. Hence, they are in a good position to verify and potentially correct and confirm the system's category assignment.

To evaluate the ability of automatically categorizing argument elements, we compare predictions of the trained RoBERTa model to the baseline. As shown in Table 2, it appears the performance differs considerably across types. Facts and Issue argument element types are identified more reliably than Holding or Rule. Table 2 shows a confusion matrix over which classes are frequently confused with other classes. The predicted labels are shown in the rows, and the true labels in the columns. For example, we can see that holdings are frequently confused with reasoning, which may be due to the small size of the classes or the classes having low "semantic homogeneity", compare [38].

6. Discussion and Future Work

The results of the experiment evaluating H1 show that the false positive rate (see Table 1) varies between 2.0% and 3.8%, depending on the threshold. This rate appears to be acceptable given the envisioned use case and user's proficiency level (Intermediate - NIH Level 3). The false negatives rate (i.e., the system abstains when a warning should have been raised) varies between 48.6% and 35.5%. While such a rate is high, we argue that in case of an isolated error due to factors such as fatigue, stress, or lack of attention, the missed warning is tolerable. If a learner has a systematic misconception, the user errors will repeat and the system will likely detect a larger portion of those. Hence, the learner will receive clear and timely feedback triggering further learning.

Evaluation of H2 shows promising performance of the fine-tuned RoBERTa model, although the performance is far from perfect. This is acceptable since this component supports a user at Advanced or Expert proficiency level (NIH Levels 4 and 5). Therefore the potential to confuse a user by an incorrect prediction is relatively low. Since the user actively selects the argument element and is immediately presented with a prediction they are in a comfortable position to perform a correction. We argue that it is far more efficient to inspect automatic predictions and make corrections when needed (in about 25% of predictions), compared to categorizing the key argument elements manually.

While the experimental results confirm our working hypotheses, there are several important limitations to the presented study. *First*, the design of the experiments only takes into account passages of text that have been selected for inclusion in the case briefs by legal experts. However, the user may occasionally make mistakes in their selections. This is particularly true for users at the Intermediary proficiency level (NIH level 3). *Second*, we do not address the challenge of assessing the current level of proficiency of the user. *Third*, the functionality of the system at the Fundamental Awareness (NIH Level 1) proficiency level requires a limited but highly curated dataset of annotated cases with detailed feedback addressing common misconceptions—a resource we have not yet created. *Fourth*, we did not conduct a feasibility study of the highlighting functionality at the Expert proficiency level (NIH Level 5). *Fifth*, and most importantly, we have not conducted a pilot user study to tentatively gauge the expected improvements in learning outcomes. We plan to address these limitations in future work.

7. Conclusion

We proposed an adaptive environment to support case law analysis based on a novel cognitive computing framework that matches various ML capabilities to the proficiency of a user. We have shown how the environment could (i) support a learner in mastering the skill of identifying key argument elements in a court opinion, and (ii) support a professional in performing the same task more efficiently. We have demonstrated that it is possible to detect if a sentence is placed in an incorrect section in case brief (H1), and to predict the actual argument element type of a case brief sentence (H2) with a reliability sufficient for the envisioned use case based on the proficiency level of a user. Hence, we have taken the initial steps in establishing the feasibility of the proposed system.

Acknowledgements Hannes Westermann and Karim Benyehklef acknowledge the generous support from the Cyberjustice Laboratory, LexUM Chair on Legal Information, and Autonomy through Cyberjustice Technologies project. Figure 2 has been designed using resources from Flaticon.com.

References

[1] Aleven, V. "Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment." *Artificial Intelligence*. v. 150, nn. 1-2, pp. 183–237. Elsevier. 2003.
[2] Ambrose, S. A. et al. *How Learning Works*, John Wiley & Sons, 2010.
[3] Bhattacharya, P., et al. "Identification of Rhetorical Roles of Sentences in Indian Legal Judgments." arXiv preprint arXiv:1911.05405 (2019).
[4] Bittencourt, I., Costa, E., Fonseca, B., Maia, G., and Calado, I. "Themis, a Legal Agent-based ITS." AIED Applications in Ill-Defined Domains. p. 11. 2007.
[5] Black, P., and William, D. Assessment and classroom learning. *Assessment in Education*, 5, 7–74, 1998.
[6] Brostoff, T. and Sinsheimer, A. (2013). *United States Legal Language and Culture: An Introduction to the US Common Law System*. Ch. 3. Third Edition, Oxford University Press. 2013.
[7] Cormack, G., and M. Grossman. "Autonomy and reliability of continuous active learning for technology-assisted review." *arXiv preprint arXiv*:1504.06868 (2015).
[8] Ericsson, K. A., Krampe, R. T., and Tescher-Romer, C. The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100, pp. 363–406, 2003.
[9] Farzindar, A. & G. Lapalme. "Letsum, an automatic legal text summarizing system." JURIX, 2004.

[10] Grabmair, M., et al. "Introducing LUIMA: an experiment in legal conceptual retrieval of vaccine injury decisions using a UIMA type system and tools." Proc. 15th Int'l Conf. on AI and Law. 2015.
[11] Hachey, B. and C. Grover. "Extractive summarisation of legal texts." AI and Law 14, 305-345, 2006.
[12] Harašta, J., et al. "Automatic Segmentation of Czech Court Decisions into Multi-Paragraph Parts." Jusletter IT 4.M (2019).
[13] Hattie, J., and Timperley, H. The power of feedback. *Rev. of Educational Research*, 77, 81–112, 2007.
[14] Healy, A. F., Clawson, D. M., and McNamara, D. S. The long-term retention of knowledge and skills. In D. L. Medin (Ed.), *The psychology of learning and motivation*, pp. 135–164, 1993.
[15] Hogan, C., R. Bauer, and D. Brassil. "Human-aided computer cognition for e-discovery." In *Proc. 12th Int'l Conf. on Artificial Intelligence and Law*, pp. 194-201. 2009.
[16] Licklider, JCR. "Man-computer symbiosis." IRE trans. on human factors in electronics 1 (1960): 4-11.
[17] Liu, Y., et al. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).
[18] Makdisi, M., and Makdisi, J. How to write a case brief for law school. *Introduction to the Study of Law: Cases and Materials*. 3rd Ed, LexisNexis, 2009.
[19] Moens, M.-F., "Summarizing court decisions." Info. Processing & Management 43, 6, 1748-1764. 2007.
[20] Nievelstein, F. et al. The worked example and expertise reversal effect in less structured tasks: Learning to reason about legal cases. *Contemporary Educational Psychology*, 38(2), pp. 118–125, 2013.
[21] Muntjewerff, A. and Breuker, J. "Evaluating PROSA, a system to train solving legal cases." Proceedings of AIED. pp. 278–285. IOS Press Amsterdam. 2001.
[22] Pinkwart, N., Ashley, K., Lynch, C., and Aleven, V. "Evaluating an intelligent tutoring system for making legal arguments with hypotheticals." Int'l J. of AI in Education. v. 19, n. 4, 401–424. IOS Press. 2009.
[23] Routen, T. "Reusing formalisations of legislation in a tutoring system." AI Review 6, 145 – 159, 1992
[24] Saravanan, M., B. Ravindran, and S. Raman. "Improving legal document summarization using graphical models." Frontiers in Artificial Intelligence and Applications 152 (2006): 51.
[25] Šavelka, Jaromír, Gaurav Trivedi, and Kevin D. Ashley. "Applying an interactive machine learning approach to statutory analysis." In *Legal Knowledge and Information Systems*, pp. 101-110. 2015.
[26] Savelka, J., & Ashley, K. "Using conditional random fields to detect different functional types of content in decisions of U.S. courts with example application to sentence boundary detection." ASAIL 2017.
[27] Savelka, Jaromir, et al. "Sentence boundary detection in adjudicatory decisions in the united states." Traitement automatique des langues 58 (2017): 21.
[28] Savelka, J. & Ashley, K. "Segmenting US Court Decisions into Functional and Issue Specific Parts." JURIX. 2018.
[29] Savelka, Jaromir et al. "Lex Rosetta: Transfer of Predictive Models across Languages, Jurisdictions, and Legal Domains." ICAIL 2021, 129–38. https://doi.org/10.1145/3462757.3466149.
[30] Savelka, Jaromir, Hannes Westermann, and Karim Benyekhlef. "Cross-domain generalization and knowledge transfer in transformers trained on legal data." *ASAIL*, 2020.
[31] Sovrano, F., K. Ashley, P. Brusilovsky, and F. Vitali. "YAI4Edu: an Explanatory AI to Generate Interactive e-Books for Education." 4th Int'l Wkshp on Intelligent Textbooks. CEUR 3192, 31–39, 2022.
[32] Sweller, J., 2006. The worked example effect and human cognition. *Learning and instruction*.
[33] Walker, V.R., et al. "Automatic Classification of Rhetorical Roles for Sentences: Comparing Rule-Based Scripts with Machine Learning." ASAIL@ ICAIL. 2019.
[34] Walker, V. R., et al. "Semantic types for computational legal reasoning: propositional connectives and sentence roles in the veterans' claims dataset." Proc. 16th Int'l Conf. AI and Law. 2017.
[35] Waltl, B., J. Muhr, I. Glaser, G. Bonczek, E. Scepankova, and F. Matthes. "Classifying Legal Norms with Active Machine Learning." In JURIX, pp. 11-20. 2017.
[36] Westermann, H., et al. "Computer-Assisted Creation of Boolean Search Rules for Text Classification in the Legal Domain." JURIX. 2019.
[37] Westermann, H., Savelka, J., Walker, V., Ashley, K. & Benyekhlef, K. "Sentence Embeddings and High-Speed Similarity Search for Fast Computer Assisted Annotation of Legal Documents." *Jurix 2020*.
[38] Westermann, H. et al. "Data-Centric Machine Learning in the Legal Domain." ArXiv:2201.06653 [Cs].
[39] Wyner, Adam, et al. "Approaches to text mining arguments from legal cases." Semantic processing of legal texts. Springer, Berlin, Heidelberg, 2010. 60-79.
[40] Xu, H., Savelka, J., Ashley, K. "Using Argument Mining for Legal Text Summarization" JURIX. 2020.

Short Papers

This page intentionally left blank

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220458

Unpacking Arguments

Trevor BENCH-CAPON [a], and Bart VERHEIJ [b]
[a] *Department of Computer Science, University of Liverpool*
[b] *Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence, University of Groningen*

Abstract. Although argumentation is often studied in AI using abstract frameworks, actual debate often shows a dynamic interaction between argument structure and attack. Often intermediates steps in the reasoning are omitted, but it may be these intermediate steps which are the vulnerable parts of the argument. Inspired by Loui and Norman's work on the rationale of arguments, we study the relation between argument structure and attack in terms of the unpacking of arguments. The paper provides an analysis of two kinds of rationales discussed by Loui and Norman. Example dialogues inspired by Dutch tort law are used for illustration.

Keywords. Computational argumentation, Legal reasoning, Argument attack

1. Introduction

Abstract argumentation [1] has been very influential in AI. It provides a rich formal analysis of the evaluation of the status of a set of arguments and is applicable to many application settings, but takes the set of arguments and the attack relations between them as given. When generating arguments from a knowledge base, as is done in ASPIC+ [2], this presents no problems. But when modelling arguments proposed in actual disputes things may be more difficult. Often some intermediate steps of the argument are unstated, so that while a denial that the intermediate step is valid can attack the argument, this is not clear from what has been explicitly stated. That the vulnerable component is unstated makes the argument harder to attack because the weak point needs to be supplied by the audience. To address such interaction between argument structure and attack, Loui and Norman proposed [3] that arguments should be unpacked to uncover their *rationales*, restoring intermediate steps glossed over in the original presentation, so that vulnerabilities can be identified, attacked and, where possible, defended. The idea can also be recognized in the jurisprudential practice of "rational reconstruction".

Legal arguments are typically multistep, as we move from evidence to facts to factors to issues before finally arriving at a decision [4]. As well as being multi-step, decisions in legal cases often turn on preferences and social values promoted [5], as expressed in precedent cases. The preferences are usually not explicit, but must be recognised if the justification is to be properly understood, and may provide a way to critique the decision.

These two aspects of arguments about legal cases, the omission of intermediate steps and the implicit preferences, correspond to two of the rationales identified in [3], the c-rationale (*compression* rationale) and the r-rationale (*resolution* rationale). We believe that unpacking arguments to restore these aspects is important if the arguments are to be

properly understood, attacked and defended. In this paper, therefore, we revisit Loui and Normans's paper from a contemporary perspective.

Section 2 will summarise their paper, with particular attention to the c-rationales and r-rationales. Section 3 will relate their work to subsequent work on argumentation, in particular work on argument structures such as ASPIC+ [2] and DEFLOG [6]. Section 4 gives some discussion and concluding remarks.[1]

2. Interpreting Loui and Norman's work on unpacking arguments

A key idea in [3] is that the rationales used in an argumentative dialogue can be interpreted as the summaries ('compilations') of extended rationales with more structure. Loui and Norman show how, by unpacking such summary rationales, new argument moves are possible. Thus if someone claims *C because of reason R*, an opponent may identify an immediate step so that the claim becomes *C because of S, and S because of reason R*. Now the opponent can argue that, despite *R*, *S* does not hold for reason *T*, invalidating the conclusion *C*. A defence against this is to provide a different unpacking, claiming that *C* holds because of *U* which is established by *R*, even when *T* is true.
Here a proponent makes an argument, and the opponent unpacks that argument in a certain way, and uses that unpacking to make an attack. The proponent concedes the attack, but disagrees with the unpacking, thereby defending the original position.

Loui and Norman distinguish rationales for rules and rationales for decisions. In the authors' terminology, rule rationales express mechanisms for adopting a rule, while decision rationales express mechanisms for forming an opinion about the outcome of a case. As kinds of rule rationales, the authors distinguish compression, specialization and fit (referred to as c-rationales, s-rationales and f-rationales). The kinds of decision rationales are disputation and resolution (d-rationales and r-rationales). Another distinction used is that between object-level and meta-level disputation, where c-, s-, and d-rationales occur in object level disputes, and r- and f-rationales in meta-level discussion.

We will consider two of these rationale types which are common in legal cases as noted above. For a rule rationale we will look at c-rationales, where the unpacking restores missing intermediate steps. For a decision rationale we look at r-rationales, which identify the use of preferences.

2.1. Unpacking a compression rationale

Here is an example of a small dialogue in which a compression rationale is unpacked, subsequently attacked and then defended, following the basic format given earlier. The unpacking here has the form of adding an intermediate step, thereby interpreting a one step argument as a two step argument. Our example is from Dutch tort law (for legal background, see, e.g., [8]).

> A: I claim that there is a duty to pay the damages (`dut`) because of the act that resulted in damages (`act`).
> B: Unpacking your reasoning, you seem to claim `dut` because of `act` using the additional intermediate reason that there is a breach of contract (`bre`). I disagree with `bre`, because there

[1] The present paper builds on and extends the discussion of Loui and Norman's paper [3] in [7]. The second author acknowledges support by the NWO Zwaartekracht Hybrid Intelligence project.

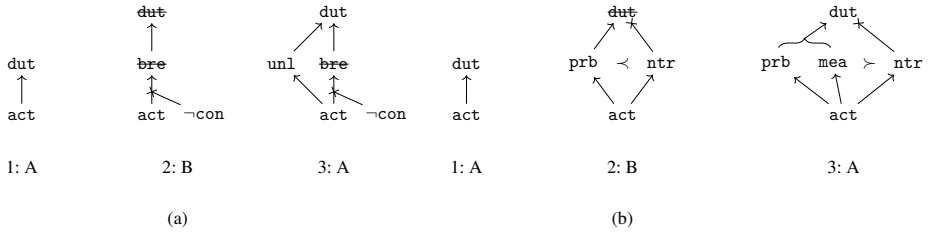


Figure 1. Unpacking (a) a compression rationale and (b) resolution rationale

was no contract (¬con), so there is no support for bre. Hence there is also no support for dut.

A: I agree with your reason ¬con and that hence there is no support for bre. But I was not using bre as an intermediate step supporting dut. Instead I used the intermediate step that the act was unlawful (unl), hence my claim dut because of act.

A graphical summary of the 3-step dialogue is shown in Figure 1. Normal arrows indicate a supporting reason and arrows ending in a cross indicate an attacking reason. All abbreviated statements are considered to be successfully supported, except those that are struck-through. Writing the first argument by A as act → dut, B replies in the second move by interpreting the argument as actually having two steps, act → bre → dut, and then attacks the unpacked argument on the intermediate step using the argument ¬con, so that bre and dut are no longer successfully supported. But then at the third step A concedes that ¬con, while denying the unpacking via bre, instead claiming the unpacking act → unl → dut, providing an alternative way to support dut, thereby still maintaining act → dut. B unpacks A's rationale into two steps, attacking the first step. As a result, for B, dut has no successful support. A accepts everything that B has said, but provides an alternative unpacking of the rationale, via unl, so justifying dut.

2.2. *Unpacking a resolution rationale*

Again we give a mini-dialogue, illustrating how the idea of resolution rationales and argument attack is approached in [3]. The example unpacks an argument as expressing a preference between two conflicting reasons.

A: I claim that there is a duty to pay the damages (dut) because of the act that resulted in damages (act).

B: Unpacking your reasoning, you seem to claim dut because of act using the weighing of two reasons, one for the duty to pay (the high probability of damages, prb) and one against (the mild nature and low scale of the possible damages, ntr). I disagree with this weighing, because the nature and scale of the possible damages was exceptionally low and so outweighs the high probability of damages. Hence I disagree with your claim dut because of act.

A: I agree with your weighing of the two reasons you mention. But I was using an additional reason for the duty to pay (it was easy to take precautionary measures, mea), and the two reasons for the duty to pay taken together (prb ∧ mea) outweigh the one reason against (ntr). Hence my claim dut because of act.

A graphical summary is shown in Figure 1(b). If we write A's argument in the first dialogue move as act → dut, then in the second move B unpacks A's reasoning by claiming that A has weighed prb → dut and ntr → ¬dut. According to B's weighing

of these two reasons, the conclusion should be ¬dut. In the third step, A agrees with B's weighing of the two reasons, but adds a third reason (mea → dut) that turns the outcome to the other side, concluding for dut.

2.3. Characteristics of unpacking arguments

From this interpretation we can see that we can use unpacking to attack and defend an argument in a dialogue as follows:

1. *An argument is unpacked.* We have seen two kinds of unpacking: first by interjecting an intermediate step, and second by decomposing the argument as the preference-based resolution of a intermediate conflict of reasons.
2. *A new attack is made.* We have seen that a new intermediate step is attacked by an exceptional circumstance, and that the new interpretation of the argument as the resolution of a conflict of reasons is given an opposite outcome by a reversed preference.
3. *A new defence is made.* We have seen an alternative unpacking of the argument, immune to the new attack, and an extended set of conflicting reasons maintaining the original preference-based resolution.

3. Applying later developments to unpacking arguments

Loui and Norman's paper [3] appeared in the same year as Dung's paper [1], which significantly influenced the subsequent formal and computational study of argumentation. In this section, we study the unpacking of arguments as discussed in Section 2, in terms of subsequent developments.

3.1. Abstract and Structured Argumentation

Following the publication of Dung's formal study of the semantic evaluation of argument attack relations [1], his abstract argumentation frameworks became the standard reference for the semantics of argumentation. In that paper, directed graphs are used to represent argument attack, and the key evaluative principle is that an argument is accepted if there is no accepted attacking argument, and rejected otherwise. The approach is referred to as 'abstract argumentation', because the arguments in Dung's framework have no properties other than the attack relation.

Often, however, it is necessary to consider the structure of the arguments. This is particularly so in law where a claim is not useful without its justification. One approach to representing structure is ASPIC+ [2], in which the arguments can be seen as comprising subarguments, and the attacks distinguished according to the element of the argument (conclusion, premise or inference rule) that is attacked.

3.1.1. Compression

Figure 2 illustrates the above compression rationale dialogue. On the left the developing abstract framework is shown using the notation of [1]. On the right the structured arguments are shown using a representation inspired by the ASPIC+ framework [2]. extended

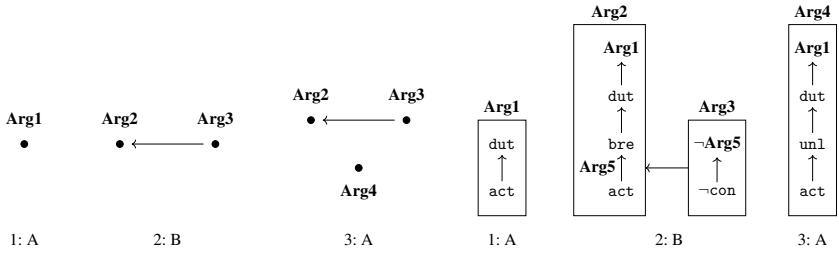


Figure 2. Unpacking a compression rationale using abstract argumentation

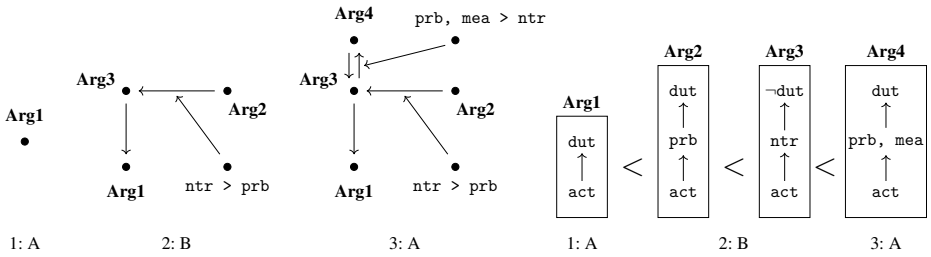


Figure 3. Unpacking a resolution rationale using abstract argumentation

to allow arguments to be conclusions as well as premises. At the first move, there is one argument, Arg1, and there is no attack. After B's move there are two arguments, Arg2 and Arg3. Arg2 is introduced as the justification for Arg1, and so Arg1 moves *inside* the node. Arg3 attacks Arg2 on a subargument (Arg5). so only Arg3 is accepted. In the third move a different justification of Arg1, Arg4, is given. This is not attacked. Now Arg1 is accepted, but only as the claim of the accepted Arg4, rather than as an argument in its own right. At the level of abstraction of Arg1, where all intermediate structure is ignored, Arg2 and Arg4 are indistinguishable. But it matters whether A was originally putting forward Arg1 as an abstraction of Arg3 or Arg4: in law it is important not only that the correct claims are accepted, but that the correct justification for them is given. Note also that in the abstract framework the relationship between Arg1 and Args 2 and 4 is lost.

3.1.2. Resolution

ASPIC+ also allows for preferences between arguments, allowing us to model the resolution rationale. Here B shows that Arg1 requires the resolution of a preference between Arg2 and Arg3, and resolves it so as to deny the claim of Arg1. A now responds by finding a second reason for the original claim, Arg4, so that the *combination* of `prb` and `mea` can be preferred to `ntr`, defeating Arg3 and reinstating Arg1. The situation is shown graphically in Figure 3, with the abstract version on the right shown as an Extended Argumentation Framework [9], which expresses preferences as attacks on attacks.

3.2. Sentence-based argument structure

Second we discuss a sentence-based approach to argument structure. In such an approach, argument support and attack are not treated separately by first determining supporting arguments and then abstracting from them by focusing on attack (as in the ap-

proach in the previous subsection). Instead, all argument structure is expressed by explicit sentences, using dedicated sentences for both support and attack.

Concretely, the argument structure and its evaluation as suggested in Figure 2 can be reconstructed in the DefLog formalism [6], as follows. Three sets of sentences represent the assumptions made by A and B in the three moves:

A_1: act; act $\rightsquigarrow$ dut (justifying dut)
B_2: act; act $\rightsquigarrow$ bre; bre $\rightsquigarrow$ dut; $\neg$con; $\neg$con $\rightsquigarrow \times$(act $\rightsquigarrow$ bre)
(neither justifying bre nor dut since it follows that $\times$(act $\rightsquigarrow$ bre), hence act $\rightsquigarrow$ bre is defeated)
A_3: act; act $\rightsquigarrow$ bre; bre $\rightsquigarrow$ dut; $\neg$con; $\neg$con $\rightsquigarrow \times$(act $\rightsquigarrow$ bre); act $\rightsquigarrow$ unl; unl $\rightsquigarrow$ dut
(justifying unl and dut, but not bre)

Note that in this reconstruction in the second move B revises the commitments made by A in the first move (in the sense that B does not assume A's assumption act $\rightsquigarrow$ dut), while A in the third move commits to all assumptions made by B in the second (in the sense that all B's assumptions are also assumed by A).

4. Concluding Remarks

In Section 3, we discussed unpacking arguments using developments in computational argument that appeared after [3]. Both abstract argumentation and a sentence based approach can make the unpackings explicit, but neither retains the connection between the unpacked and the unpacking arguments. This could be studied using case models [10].

Finally, unpacking arguments should not be confused with identifying enthymemes. Unpacking identifies additional, possibly dubious, arguments, rather than assumptions. In terms of the knowledge base, what is made explicit is a *rule*, rather than a *statement*.

References

[1] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence. 1995;77:321-57.
[2] Prakken H. An Abstract Framework for Argumentation with Structured Arguments. Argument and Computation. 2010;1(2):93-124.
[3] Loui RP, Norman J. Rationales and Argument Moves. Artificial Intelligence and Law. 1995;3:159-89.
[4] Bench-Capon T, Atkinson K. Using Argumentation Schemes to Model Legal Reasoning. arXiv preprint arXiv:221000315: Presented at 4th European Conference on Argumentation. 2022.
[5] Bench-Capon T, Sartor G. A Model of Legal Reasoning with Cases Incorporating Theories and Values. Artificial Intelligence. 2003;150(1):97-143.
[6] Verheij B. DefLog: on the Logical Interpretation of Prima Facie Justified Assumptions. Journal of Logic and Computation. 2003;13(3):319-46.
[7] Governatori G, , Verheij B, Araszkiewicz M, Bench-Capon T, Francesconi E, et al. Thirty Years of AI and Law: The first decade. AI and Law. 2022 In Press;30(4).
[8] Verheij B, Hage JC, Lodder AR. Logical Tools for Legal Argument: a Practical Assessment in the Domain of Tort. In: Proceedings of the 6th ICAIL. New York (New York): ACM Press; 1997. p. 243-9.
[9] Modgil S. Reasoning about Preferences in Argumentation Frameworks. Artificial Intelligence. 2009;173(9):901-34.
[10] Verheij B. Formalizing Arguments, Rules and Cases. In: Proceedings of the 16th ICAIL. New York (New York): ACM Press; 2017. p. 199-208.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220459

The Illinois Intentional Tort Qualitative Dataset

Joseph BLASS [a,1] and Kenneth FORBUS [a]
[a] *Qualitative Reasoning Group, Northwestern University*

Abstract. We introduce the Illinois Intentional Tort Qualitative Dataset, a set of Illinois Common Law cases in Assault, Battery, Trespass, and Self-Defense, machine-translated into qualitative predicate representations. We discuss the cases involved, the natural language understanding system used to translate the cases into predicate logic, and validation measures that serve as performance baselines for future AI research using the dataset.

Keywords. Legal Corpus, Predicate Logic, Tort Law

1. Introduction and Background

In Common Law legal systems, cases are resolved by reference to prior cases involving similar claims. Datasets of such cases, and their developers' formal commitments and assumptions, have been important to AI & Law research. We present a dataset of case texts translated into predicate logic using a publicly available natural language understanding system.

Several large datasets collect cases in their original language [1, 2], and researchers using large-scale machine-learning techniques have recently developed legal retrieval and reasoning systems that operate over raw text [3]. But many AI legal reasoning systems require formal machine-interpretable representations, with which researchers have had to annotate their cases. The most widely-used formalism is *factors* [4], legally-salient principles relevant to finding case outcomes and that favor a party. An expert identifies the total list of factors in a domain, then cases are tagged with them (often by hand, but automatically as well [5]). Originally developed for use in HYPO-style reasoners [4], factors and similar formalisms continue to be used in a variety of AI & Law techniques. For example, Horty's *reason model* learns and applies defeasible rules from factored cases [6]. *Abstract dialectical frameworks* encode relationships between arguments and outcomes to generate arguments in factored cases [7]. And Verheij's *case models* are logically consistent sets of cases in propositional logic – not factors, since the statements do not favor a side – that together encode rules [8].

Symbolic case representations have indeed long been useful to AI & Law research. They are a natural fit for legal reasoning, because rationality, explainability, consistency, and transparency of reasoning are hallmarks of good *human* legal reasoning. But in an era of big data and large language models, it is increasingly uncomfortable to rely on human annotation. Semantic interpreters that translate text into symbolic representations

[1] Corresponding Author: Joseph Blass, joeblass@law.northwestern.edu

can split the difference and scale beyond hand-encoding while providing rich representations for legal reasoning.

2. The Companions Natural Language Understanding System (CNLU)

CNLU [9] is the semantic interpreter for the Companions cognitive architecture [10]. Companions use an ontology derived from Cyc [11]. The knowledge base, NextKB, supplements OpenCyc and contains over 18,000 concepts and over 1000 relations constrained by nearly a million facts. Microtheories partition knowledge and can be linked to form logical environments. NextKB meets the ontological requirements described in [12].

CNLU produces hierarchical parse trees using Allen's bottom-up chart parser [13]. At the leaf nodes (lexical tokens), subcategorization frames are used to generate choice sets of nodes' possible semantic interpretations. Consistent sets are selected to create sentence interpretations, manually or automatically [14]. Coreference resolution merges references to the same entity. CNLU operates over simplified English syntax to focus on semantic over syntactic breadth, that is, to prioritize the system's understanding of ideas over sentence forms. CNLU uses Discourse Representation Theory [15] to handle negation, implication, quantification, and counterfactuals using nested discourse representation structures. Those structures are converted to standard CycL representations in the last language processing step.

The lexical information was semi-automatically extracted from a public domain edition of Webster's dictionary, then augmented. Each word, for each part of speech, has one or more semantic translations derived from FrameNet [16] that map the lexicon to the NextKB ontology to express possible meaning. Syntactic analysis maps complements to role relations. For composability, CNLU uses neo-Davidsonian event representation (Figure 1). Thus "Dave eats ice cream" maps Eat-TheWord to an EatingEvent, and Dave to its performer. Finally, *Narrative Functions* (NFs) [17] support interpretation with abductive explanation. NFs operate across choice sets to build parse explanations, making abductive assumptions as needed.

3. The Illinois Intentional Tort Qualitative Dataset

Illinois was chosen as the dataset's jurisdiction because our institution is located there. CNLU cannot currently handle statutes' linguistic complexity and legalistic formalism, so we sought pure common-law doctrines, where judicial opinions express rules in plain English. We included the Tort doctrines of Trespass, Assault, Battery, and Self-Defense. We found cases using WestLaw and LexisNexis and traced their cited and citing cases. Cases were excluded if unreasoned, overturned, decided before 1870, or unrelated to prior or subsequent cases.

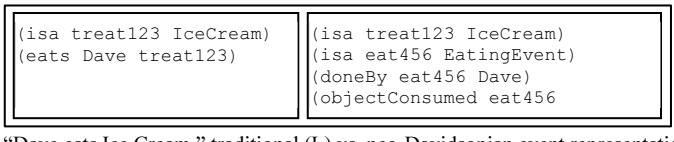


Figure 1. "Dave eats Ice Cream," traditional (L) vs. neo-Davidsonian event representation (R).

Original Text, Bishop v. Ellsworth: "On July 21, 1965, defendants, Mark and Jeff Ellsworth and David Gibson, three small boys, entered [the plaintiff Dwayne Bishop's] salvage yard premises at 427 Mulberry Street in Canton, without his permission, and while there happened upon a bottle partially embedded in the loose earth on top of a landfill, wherein they discovered the sum of $ 12,590 in United States currency. [The] boys delivered the money to the municipal chief of police who deposited it with [the] Canton State Bank. The defendants caused preliminary notices to be given as required by Ill Rev Stats, chapter 50, subsections 27 and 28, (1965)."

Simplified Text: "The plaintiff owns a salvage yard. The defendants are young boys. The defendants entered the salvage yard. The plaintiff did not permit them to enter the salvage yard. The defendants found a bottle containing $12590 of money in the plaintiff's salvage yard. The defendants brought the money to the chief of police. The defendants placed notices about the money in the newspaper."

Figure 2. Original vs. Simplified Case Facts. Bishop v. Ellsworth, 91 Ill. App. 2d 386 (1986)

Collected cases were organized by doctrine and annotated with decision year, court, and case reporter metadata. Case facts and conclusions were manually identified and stored as a string argument to a metadata fact. In eleven cases, appellate courts laid out alternate set of facts left ambiguous by a lower court and identified the corresponding legal conclusions. Each of these was converted into two dataset cases, for each set of facts and conclusion. The dataset comprises 88 cases illustrating 112 distinct tort claims. These include 17 assault cases (12 positive cases where the court found an assault had occurred, 5 negative cases where the legal standard was not met), 40 battery cases (30 positive, 10 negative), 43 trespass cases (29 positive, 14 negative), and 12 self-defense cases (5 positive, 7 negative). Positive cases outnumber negatives because they are more likely to be published and later relied upon.

Judges' descriptions of case facts often include run-on sentences, long lists and descriptions, and asides, so case texts were simplified for CNLU (Figure 2). Parties' names were reduced to party designations. Names (people, places, and things), prices, and dates were removed. For cases with unrelated causes of action, facts identified as only relevant to an unrelated claim were removed. Words not in CNLU's vocabulary were replaced with synonyms (or added to the vocabulary). Longer sentences were broken down into simpler clauses, complex grammatical structures were rephrased, and compound nouns were sometimes rephrased as declarative sentences (e.g., a long sentence referring to "the plaintiff's salvage yard" became several sentences including "The plaintiff owns a salvage yard."). Texts were then processed by CNLU, with choice sets selected manually to ensure maximum fidelity.

Legal reasoning operates over complex real-world situations, so a rich, accurate understanding of legal texts is critical. NFs can infer sentence meaning beyond strict semantics. To illustrate: given the phrase "the plaintiff climbed to the balcony," CNLU might yield the facts in Figure 3: a climbing event, done by the plaintiff, ending at the balcony. Missing is the fact that *the plaintiff is now on the balcony*. Trespass involves being on private property without permission, so that missing fact is needed to understand the plaintiff may be trespassing. Similarly, understanding Assault or Battery means

```
(isa balcony4180 BalconyLevelInAConstruction)
(isa climb4579 Climbing) (isa plaintiff4574 Plaintiff)
(performedBy climb4579 plaintiff4574)
(to-Generic climb4579 balcony4180)
(isa plaintiff4574 Plaintiff)
(toLocation climb4579 balcony4180)
```

Figure 3. "The plaintiff climbed to the balcony."

understanding when actions constitute threats or physical contact, but such information is so obvious to humans that it is rarely explicitly stated. NFs can make such inferences within CNLU's language processing and understanding (not post-processing), to understand what words mean, not just what they say.

We wrote NFs to make commonsense inferences for frequently recurring situations (200 NF detection rules for 93 NFs). Most were to infer (1) object locations (like the climbing example), (2) when events cause damage, (3) transitive ownership (e.g., a building's owner owns its balcony), (4) when events involve physical contact, (5) part/whole relationships, and (6) when actions create new entities (e.g., Alan suing Bob creates a lawsuit entity). To ensure we only wrote language understanding rules, not legal rules, we only wrote rules that would be true outside legal proceedings, and did not write rules to infer legal conclusions (i.e., facts the opinions indicated were true as a matter of law). Still, determining what is a commonsense versus legal inference is tricky; our use of NFs is discussed below in the limitations section.

4. Experimental and Baseline Results

Our approaches build probabilistic relational schemas of cases using a modified version of the SAGE analogical generalization engine [18, 19], and use them to reason about other cases. Schemas and ungeneralized cases are stored in a generalization pool (gpool) and reasoned with by analogy using the Structure Mapping Engine (SME, [20]). SME creates a mapping between two cases and uses it to make candidate inferences (CIs) between them. Given a case at bar, our systems make generalizations from the other cases, retrieve from the gpool using the held out case, map the retrieved case onto the new one, and check the CIs for the held out solution. Solutions are predicates saying who did what to whom, so "Carl trespassed on Dan's lawn by driving on it" might be expressed as `(trespassOnPropertyByAction Carl123 DansLawn456 drive789)`. That the system must generate, not select, an answer measures its understanding. Our techniques are examined more closely in [21].

Given a case at bar, the first technique, *Purely Analogical Precedential Reasoning* (PAPR), creates generalizations from all other cases in the same doctrine, generalizing positive cases in one gpool and negative ones in another. If it finds a legal conclusion amongst the CIs (after retrieval and mapping), it proposes it as the case solution; if not, it retrieves again. To measure the system's ability to solve the case given its case base, not its ability to retrieve a good case on the first try, we have it check its work: if its proposed solution is wrong, it retrieves and checks up to six additional mappings, measuring Precision@6.

Our second technique, *Analogical Reasoning with Positive Generalizations* (ARPG), reflects the fact that it is positive cases that encode legal doctrines: negative cases may have in common only the absence of positive case facts. ARPG depends on the assumption that legal cases have sufficiently different legally-irrelevant facts that schemas of positive examples will encode only legally-relevant facts. ARPG retrieves from only positive generalizations (not ungeneralized examples) and inspects the CIs. If they contain a case conclusion, ARPG checks for *other* CIs. Extra CIs mean some schema fact besides the conclusion is missing in the case, so the case is negative and the

Table 1. Performance of 4 baselines on Illinois Intentional Tort Qualitative Dataset, in % Accuracy

Technique	All Cases	Assault	Battery	Trespass	Self-Defense	Positive	Negative
ARPG	35%	35%	25%	44%	-	10%	97%
PAPR	72%	94%	68%	67%	-	75%	66%
legalBERT	33%	53%	36%	23%	42%	33%	35%
GPT-J	52%	35%	50%	61%	25%	62%	28%

Table 2. Different techniques' performances compared using proportion tests. Significance reported at $p < 0.05$.

Method	GPT-J	legalBERT	PAPR (part'l credit)
ARPG	GPT-J performs better overall, on Battery and Trespass, and on Pos cases. ARPG performs better on Neg cases.	No sig. diff overall. ARPG outperforms legalBERT on Trespass and Neg cases; legalBERT beats ARPG on Pos cases.	PAPR outperforms ARPG overall, by doctrine, and on Pos cases; ARPG outperforms PAPR on Neg cases.
PAPR (part'l credit)	PAPR outperforms GPT-J, overall and on Assault cases and Neg cases.	PAPR outperforms legalBERT, overall and when broken down by Positive v Negative cases.	-
legalBERT	GPT-J outperforms legalBERT overall and on battery, assault, and positive cases.	-	-

absent fact is a missing claim element. If the conclusion is the only CI, the case is positive. ARPG is also evaluated using Precision@6.

Both techniques can test whether the system is partway towards an answer by accepting a conclusion CI in which all but one entities are correct. In a partially-true conclusion CI, the system identifies the tortfeasor and *either* the tortious action or the victim. Here we report performance for ARPG with a strict truth test and PAPR with a partial truth test, our floor and ceiling performances in [21]. Finally, we tested our systems only on Trespass, Assault, and Battery; we are still studying modeling affirmative defenses like Self-Defense.

We report two baselines using off-the-shelf ML techniques. LegalBERT is a BERT model specialized on legal texts [22]. It was tested as a multiple-choice system. We created multiple-choice cases by negating conclusions and reversing party roles. GPT-J is an open-source model based on OpenAI's GPT-3 [23]. GPT-J was fine-tuned and tested on our dataset using holdout and 5-ply cross-fold validation. To test GPT-J, we prompted it with the simplified case facts and had it generate six text completions, which we examined for a simple expression of the conclusion. Performances are reported in Table 1 and compared in Table 2.

5. Discussion and Limitations

We discuss implications the analysis reveals about the dataset; methods are discussed in [21].

A greater number of positive than negative cases, as well as cases where the defendant is the one accused of tortious conduct, may lead statistical methods to accuse the defendant and generally perform well. Indeed, statistical methods outperformed ARPG in positive but not negative cases. To assess a reasoning technique's performance, special attention should be paid to negative cases and those where someone other than the defendant is the accused.

The limitations of CNLU and of the process by which commonsense NF rules were generated must be acknowledged. The dataset does not yet reach the goal of being generated by feeding raw legal text into a language understanding system, because no system can reliably both handle the complexity of legal texts and generate accurate symbolic representations from them. CNLU features three limitations, each of which is an area of active research. First, CNLU still relies on a human simplifying the original text: the complexity of surface forms CNLU can handle has progressed, but it cannot yet handle arbitrarily complex English input. A stopgap solution may be to train a large language model to simplify texts to CNLU's level. Second, for this dataset CNLU's choice sets were manually selected to ensure semantic fidelity to the text. CNLU can automatically select choice sets quite well [14, 24]; we hand-selected because our goal was to create an accurate dataset, not to evaluate CNLU. Third, something like NFs are necessary to express what a text means, not just what it literally says. Common-sense reasoning is a persistent problem in AI research. For now, the options are to create generally-applicable rules, or to accept that facts obvious to humans remain unknown. Because such facts are critical to understanding legal cases and their outcomes, leaving them unknown guarantees that a computer system will either fail to learn legal concepts or will learn the wrong ones. We invite disagreement and discussion on this point.

References

[1] Petrova A, Armour J, Lukasiewicz T. Extracting Outcomes from Appellate Decisions in US State Courts. Proc's of JURIX Conference 2020; p. 133-142.
[2] Grabmair M, Ashley K, Chen R, Sureshkumar P, Wang C, Nyberg E, Walker, V. Introducing LUIMA: an exp't in legal conceptual retrieval of vaccine injury decisions using a UIMA type sys. & tools. ICAIL 2015.
[3] Branting K, Weiss B, Brown B, Pfeifer C, Chakraborty A, Ferro L, Yeh, A. Semi-supervised methods for explainable legal prediction. In: Proc's of 17th ICAIL; 2019; p. 22-31.
[4] Ashley, KD. Toward a computational theory of arguing with precedents. In: Proc's of ICAIL; 1989; p. 93-102.
[5] Bruninghaus S, Ashley KD. Predicting Outcomes of Case Based Legal Arguments. ICAIL; 2003; p. 233-242.
[6] Horty JF. Reasons and Precedent. In: Proc's of 13th ICAIL; 2011; p. 41-50.
[7] Al-Abdulkarim L, Atkinson K, Bench-Capon T, Whittle S, Williams R, Wolfenden C. Noise induced hearing loss: Building an application using the ANGELIC methodology. Argument & Comp., 2019; 10(1):5-22.
[8] Verheij B. Formalizing Arguments, Rules and Cases. In: Proc's of ICAIL; 2017; p. 199-208.
[9] Tomai E, Forbus K. EA NLU: Practical Language Understanding for Cog. Modeling. Proc's of FLAIRS; 2009.
[10] Forbus K, Hinrichs T, Crouse M, Blass, J. Analogies vs Rules in Cog. Arch. In: Adv. in Cog. Sys.; 2020.
[11] Lenat D. CYC: A large-scale investment in knowledge infrastructure. Comms of ACM; 1995; 38(11):33-38.
[12] Ashley KD. Ontological req's for analogical, teleological, and hypothetical legal reas'g. ICAIL; 2009; p.1-10.
[13] Allen JF. Natural Language Understanding. (2nd ed). 1994; Addison Wesley; Redwood City, CA
[14] Barbella D, Forbus K. Analogical word sense disambiguation. In: Adv. in Cog Sys 2013; 2(1):297.
[15] Kamp H, Reyle U. From discourse to logic: Intro to model-theor'c semantics of nat. lang. 1993; Kluwer Acad.
[16] Ruppenhofer J, Ellsworth M, Schwarzer-Petruck M, Johnson CR, Scheffczyk J. FrameNet II: Extended theory and practice. International Computer Science Institute; 2016.
[17] McFate C, Forbus K, Hinrichs T. Using narrative function to extract qualitative information from natural language texts. In: Procs of the AAAI Conf. on AI; 2014; p. 373-379.
[18] Kuehne S, Forbus K, Gentner D, Quinn B. SEQL: Category learning as progressive abstraction using structure mapping. In: Procs of 22nd Mtg. of Cog Sci Society; 2000; p. 770.

[19] Blass J, Forbus K. Conclusion-Verified Analogical Schema Induction. In: Adv. in Cog. Systems; 2022.
[20] Forbus KD, Ferguson RW, Lovett A, Gentner D. Extending SME to handle large-scale cognitive modeling. Cognitive Science 2017; 41(5):1152-1201.
[21] Blass J, Forbus K. Analogical Reas'g, General'n, & Rule Learning for Common Law Reasoning. In prep (draft manuscript available upon request).
[22] Chalkidis I, Fergadiotis M, Malakasiotis P, Aletras N, Androutsopoulos I. LEGAL-BERT: The muppets straight out of law school. arXiv preprint arXiv:2010.02559; 2020.
[23] Floridi L, Chiriatti M. GPT-3: Its nature, scope, limits, and conseq's. Minds and Machines 2020; 30(4).
[24] Ribeiro D, Forbus K. Combining analogy with language models for knowledge extraction. AKBC Conf. 2021.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220460

An Automata-Based Formalism for Normative Documents with Real-Time

Stefan CHIRCOP [a,1], Gordon J. PACE [a] and Gerardo SCHNEIDER [b]
[a] *Department of Computer Science, University of Malta, Msida, Malta*
[b] *Department of Computer Science and Engineering, University of Gothenburg, Gothenburg, Sweden*

Abstract. Deontic logics have long been the tool of choice for the formal analysis of normative texts. While various such logics have been proposed many deal with time in a qualitative sense, i.e., reason about the ordering but not timing of events, it was only in the past few years that real-time deontic logics have been developed to reason about time quantitatively. In this paper we present timed contract automata, an automata-based deontic modelling approach complementing these logics with a more operational view of such normative clauses and providing a computational model more amenable to automated analysis and monitoring.

Keywords. real-time logic, deontic logic, normative systems, legal contracts

1. Introduction

The duality of automata-based and logic-based formalisms has long been acknowledged in computer science. The former excelling on providing a visual and operational model, while the latter provide a more compositional and denotational view, the two approaches complement each other. Automata-based approaches are ideal for describing models and for automated analysis, logic-based approaches for writing specifications. In earlier work, we have developed contract automata [8], an automata-based approach to a class of deontic logics and proved equivalence of expressiveness between the two [3]. The class of logics contract automata addressed was that of logics which have a qualitative notion of time, i.e., caring about the ordering of action occurrence but not the actual real-time elapsed between them.

Since then, various real-time deontic logics and calculi have been proposed as ways to express real-time clauses in normative documents. In this paper we borrow from work done in real-time automata-based formalisms, particularly timed automata [1] to extend our previous work on contract automata to deal with real-time aspects and norms. We present *timed contract automata*, which can be seen either as contract automata [8] enriched with real-time constraints, or as timed automata [1] enriched with deontic notions. In order to evaluate the effectiveness of our approach, we present a use-case describing an airline-passenger agreement [5].

[1]Corresponding Author: Department of Computer Science, University of Malta; E-mail: stefan.chircop.15@um.edu.mt.

Related work. The starting point of our work are *contract automata* [2], an untimed operational approach to the formalisation of normative systems. Contract automata are finite state machines where states are annotated with permissions, prohibitions and obligations over single actions, and transitions are labelled with actions. The formalism allows for the encoding of reparations. In our work we take contract automata and extend it with clocks, inspired by *timed automata* [1], resulting in a combination of the two.

Different partial formalisations of normative specifications with time have been given by Governatori et al., for instance in [6,7], in the context of *defeasible logic*, classifying timed deontic actions depending on their duration and scope: *achievement*, *maintenance* and *punctual*. In our work we consider *achievement* obligations, *maintenance* permissions and treat prohibitions as *maintenance* obligations to *avoid* an event.

Finally, we refer the reader to the paper [4] for a discussion on the difficulties and challenges of defining and monitoring a formal timed normative language.

2. Timed Contract Automata

Timed contract automata regulate the behaviour of multiple parties over time. They bring together deontic notions from *contract automata* [9] and real-time notions from *timed automata* [1] in order to express how the actions performed by the different parties over time reflect their expected behaviour moving forward.

The underlying notion of time we will use throughout the paper is a continuous one, ranging over the non-negative reals $\mathbb{R}^+ \cup \{0\}$, and denoted by $\mathbb{T}$. In keeping with timed automata, we allow automata to use multiple clocks from a denumerable set $\mathbb{C}$. We will allow for the resetting of clocks, all of which will be assumed to run at the same rate. We will assume throughout the existence of a global clock (which cannot be reset) $\gamma \in \mathbb{C}$. A *clock valuation*, of type $\mathbb{C} \to \mathbb{T}$, gives a snapshot of the values carried by the clocks. We use $val_{\mathbb{C}}$ to denote the set of of all clock valuations. We will need to write conditions over the values of clocks, for which we will use *clock predicates* ranging over $val_{\mathbb{C}} \to \mathbb{B}$ and which, given a clock valuation, return whether the predicate is satisfied. We use $pred_{\mathbb{C}}$ to refer to the set of all such predicates.

We will write $v \gg \delta$ to denote the advancement of clock values in v by δ to be defined as $\lambda c \cdot v(c) + \delta$. A valuation v is said to exceed the latest satisfaction of a clock predicate τ, written $v > \max(\tau)$, if for any non-negative progress in time $\delta \in \mathbb{T}$, the predicate is not satisfied $\neg(\tau(v+\delta))$.[2] Finally, we also defined the overriding of a clock valuation by another $v \oplus v'$ to be defined to be the clock valuation which returns the value given by v' when defined, or that given by v otherwise.

The underlying deontic approach used in these automata is a party-aware and action-based one, i.e., they talk about what, for instance, a *party ought-to-do* as opposed to the *state the party ought-to-be in*. Timed contract automata will be parametrised by the set $\mathbb{P}$ of parties involved and actions $\mathbb{A}$.

In order to identify violations to permissions, we will assume that attempts to perform an action are observable. In order to do so, given a set of actions $\mathbb{A}$, we will write $\mathbb{A}_{attempted}$ to denote the enriched alphabet $\mathbb{A} \cup \{a_{attempt} \mid a \in \mathbb{A}\}$.

Our semantics will be based on an observed timed trace of actions (and attempted ones). A *timed trace* over a set of parties $\mathbb{P}$ and alphabet $\mathbb{A}$ is a finite sequence of ob-

[2] This does not take into account the possibility of clocks being reset.

served events — where an *event* is an action with associated party and timestamp (as per the global clock): $seq(\mathbb{P} \times \mathbb{A}_{attempted} \times \mathbb{T})$ such that the timestamp progresses with each observed action, i.e., given timed trace ts, then for any i and j such that $i < j$, if $ts(i) = (p_i, a_i, t_i)$ and $ts(j) = (p_j, a_j, t_j)$ then $t_i < t_j$.

In keeping with the action-based approach, norms refer to particular parties and actions. Timed contract automata allow a range of norms to be expressed, ranging over obligations, prohibitions and permissions. Real-time norms have been discussed in the literature with various useful semantics. For instance, consider granting John the permission to access a digital resource over the coming 10 minutes. Two possible semantics can be given to this permission — a one-time semantics, i.e., John can access the resource over the coming 10 minutes, but uses up that permission in doing so, or a continuous semantics, i.e., John can access that resource any number of times over the coming 10 minutes. Neither is intrinsically correct (or incorrect), since it depends on what sort of permission one intends to give John. Rather than replicate the discussion of which norms are appropriate in the real-time context, we limit ourselves to a number of norms which can, however, be extended if we want to adopt other norms as part of timed contract automata in the future.

Definition. The norms we will use are parametrised by: (i) the party on which the norm applies; (ii) temporal constraints when the norm applies; and (iii) the action on which the norm applies. For a given set of actions $\mathbb{A}$, a set of clocks $\mathbb{C}$ and parties $\mathbb{P}$, the set of possible norms, denoted by $\mathbb{D}$, covers objects of the form: $\mathscr{N}_\tau(p : a)$ where: (i) $\mathscr{N}$ is norm ranging over permission $\mathscr{P}$, prohibition $\mathscr{F}$ and obligation $\mathscr{O}$; (ii) $p \in \mathbb{P}$ is the party to whom the norm applies; (iii) $\tau \in pred_{\mathbb{C}}$ is a clock predicate indicating when the norm applies; and (iv) $a \in \mathbb{A}$ is the action which the norm refers to.

Syntax. Time contract automata can be seen as a combination of timed automata [1] in that they allow the use of real-time clocks and clocked events, and contract automata [8] in that they use states as norm-carrying modes. Similar to timed automata, (i) we allow for multiple clocks (all progressing at the same rate); (ii) transitions are guarded by clock conditions; and (iii) transitions may reset any number of clocks to particular values.

Similar to contract automata, states are associated with a number of norms. However, the temporal modality offered by the automaton can interrupt deontic norms. For instance, being in a particular state may prohibit John from reading a file as long as clock c does not exceed 10 minutes. However, a transition is taken from that state when c still reads 2 minutes, thus exiting that state. Whether the prohibition is discarded (since we are no longer in that state) or persists (since the temporal constraint has not yet run out) is a choice one has to make. On one hand, we can see the norms in the states as being *active* as long as we are in the state, or as being *enacted* when we enter the state. Both forms can be useful, and we keep both forms of *ephemeral* and *persistent* norms.

Definition. A *timed contract automaton* C, over parties $\mathbb{P}$, actions $\mathbb{A}$ and that uses clocks $\mathbb{C}$, is a tuple $\langle Q, q_0, \rightarrow, \rightarrow_{timeout}, pers, eph \rangle$ where: (i) Q is the set of states, with $q_0 \in Q$ being the initial state; (ii) $\rightarrow \subseteq Q \times (\mathbb{P} \times \mathbb{A} \times pred_{\mathbb{C}} \times val_{\mathbb{C}}) \times Q$ is the transition relation labelling each transition with a party and action which trigger it, a clock predicate which guards it, and a (possibly partial) clock valuation to reset any number of clocks upon taking the transition; (iii) $\rightarrow_{timeout} \subseteq Q \times (\mathbb{C} \times \mathbb{T} \times val_{\mathbb{C}}) \times Q$ is the timeout transition relation with resets, enabling leaving a state when a particular timer reaches a particular value and resetting any number of clocks; and (iv) $pers, eph \in Q \rightarrow 2^{\mathbb{D}}$ are functions, which given a state, return the sets of persistent and ephemeral norms active when in that

state. We will write $q \xrightarrow{p:a \mid \tau \mapsto \rho} q'$ to denote $(q,\ (p,\ a,\ \tau,\ \rho),\ q') \in \rightarrow$ and $q \xrightarrow{c=t \mapsto \rho}_{timeout} q'$ to denote $(q,\ (c,\ t,\ \rho),\ q') \in \rightarrow_{timeout}$.

A timed contract automaton is *well-formed* if (i) the global clock is never reset, i.e., if $q \xrightarrow{p:a \mid \tau \mapsto \rho} q'$, then $\gamma \notin \mathrm{dom}(\rho)$; and (ii) the automaton is *deterministic*, i.e., an observed action only allows for one transition to fire: if $q \xrightarrow{p:a \mid \tau_1 \mapsto \rho_1} q_1$ and $q \xrightarrow{p:a \mid \tau_2 \mapsto \rho_2} q_2$, then either $q_1 = q_2$ and $\rho_1 = \rho_2$, or for any clocks valuation v, $\neg(\tau_1(v) \wedge \tau_2(v))$. In the rest of the paper we will assume that timed contract automata are well-formed.

Timed Semantics. In order to define the semantics, we start by defining the configuration of a timed contract automaton. This stores all relevant information about the automaton during an execution, namely (i) current state; (ii) current value of clocks; and (iii) active persistent and ephemeral deontic norms.

Definition. A *configuration* of a timed contract automaton $M = \langle Q,\ q_0,\ \rightarrow,\ pers,\ eph \rangle$ has type: $Q \times val_{\mathbb{C}} \times \mathbb{D} \times \mathbb{D}$. We write $Conf_M$ to denote the set of all configurations, leaving out M when clear from the context. The initial configuration $conf_0$ is $(q_0, \lambda c \cdot 0, pers(q_0),\ eph(q_0))$.

Based on this, we can define the temporal progression of configurations upon observing a new event (p,a,t). Recall that the time t of the event in the trace will be according to the global clock γ. We define the configuration relation $conf \xRightarrow{p:a,\,t} conf'$ showing how a configuration evolves, breaking it down into (i) a temporal step $conf \xRightarrow[temp]{p:a,\,t} conf'$; and (ii) a deontic step $conf \xRightarrow[norm]{p:a,\,t} conf'$. Firstly, we allow progression along a matching timeout transition using the following rule:

$$\frac{q \xrightarrow{c=t' \mapsto \rho}_{timeout} q' \qquad (q',\ (v \gg \delta) \oplus \rho,\ P \cup pers(q'),\ eph(q')) \xRightarrow[temp]{p:a,\,t} C}{(q,\ v,\ P,\ E) \xRightarrow[temp]{p:a,\,t} C}\ \delta = t' - v(c),\ t' - v(\gamma) > \delta$$

Note that if a timeout transition fires before the event time, that transition is taken, and we must move to the destination state of the timeout transition, updating the persistent and ephemeral norms accordingly. If no timeout transition matches the antecedent of the rule above, we can consume the event as per the following rule:

$$\frac{q \xrightarrow{p:a \mid \tau \mapsto \rho} q'}{(q,\ v,\ P,\ E) \xRightarrow[temp]{p:a,\,t} (q',\ (v \gg \delta) \oplus \rho,\ P \cup pers(q'),\ eph(q'))}\ \delta = t - v(\gamma),\ \tau(v \gg \delta)$$

If no transition matches the rule above, we progress by remaining in the same state:

$$\frac{}{(q,\ v,\ P,\ E) \xRightarrow[temp]{p:a,\,t} (q',\ v \gg \delta,\ P,\ E)}\ \delta = t - v(\gamma)$$

Deontic Semantics. We can now turn to the deontic aspect of the semantics of timed contract automata. The semantics of the individual norms is characterised using a satisfaction and a violation predicate which decides how an observed action interacts with that norm, allowing to extend the progress relation to address configuration changes from a deontic both in the case of a violation or otherwise.

$$vio(\mathscr{P}_\tau(p:a),\ (p:a_{attempt},\ v)) \stackrel{df}{=} \tau(v) \qquad sat(\mathscr{P}_\tau(p:a),\ (p':a',\ v)) \stackrel{df}{=} v > \max(\tau)$$
$$vio(\mathscr{F}_\tau(p:a),\ (p:a,\ v)) \stackrel{df}{=} \tau(v) \qquad sat(\mathscr{F}_\tau(p:a),\ (p':a',\ v)) \stackrel{df}{=} v > \max(\tau)$$
$$vio(\mathscr{O}_\tau(p:a),\ (p':a',\ v)) \stackrel{df}{=} v > \max(\tau) \qquad sat(\mathscr{O}_\tau(p:a),\ (p:a,\ v)) \stackrel{df}{=} \tau(v)$$

$$\frac{\exists n \in P \cup E \cdot vio(n,\ (p : a, v \gg \delta))}{(q,\ v,\ P,\ E) \xrightarrow[norm]{p:a,\ t} \bot}\ \delta = t - v(\gamma)$$

$$\frac{\neg\exists n \in P \cup E \cdot vio(n,\ (p : a, v \gg \delta))}{(q,\ v,\ P,\ E) \xrightarrow[norm]{p:a,\ t} (q,\ v,\ active(P,\ (p : a,\ v)),\ active(E,\ (p : a,\ v)))}\ \delta = t - v(\gamma)$$

Note that *active* removes satisfied norms given an observed event, i.e., $active(N,\ (p : a,\ v))$ is defined to be $\{n \in N \mid \neg sat(n,\ (p : a,\ v))\}$. In addition, we will have rules to ensure that a violation $\bot$ will not evolve further, i.e., $\bot \xrightarrow[temp]{p:a,\ t} \bot$ and $\bot \xrightarrow[norm]{p:a,\ t} \bot$.

Combining Temporal and Deontic Semantics. We can combine these relations by putting them in sequence, i.e., $c \overset{e}{\Rightarrow} c'$ is defined to mean that there exists configuration c'' such that $c \xrightarrow[norm]{e} c'' \xrightarrow[temp]{e} c'$. The residual configuration after a well-formed timed trace can be computed using the transitive closure of this combined relation, starting from the initial configuration $conf_0$. A timed trace *ts violates* the timed contract automaton if and only if $conf_0 \overset{ts}{\Rightarrow} \bot$.

3. Use Case: Airport Regulations

We consider a use case from the literature expressing airport regulations, and based on the Madrid Barajas airport regulations [5]. Due to space restrictions, we only present a selection of the regulations, as shown below. The parties involved are (i) the passenger p; and (ii) the airline company ac.

1. The passenger is permitted to *check in* (*ci*) 2 hours before take-off. However, the check in desk is closed half an hour before take-off, and the passenger is prohibited from checking in from that point onwards.
2. The passenger is then obliged to *present their boarding pass* (*bp*) within 5 minutes, after which they have another 5 minutes to *produce their passport* (*ppt*).
3. Having done so, the passenger is permitted 10 minutes to *dispose of any liquids in their hand luggage* (*dlhl*), and *present it to the staff* (*prs*). The passenger is also prohibited from *carrying any weapons* (*wps*).
4. In the meantime, should the airline company find reason to *stop the passenger* (*stop*), then they must put their *hand luggage in the hold* (*hold*) within 20 minutes, as well as *call security* (*sec*) within 1 minute.
5. Should the staff *find no issues* (*clear*), then the passenger is permitted to *board the plane* (*board*) within 90 minutes since producing the passport.

We can express this snippet of the regulations using the timed contract automaton shown in Fig. 1. Note that we label transitions as $p : a \mid \tau \mapsto \rho$ to denote the transition tagged by party p, action a, clock constraints τ and resets ρ. Also note that we write $\top$ for the clock constraint which always returns true, and we express resets as assignments. Ephemeral and persistent norms are tagged individually for clarity.

Note that the automaton uses much of the structure of the original text. On the other hand, it provides a more operational view of the agreement, and is more amenable to automated analysis.

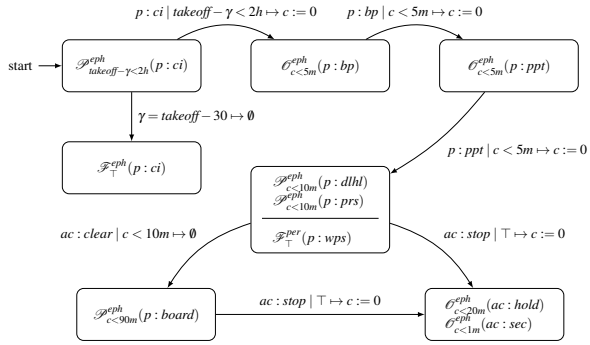


Figure 1. Automaton for the airport regulation use case.

4. Conclusions

In this paper we have presented timed contract automata, combining contract automata with timed automata to enable the operational modelling of real-time normative agreements. We do not envisage such automata as the specification language in which agreements can be modelled. Logic-based deontic approaches are more effective in that they provide better structure. Instead, we see timed contract automata as the operational model in which one can reason more effectively about real-time agreements. We are currently looking at formally correct compilation from deontic logics into timed contract automata, and algorithms for efficient analysis of timed contract automata. We already inherit many decidability (and non-decidability) results from timed automata, and the interesting question is how far we can push analysis such as conflict analysis and model checking of timed contract automata, and their use in runtime verification.

References

[1] Rajeev Alur and David L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[2] Shaun Azzopardi, Gordon J. Pace, and Fernando Schapachnik. Contract automata with reparations. In *JURIX'14*, pages 49–54. IOS Press, 2014.

[3] Shaun Azzopardi, Gordon J. Pace, Fernando Schapachnik, and Gerardo Schneider. Contract automata - an operational view of contracts between interactive parties. *Artif. Intell. Law*, 24(3):203–243, 2016.

[4] Shaun Azzopardi, Gordon J. Pace, Fernando Schapachnik, and Gerardo Schneider. On the specification and monitoring of timed normative systems. In *RV'21*, volume 12974 of *LNCS*. Springer, 2021.

[5] Alberto García, María-Emilia Cambronero, Christian Colombo, Luis Llana, and Gordon J. Pace. Themulus: A timed contract-calculus. In *MODELSWARD'20*, pages 193–204. SciTePress, 2020.

[6] Guido Governatori, Joris Hulstijn, Régis Riveret, and Antonino Rotolo. Characterising deadlines in temporal modal defeasible logic. In *AI'07*, pages 486–496, 2007.

[7] Guido Governatori and Antonino Rotolo. Justice delayed is justice denied: Logics for a temporal account of reparations and legal compliance. In *CLIMA XII*, pages 364–382, 2011.

[8] Gordon J. Pace and Fernando Schapachnik. Contracts for Interacting Two-Party Systems. In *FLACOS'12*, volume 94 of *ENTCS*, 2012.

[9] Gordon J. Pace and Fernando Schapachnik. Types of rights in two-party systems: A formal analysis. In *JURIX'12*, pages 105–114, 2012.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220461

Recognising Legal Characteristics of the Judgments of the European Court of Justice: Difficult but Not Impossible

Alessandro CONTINI [a] Sebastiano PICCOLO [a,1], Lucia LOPEZ ZURITA [a], and Urska SADL [a]

[a] *Copenhagen University, Faculty of Law, iCourts*

ORCiD ID: Alessandro Contini https://orcid.org/0000-0002-8999-2610, Sebastiano Piccolo https://orcid.org/0000-0002-6986-3344, Lucia Lopez Zurita https://orcid.org/0000-0001-6092-0114, Urska Sadl https://orcid.org/0000-0003-4635-3816

Abstract. Machine learning has improved significantly during the past decades. Computers perform remarkably in formerly difficult tasks. This article reports the preliminary results on the prediction of two characteristics of judgments of the European Court of Justice, which require the knowledge of concepts and doctrines of European Union law and judicial decision-making: The legal importance (doctrinal outcome) and leeway to the national courts and legislators (deference). The analysis relies on 1704 manually labelled judgments and trains a set of classifiers based on word embedding, LSTM, and convolutional neural networks. While all classifiers exceed simple baselines, the overall performance is weak. This suggests first, that the models learn meaningful representations of the judgments. Second, machine learning encounters significant challenges in the legal domain. These arise doe to the small training data, significant class imbalance, and the characteristics of the variables requiring external knowledge.

The article also outlines directions for future research.

Keywords. Classification, European Court of Justice, Word embedding, LSTM, CNN

1. Introduction

Deep neural networks (DNN) and computer hardware have expanded the range of tasks in which machines outperform humans [1]. Examples include the remarkable progress in computer vision [2], natural language processing [3] and gaming [4]. Artificial intelligence has also transformed the legal profession, providing sophisticated tools for implementing computational legal reasoning, thus enabling argument extraction from legal texts [5,6]. That said, the legal analysis of rights, duties, precedent, and legal development (legal doctrine) has seemingly remained a safe space of a trained human lawyer. A deep-seated opinion is that law is a distinctively human domain, involving deep under-

[1] Corresponding Author: Sebastiano Piccolo, jvt612@ku.dk.

standing of legal sources, situation sense, the ability to read legal texts between the lines, constructing the systems of knowledge [7]. In sum, machines can not (for now) conclusively answer questions about the content of the law. This raises the question exactly how much machine learning can contribute to the analysis of judicial decisions.

The article develops classifiers based on word embedding, LSTM, and CNN (convolutional neural network), which consider the text of a judgment to *1)* Predict the legal importance of the Court's judgment (whether the Court makes a strong contribution to the legal doctrine, such as creating new concepts or principles); and *2)* Detect whether the Court gives the national judge or legislator a leeway to adopt the final decision (defers the final decision about the law to the national court or the legislator). The latter aspect is particular to European Union law, calling for the division of labor between the Court and the national courts. The article trains the classifier on the full judgments and on single paragraphs of the judgments, aggregating the single scores to obtain predictions on the judgment level.

The findings confirm the expectation that predicting legal importance is harder than detecting deference. Concretely, predicting single paragraphs and aggregating their scores is sub-optimal (around 25% lower than the performance of a classifier trained on the full judgment). Moreover, classifiers based on LSTM perform better than those based on CNN. The best score on predicting deference is F1=0.463, while the best score on predicting doctrinal outcome is F1=0.376. The findings echoes the observation from Habernal et al. [6] that legal experts rely on the context beyond the single paragraph used as input for their algorithm to label the arguments. A number of factors contribute to the weak performance of the algorithms: a relatively small training set, the high class imbalance, and the fact that the selected variables require extensive knowledge of complex legal concepts and legal doctrine. Given that all factors are intrinsic to the legal domain, future research should focus on developing more sophisticated models.

2. Data and Methods

2.1. Dataset

The dataset includes 1704 Judgements issued by the Court of Justice of the European Union between 1954 and 2020. All judgements are in English and freely available from the official portal of the European Union Eur-lex. Content-wise, the judgments concern the free movement of goods and the free movement of persons, both an ideal test bed. On the one hand, the Court has fashioned the fundamental principles of European Union law and developed its central doctrines in those areas, which makes them ideal for the prediction of legal importance (doctrinal outcome). On the other hand, with the development of fundamental principles, it became relevant whether the Court left the key practical decisions to the courts and the legislators of the Member States (deference). Legal experts (human coders) labelling each judgment specified whether the judgment was legally important (Doctrinal Outcome or DOCOUT) and whether the Court deferred the final decision to the national courts or legislators (Deference or DEF).

The data is divided in two datasets: the first contains the full judgements and their relative predicted label. The second contains single paragraphs of the judgments, each with assigned label predicted for their judgement. Compared to similar researches our

dataset is small: Wei et al. [8] sampled from a dataset composed by millions of judgments; Xiao et al. [9], in 2018, used a dataset composed by 2.6 million criminal cases published by the Supreme People's Court of China. Small training data is commonly known to result in poor performance, particularly in Deep Learning[10]. However, it is time expensive to produce hand coded training sets.

2.2. Variables

Doctrinal Outcome relates to the Court's law-making activity in the narrow / legal doctrinal sense. Doctrine is defined as a set of rules and principles, which determine the scope and the content of rights and duties. The coding relies on the opinion of legal experts and lawyers. There are two possible outcomes: weak (=0) and strong (=1). The Court can entrench, strengthen or expand its doctrines, create new concepts or develop principles (DOCOUT=1). By contrast, it can moderate its strong doctrinal positions or restate and apply established doctrines, concepts and principles, without further extending their scope (DOCOUT=0).

Deference indicates whether the Court defers the final decision to the national court or the legislator; that is, whether it gives the national judge leeway as per the final decision/outcome of the case. The following language is indicative of the existence of deference (DEF=1): 'it is for the referring court to decide / establish / determine / examine', 'the national court must provide or decide'. When there are no references to the national courts, the outcome is DEF=0. Both variables present a high class imbalance.

2.3. Models

The article implements models based on two types of neural networks: Convolutional Neural Network (CNN) and Bidirectional Long short-term memory Recurrent Neural Network (LSTM), as they have been shown to be excellent methods for text classification [11]. The models' structure is organised in layers: The first is a Text Vectorisation layer followed by an Embedding layer. The Embedding layer will learn a vectorial space where similar words, or words that appear in similar contexts, are at a closer distance than words that appear in different contexts. The second is a Bidirectional LSTM layer. This layer *reads* the text sequentially and is therefore able to detect sequential dependencies as well as *remember* past information and context. Finally, there is a variable number of Dense layers, using ReLu activation function and a final output layer implementing a Sigmoid activation function. Furthermore, each hidden layer implements dropout as a means of regularization to reduce chances of over-fitting. Dropout is more effective than other standard computationally expensive regularizers [12]. The models based on CNN follow the same structure, with the difference that a CNN substitutes the bidirectional LSTM, and the CNN layer is followed by a max pooling layer. The number of hidden Dense layers and their dropout rate, as well as the size of the embedding, the number of neurons, the learning rate and the weight of every positive example are computed through the ParEGO hyperparameter tuning algorithm [13]. Table 1 reports the values.

Before fitting the model, we converted the text into lowercase and removed punctuation and numbers. We decided not to remove stopwords, as that did not help to improve the performance of our models. We restricted the size of the vocabulary for the word embedding to 30000 words. The datasets were randomly divided in a training set

Table 1. Parameters found by ParEGO for each variable to predict, dataset (whether we use the full judgments or the paragraphs), and network type. Values are rounded to the third most significant digit.

Variable	DOCOUT				DEF			
Dataset	Full Judgments		Paragraphs		Full Judgments		Paragraphs	
Net Type	LSTM	CNN	LSTM	CNN	LSTM	CNN	LSTM	CNN
Emb. dimension	36	136	204	197	30	113	38	183
Num. units	41	42	167	47	26	31	31	49
Learning rate	1.89e-4	3.9e-5	2.5e-5	4.44e-4	1.54e-4	3.9e-5	9e-3	1.47e-3
Layers Num	0	1	3	1	0	1	1	1
Pos. weight	2.775	1.13	5.02	1.21	1.09	1.129	2.662	1.165
Dropout Conv	-	0.133	-	0.046	-	0.202	-	0.208
Dropout Dense	0.079	0.203	0.2	0.226	0.14	0.219	0.0293	0.319
Conv. Kernel	-	11	-	8	-	8	-	6

and a test set with 85% and 15% of the data. The training set was furthermore randomly split into 4 cross validation folds for the model training. Finally, the performance of the models are computed on the test set. We trained our models until convergence, using an early stopping criterion that monitored the F1 score, using the Adam optimization algorithm and Binary Cross Entropy as loss functions. This is a good strategy which prevents over-fitting and saves some computational time.

3. Results and Discussion

The evaluation of the performance of the models uses the F1 score [14] and the ROC-AUC, as the accuracy is inappropriate in presence of high class imbalance [15]. In fact, a dummy classifier – that is a classifier that classifies everything as the majority class – would obtain an extremely high accuracy score because of such imbalance. The F1 and ROC are reported in Table 2 for both predicted variables, the type of model considered, and the type of prediction – i.e. classifying the full judgment or classifying single paragraphs. The comparison between the results on cross validation and the test set suggest no over-fitting of the data. However, in the case of the LSTM on full judgments, for both DOCOUT and DEF, the performances on the test set are higher than on the cross validation. This might indicate some under-fitting. The overall performance of the models is weak due to a number of difficulties: small training dataset, substantial class imbalance, and the fact that predicting doctrinal outcome is likely to require extensive knowledge of complex legal concepts and legal doctrine. The findings are nonetheless important and telling.

First, LSTM appears to perform better than CNN on all tasks. Future research could push the results forward by increasing the size of the training data (which implies time consuming and expensive hand coding of the data). The performance of deep learning models is known to increase with the size of the training data [10]. Alternatively, researchers could explore more complex models: from multiple LSTMs in sequence, to encoder-decoder architectures, and more recent BERT-based models[16].

Second, predicting deference is easier than predicting doctrinal outcome. This is expected, as deference has a lower class imbalance than doctrinal outcome. In fact, for the LSTM on the full judgments, the positive weight for deference selected by ParEGO

Table 2. Performance of our models on classifying full judgments or single paragraphs for doctrinal outcome (DOCOUT) and deference (DEF)

			Cross validation		Test-set	
Label	Dataset	Net Type	F1	ROC	F1	ROC
DOCOUT	Full Judgments	LSTM	0.317	0.682	0.376	0.649
		CNN	0.311	0.663	0.316	0.66
	Paragraphs	LSTM	0.570	0.870	0.569	0.884
		CNN	0.550	0.849	0.539	0.853
DEF	Full Judgments	LSTM	0.372	0.715	0.463	0.639
		CNN	0.380	0.700	0.342	0.677
	Paragraphs	LSTM	0.574	0.801	0.605	0.840
		CNN	0.589	0.837	0.605	0.863

(Table 1) is 1.09, as opposed to 2.775 for the doctrinal outcome. Additionally, from the legal perspective, it is more difficult to identify a strong or weak doctrinal outcome than a deferential outcome. The former is often implicit in the text, and often a matter of scholarly analysis rather than an information contained in the text of the judgment. [17]. The latter relies more on the text and the presence of certain expressions.

Finally, predictions on the paragraph level exhibit higher performance than prediction on the whole judgment. However, the strategy of aggregating paragraph predictions onto full judgment predictions yields performance 25% worse than those obtained through direct classification of full judgments. As such, in order to classify legal texts, we need to cope with long sequences. Besides the already mentioned encoder-decoder architectures, other ideas worth of further investigation are *1)* feeding multiple paragraphs in parallel to the prediction algorithm, thus training a network with multiple inputs, and *2)* summarising/filtering the judgments to retain only the most salient parts of the text.

4. Conclusions

The article investigated how much machine learning could contribute to the legal analysis of judicial decisions by predicting two legally interesting and demanding characteristics: legal importance (doctrinal outcome) and deference. It trained a set of classifiers based on word embedding, LSTM, and CNN on a dataset of manually labelled judgments of the European Court of Justice. The tasks proved difficult and performance were weak, with LSTM performing better than CNN.

Further analysis and experimentation would be required to understand the significance of these results. These include: developing more sophisticated models, incorporating more hand-coded judgements, and finding ways to deal with long text sequences. This work can be viewed as a starting point for studying the impact of text classification and the potential of deep learning models in very specific NLP fields, such as the legal domain. At the same time, the article suggests that the legal experts remain the final authority when it comes to legal doctrine.

References

[1] E. Brynjolfsson and T. Mitchell, "What can machine learning do? workforce implications," *Science*, vol. 358, no. 6370, pp. 1530–1534, 2017.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[3] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

[4] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[5] K. D. Ashley, *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge University Press, 2017.

[6] I. Habernal, D. Faber, N. Recchia, S. Bretthauer, I. Gurevych, C. Burchard, *et al.*, "Mining legal arguments in court decisions," *arXiv preprint arXiv:2208.06178*, 2022.

[7] F. Schauer, *Thinking like a lawyer: a new introduction to legal reasoning*. Harvard University Press, 2009.

[8] F. Wei, H. Qin, S. Ye, and H. Zhao, "Empirical study of deep learning for text classification in legal document review," *2018 IEEE International Conference on Big Data (Big Data)*, pp. 3317–3320, 2018.

[9] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang, and J. Xu, "Cail2018: A large-scale legal dataset for judgment prediction," 2018.

[10] M. Z. Alom, T. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. Nasrin, M. Hasan, B. Essen, A. Awwal, and V. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, p. 292, 03 2019.

[11] M. S, J. D, and D. M, "Evaluation of impact of neural networks in text classification," *Journal of University of Shanghai for Science and Technology*, vol. 23, pp. 1279–1292, 07 2021.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[13] J. Knowles, "Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

[14] L. Derczynski, "Complementarity, F-score, and NLP evaluation," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, (Portorož, Slovenia), pp. 261–266, European Language Resources Association (ELRA), May 2016.

[15] K. Spackman, ". signal detection theory: Valuable tools for evaluating inductive learning," *Proc. Sixth Internat. Workshop on Machine Learning. Morgan Kaufman, San Mateo, CA, pp. 160–163*, 1989.

[16] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Legal-bert: The muppets straight out of law school," *arXiv preprint arXiv:2010.02559*, 2020.

[17] U. Sadl and Y. Panagis, "What is a leading case in eu law? an empirical analysis," *European Law Review*, vol. 40, pp. 15–34, Feb. 2015.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220462

Automating the Response to GDPR's Right of Access

Beatriz ESTEVES [a,1], Víctor RODRÍGUEZ-DONCEL [a], Ricardo LONGARES [a]

[a] *Ontology Engineering Group, Universidad Politécnica de Madrid, Spain*

Abstract.
With the enforcement of the European Union's General Data Protection Regulation, users of Web services – the 'data subjects' –, which are powered by the intensive usage of personal data, have seen their rights be incremented, and the same can be said about the obligations imposed on the 'data controllers' responsible for these services. In particular, the 'Right of Access', which gives users the option to obtain a copy of their personal data as well as relevant details such as the categories of personal data being processed or the purposes and duration of said processing, is putting increasing pressure on controllers as their execution often requires a manual response effort, and the wait time is negatively affecting the data subjects. In this context, the main goal of this work is the development of an API, which builds on the previously mentioned structured information, to assist controllers in the automation of replies to right of access requests. The implemented API method is then used in the implementation of a Solid application whose main goal is to assist users in exercising their right of access to data stored in Solid Pods.

Keywords. digital rights management, GDPR, right of access, Solid

1. Introduction

With the enforcement of the European Union's General Data Protection Regulation (GDPR), users of Web services have seen their rights as GDPR 'data subjects' being expanded when it comes to the processing of their personal data. On the other hand, on top of other GDPR-related obligations, 'data controllers', the entities that effectively process the data, have seen an increase in workload related to the response to data subject's right-related requests. GDPR's Chapter III[2] details a set of 10 data subject rights, starting with the 'Right to be Informed' described in Articles 13 and 14 and ending with the 'Right to object to automated decision making' in Article 22. Considering this, data controllers would benefit from having the information they need to provide to data subjects in a structured format to automate the response to such requests [1]. In particular, the 'Right of

[1]Corresponding author: beatriz.gesteves@upm.es
[2]https://gdpr-info.eu/chapter-3/

Access'[3] is putting more and more pressure on controllers as they not only have to provide the purpose for which the data is being used or the types of data being processed but also need to provide a copy of said data. As this task is usually done manually, the wait time can negatively affect data subjects.

In addition, with the emergence of decentralised data storage solutions, such as Solid[4], as an alternative to the traditional centralised data silos, new challenges appear as the data subject–data controller roles are still not adequately defined in this decentralised contexts. In this context, the creation of Application Programming Interface (API) services would help with the automation of right-related requests as, at their core, they are a 'request–response' type of software interface and consequently can be used in different types of software.

Taking this into consideration, this contribution will focus on the following research objectives:

RO1. Implementing an API method that automates the reply to an access right request, making use of RDF information.
RO2. Developing a Solid application which uses the implemented method to assist users in exercising their right of access in a decentralised storage environment, such as data stored in Solid Pods.

This paper is organized as follows: Section 2 introduces the GDPR's Data Subject Access Right and its core requirements, Section 3 discusses related work in this area, Section 4 describes the implementation of an API method that automates the response to an access right request and a Solid application that uses said API method to assist users in exercising their right of access to data stored in Solid Pods and Section 5 presents conclusions and future lines of work. Online supplementary material is provided at https://protect.oeg.fi.upm.es/access-right/, including a demonstration of the implemented Solid application.

2. The Right of Access

In an attempt to provide users with more transparency on how their personal data is being processed, GDPR's right of access puts emphasis not only on the "right to obtain a copy" of said data but also an additional set of information pieces needs to be provided to the subject related with the context in which the processing is made. Therefore, the right of access's main goal is to give people sufficient and clear information about how their personal data is processed so that subjects know that their data is being handled according to their expectations and its quality is being verified – this will then facilitate the exercise of other rights, such as the right to be forgotten or to rectification. Data subjects do not have to provide a justification to request access to the data nor do they need to pay a fee to exercise such right unless further copies of the same data are requested. Unless it is obvious that the request is being made in violation of other regulations, the data controller has the duty to reply to the data subject's request. Moreover, in addition to confirming whether or not they are processing personal data and

[3] https://gdpr-info.eu/art-15-gdpr/
[4] https://solidproject.org/

providing a copy of said data, controllers have a duty to provide information regarding the purposes of the processing, the categories of personal data being processed, the recipients or categories of recipients of said data, the duration of the processing, the existence of other data subject rights, including the right to lodge a complaint with a supervisory authority, the source of data if not collected directly from the data subject and the existence of automated decision-making. In addition to the requirements specified in GDPR, on January 18th 2022, the European Data Protection Board (EDPB) issued a set of guidelines specifically on the right of access [2].

3. Related Work

There are a few API implementations specifically targeting the response to data subject right-related requests, however, they only focus on providing a copy of the personal data and leave out all the other requirements specified by the GDPR.

Microsoft Graph [3] is a platform developed to access Microsoft 365 data. In particular, the Microsoft Graph compliance and privacy APIs [4] were developed to create and manage data subject access requests and to help developers and enterprises to easily identify data subjects and find their personal information. Although the process through which the data is obtained is automated, several components require manual intervention, such as confirmation as to whether the data is being processed.

Oracle's Data Privacy API focus on providing a solution for enterprises that use Oracle databases to store personal data [5]. Currently, it supports the implementation of two types of requests, the already discussed Right of Access and also the Right to be forgotten – a right that can be exercised by a data subject when they want the system in question to erase all data related to them.

AppsFlyer [6] goes one step further by providing an API that not only supports access and erasure requests but also deals with the Right to Data Portability – transfer data from one controller to another – and the Right to Rectification – correct inaccurate data. The request flow in AppsFlyer also involves manual intervention, as when a data subject submits a request, the app owner has to forward the request to AppsFlyer.

Through the analysis of these solutions, it is possible to conclude that there is a gap related to the implementation of an API service that fulfils all the requirements of a GDPR Right of Access request since all solutions only focus on providing a copy of the data and don't provide detailed information regarding purposes for processing, duration of the processing and so on.

4. Implementation

4.1. Research Methodology

The used methodology approach encompassed the following steps:

1. An evaluation of current gaps on the right of access APIs was performed.

2. Similar regulation from other jurisdictions was reviewed in order to understand if new requirements needed to be added into consideration.
3. Semantic Web vocabularies were used to tag the data in terms of the personal data they contain, to specify the policies that determine the access to said data and to store the consent record of an authorized access request.
4. The API method and documentation were developed.
5. Solid's personal data storage ecosystem was then chosen to verify the applicability of the API method as it is based on Web standards.

Further information on steps 1 to 3 is provided at https://protect.oeg.fi.upm.es/access-right/.

4.2. API development

The main technologies used to implement the API were the expressjs[5] and swagger-ui-express[6] libraries, used to develop the API and create its respective documentation. Inrupt's JavaScript client libraries[7] were also used to authenticate the user and to handle data stored in a Solid Pod.

Initially, the developed API only had one parameter which was related to the identity of the user as this is the only requirement described by the GDPR for a data subject to be allowed to exercise their right of access. However, since the data subject may, for example, be interested in accessing only certain categories of data or only accessing data used for a certain purpose, data categories and purposes were added as request parameters to allow the data subject to have a more fine-grained access right.

The main function of the API is to obtain the data stored in the Solid Pod and send it to the user as a JSON file with two components – a boolean variable that will be true in case personal data that matches the request is found in the Pod and a JSON object that contains the respective list of found resources. To enable access to the data, the user must be logged into their Pod. As previously stated, the authentication protocol is implemented using Inrupt's libraries – a session is generated and stored so that information regarding the identity of the user (present in a WebID profile document in the case of Solid) can be passed to the API request.

Once the user is logged in, the API can process an access request. Initially, the URI of all resources stored in the Pod is collected and matched with the request. If the request is made without specifying any further parameters, then all the resources present in the Pod will be returned independently of the categories of personal data that they include or the purpose for which its processing is allowed. If a specific set of personal data categories is specified along with the access request then only those categories will be returned. However, it must be noted that for this feature to work, the resources in the Pod need to include a RDF statement, using for instance the Extended Personal Data concepts for DPV[8], to specify which type of data they contain. Finally, in case the user only wants to

[5] https://expressjs.com/es/
[6] https://www.npmjs.com/package/swagger-ui-express
[7] https://docs.inrupt.com/developer-tools/javascript/client-libraries/
[8] https://w3id.org/dpv/dpv-pd

access data that is being used for a particular purpose, access control policies that define the purpose for processing the stored resources need to be defined and kept in their Pod. For the purposes of this work, we assume that users are using the Open Digital Rights Language (ODRL) Profile for Access Control (OAC) [7] to create policies to govern the access to their Solid-stored data[9]. Using these policies the API method matches the request purpose with the stored policies' purposes and if there is a match then the corresponding data is returned. In addition to the policies, we assume that consent records, corresponding to granted data access requests, are kept in the Pod. The generation and modelling of consent records are based on previous work [8]. These records are then used by the API to retrieve the entities that accessed the data.

A public repository with the developed code is accessible at https://github.com/besteves4/access-right-api.

4.3. Exercising the Right of Access to Solid Pod data

As previously stated, Solid is a protocol focused on providing its users with decentralised personal data storage. Currently, access control to Solid-stored resources is specified using the Web Access Control specification which uses Access Control Lists (ACLs)[10] to define which agents have access to Solid resources. However, these ACL authorisations don't allow the specification of purposes for the access to the resources as well as not permitting the definition of specific access to particular types of personal data. In this context, as specified in the previous section, this work makes use of OAC policies to overcome this issue. OAC[11] uses the Data Privacy Vocabulary (DPV), which provides taxonomies for the specification of relevant privacy and data protection information, and ODRL, which allows for the expression of rich policies over digital assets.

As the users need to be able to select specific types of personal data and/or specific purposes for the processing of said data to have a more fine-grained right of access, the developed Solid application includes two drop-down trees that use DPV's personal data categories and purposes taxonomies to populate its structure. The selected categories are then used to feed the API request call. For each returned resource, the URI is provided, as well as the category of personal data included in the file, the agents that accessed the data and a list of the policies governing the access to said resource. A download button is also available so that the user can obtain a copy of the resource data. A demonstration of the developed Solid application is available at https://protect.oeg.fi.upm.es/access-right/ and the public repository of the code is accessible at https://github.com/besteves4/access-right-solid for further development.

This solution provides an advance in relation to the state-of-the-art reviewed solutions as it provides granular information on the personal data categories contained in the resources, as well as the purpose for which it was/can be used, in addition to the provision of a copy of the data.

[9]The SOPE application available at https://github.com/besteves4/solid-sope can be used to automatically generate these policies without having knowledge on ODRL and its semantics.

[10]http://www.w3.org/ns/auth/acl#

[11]https://w3id.org/oac

5. Conclusions

This work explored the implementation of an API service that can be used for the automation of GDPR's Right of Access. Current state-of-the-art solutions developed to assist data subjects and data controllers in the exercising and resolution of data subject right-related requests focus on providing users with the 'right to obtain a copy' of the data but do not fulfil all the requirements set by the GDPR. Furthermore, these APIs are currently not equipped to deal with requests regarding data stored in decentralised systems such as Solid Pods.

Therefore, the main contribution of this work relies on the development of an open-source API that can be used in the context of decentralised systems to provide data subjects with a 'fine-grained' right of access where it can be explicitly checked which data is being used for what purpose in addition to obtaining a copy of the data.

In future lines of work, the API can be improved to provide a more complete answer to a Right of Access request – information about other data subject rights should be provided, as well as clear information regarding recipients of the data, including identity and contact information. Furthermore, audit logs of the process of exercising the right should be kept in a dedicated container in the Pod, for future inspection. Moreover, there are factors that have not been considered for the sake of practicality. For example, we have assumed that during the period of time in which the resources are stored in the Pod under the effects of a policy, they can be automatically provided through the API. This feature can also be extended to deal with new policy constraints, such as a limited time duration for the storage or a periodicity constraint. Also, new parameters for filtering the requested data could be added to the API and to the Solid application.

Funding Acknowledgments This research has been supported by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813497 (PROTECT).

References

[1] Esteves B, Rodríguez-Doncel V. Analysis of Ontologies and Policy Languages to Represent Information Flows in GDPR. Semantic Web Journal. 2022.
[2] EDPB. Guidelines 01/2022 on data subject rights - Right of access - Version 1.0; 2022. Available from: https://bit.ly/3sgtnSd.
[3] Overview of Microsoft Graph; 2022. Available from: https://bit.ly/3DhPnCp.
[4] Use the Microsoft Graph compliance and privacy APIs; 2022. Available from: https://bit.ly/3MORtwZ.
[5] Appendix B: Using the Data Privacy API; 2021. Available from: https://bit.ly/3MPJFLn.
[6] Implementing the OpenGDPR API; 2022. Available from: https://bit.ly/3eSFphG.
[7] Esteves B, Pandit HJ, Rodríguez-Doncel V. ODRL Profile for Expressing Consent through Granular Access Control Policies in Solid. In: 2021 IEEE EuroS&PW; 2021. p. 298-306.
[8] Esteves B, Rodríguez-Doncel V, Pandit HJ, Mondada N, McBennett P. Using the ODRL Profile for Access Control for Solid Pod Resource Governance. In: Groth P, Rula A, Schneider J, Tiddi I, Simperl E, Alexopoulos P, et al., editors. The Semantic Web: ESWC 2022 Satellite Events; 2022. p. 16-20.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220463

A Compression and Simulation-Based Approach to Fraud Discovery

Peter FRATRIČ [a,1], Giovanni SILENO [a], Tom VAN ENGERS [a,b] and Sander KLOUS [a]
[a] *Informatics Institute, University of Amsterdam, the Netherlands*
[b] *Leibniz Institute, TNO/University of Amsterdam, the Netherlands*

Abstract. With the uptake of digital services in public and private sectors, the formalization of laws is attracting increasing attention. Yet, non-compliant fraudulent behaviours (money laundering, tax evasion, etc.)—practical realizations of violations of law—remain very difficult to formalize, as one does not know the exact formal rules that define such violations. The present work introduces a methodological framework aiming to discover non-compliance through compressed representations of behaviour, considering a fraudulent agent that explores via simulation the space of possible non-compliant behaviours in a given social domain. The framework is founded on a combination of utility maximization and active learning. We illustrate its application on a simple social domain. The results are promising, and seemingly reduce the gap on fundamental questions in AI and Law, although this comes at the cost of developing complex models of the simulation environment, and sophisticated reasoning models of the fraudulent agent.

Keywords. fraud discovery, non-compliance detection, active learning, agent-based modelling, simulation, behavioural exploration

1. Introduction

Formalizing legislation into machine-readable artefacts is a traditional track of research in law and computer science [2]. However, the discussion on how representing and processing normative directives generally obfuscates a more fundamental problem of normative reasoning. In law, we often encounter rules such as: *Any person who willfully attempts in any manner to evade or defeat any tax imposed by this title or the payment thereof shall, in addition to other penalties provided by law, be guilty of a felony.* This rule, defining tax evasion in the United States, does not say anything about what types of behaviour can be deemed to be attempts to evade taxes. Yet, it implicitly assumes that any felony will consist of a sequence of actions, and refers (without defining it concretely) to some set of action sequences that are relevant to qualify or disqualify a certain behaviour as tax evasion. In this paper, we will focus on the problem of addressing these rules with implicit behavioural definitions (*implicit rules* for short), in particular concerning qualifications of *non-compliant* behaviour. This problem partially overlaps with the traditional *case-based reasoning* research track in AI & Law [1], aiming to reconstruct the structure

[1]Corresponding Author: Peter Fratrič, p.fratric@uva.nl. This work was partly funded by the Dutch Research Council (NWO) for the HUMAINER AI project (KIVI.2019.006).

of rationale behind case decisions, typically identifying relevant factors and their relative contributions to the conclusion; however, the "behavioural definition" issue studied here focuses primarily on capturing legally relevant behavioural *scripts*, rather than relevant contextual factors: the temporal sequence of actions will play the major role.

Two general modelling approaches can be identified (see eg. [9]): *rule based*, in which an expert identifies a set of rules that indicate evidence likely to be related to non-compliant activity (see eg. [8]); and *machine-learning based*, where a dataset of evidence related to usually both compliant or non-compliant activity is used to train a non-compliance classifier over the entire behavioural space [4]. Unfortunately, sample datasets of fraudulent behaviour suffer from class imbalance; there is only a relatively small amount of labeled instances of non-compliance compared to labeled instances of compliance. To face this issue, recent contributions have proposed to use either *inductive* (using both labeled and unlabeled instances in the training process) [3] or *transductive* (building upon local similarities among data points) [6] *semi-supervised learning* for non-compliance classification.

We propose a research direction that aims to combine recent trends into one simulation framework, where instances of non-compliance can be generated [5]. We will posit and elaborate on the following arguments: (a) the definition of what is non-compliance can be seen as a *compression* task on possible, relevant behaviours; (b) *tracking* of (intentional) non-compliance can be based on defining sound constraints and preferences (eg. expressed as pay-offs); (c) automated exploration of the behavioural space can be performed by means of *simulation*. We also observe that, despite the help of computational tools, (d) the role of human experts in directing the search and determining the legal status of the generated action sequence remains crucial.

The paper is organized as follows. Section 2 presents relevant concepts and the proposed method: the task of constructing implicit definitions of non-compliant behaviour as a combination of utility maximization and an active learning component. Section 3 presents an illustrative example. The paper ends with a note on future work.

2. Method

Simulation Environment Let $A = \Gamma \cup \Gamma^c$ be the space of all possible behaviour consisting of sequences of elementary actions. Consider a simulation environment where the agents generate action sequences, such that each instance $a \in A$ is a finite sequence of actions $(a_1, ..., a_n)$ generated by an agent observing a sequence of states $(s_0, s_1, ..., s_n)$ (as in standard Markov-decision process formalisms). The actions available to the agent may be constrained by rules that represent *hard constraints* on the action space. These may be for instance physical constraints, or explicit legal constraints that would cause non-compliance if violated. The advantage of dealing with economic crime is that a monetary gain motivating the non-compliant behaviour is usually present, which means we can assume a utility function $u : A \to \mathbb{R}$ evaluating the quality of the action sequence.[2]

[2]Not all non-compliance can be quantified by monetary gain, or even well-defined utility function might not be available. One can relax this assumption by considering more general preferential structure.

Locally optimal forms Under hard constraints of the action space, the fraudulent agent can be seen as optimizing the utility function to discover sequences that yield high utility. Fraud schemes as such do not relate only to a specific type of behaviour, but to a class of behaviours that follows a certain higher-level behavioural pattern. Intuitively, one can expect that each scheme might have its representative form, that is, a form that illustrates the essence of a particular fraud scheme. With these representative forms, one does not need to list all the instances of Γ, achieving a compression of possibly infinite set Γ into a finite number of classes. We speak of sequences that attain a local maximum as *locally optimal forms*.

Exploration We consider as given an initial dataset D consisting of behavioural instances labeled as fraudulent, non-fraudulent or unknown. These instances can be organized into a graph $\mathscr{G}$, where each edge is weighted by the similarity function d_Γ. This graph structure can then be extended by an artificial *fraudulent agent*, which is an entity (standing for an individual agent, or a group or coalition of coordinated agents) capable of generating fraudulent behaviour in the simulation environment, following a certain rationality (eg. utility maximization), such that most, if not all, locally optimal forms are discovered.

Querying the oracle and learning The fraudulent agent is generating action sequences with high utility, but since implicit rules are not formalized, the only way how to know if an action sequence $a \in A$ violates them is to query an *oracle* $Q : A \to \{-1, 1\}$. A prototypical oracle would be a human legal analyst with the competence to pass a judgement on the input behavioural instance.

Since the number of queries of the oracle is relatively small compared to all possible sequences that can be generated by the agent, one needs to choose queried sequences wisely, such that maximal information gain is obtained by the query. A selection rule R_D is applied to select which unlabelled instances of $\mathscr{G}$ are likely to provide the most relevant information for the compressed representation, as a strategy for active learning. The chosen instances are then labelled by the oracle, and used to find locally optimal forms of Γ by utilizing local similarities. The compressed representation can be used as a classifier, by checking the membership of the instance to the compressed set Γ.

3. Illustrative example

Tracing the boundary between tax planning, tax avoidance, and the role of tax havens is recognized to be a debated topic both in the academic literature and in policy circles [7], and offers therefore a prototypical domain of application of the proposed method. In our example environment, a fraudulent agent is used to generate instances of behaviour. The system then aims to compress the fraudulent instances. The aim of the example is to illustrate the proposed framework, and to show how different choice of the selection rule R_D are influencing the search process of the true compressed representation Γ.

Action space and constraints Suppose organizations are able to move certain assets (eg. goods, capital, data...) from a place to another, and can decide how to act depending on the economic payoff (eg. transactional cost, income) and the norms in place. Let us represent the movement of a single asset of a single organization as a transaction system. Places fall in one of four categories denoted by letters $T = \{r, g, b, y\}$ (standing for red,

green, blue, and yellow). Elementary (moving) actions are pairs belonging to $T \times T$. Finite sequences of these actions form the action space A. Viewing the sequence as a string, suppose that there are three hard constraints (derived from a partially formalized legal system) restricting A only to strings that do not contain gr, bg and ry as substrings (ie. moving the asset from green to red is not allowed, etc.). Moreover, the string rb cannot occur more than twice as a substring (ie. moving the asset twice from red to blue is not allowed).

Oracle To make the example easier to work with, the oracle is not a human as it would be in a practical setting. The oracle is defined by the ability to decide (non)compliance, and this decision is made by applying a regular expression `g[by]+g`. Any $a \in A$ that is matching the regular expression is regarded as a violation of implicit rule, which could be thought eg. as a known scenario of non-compliance by analysts in that social domain.

Utility function Consider an agent moving a certain capital $u_0 = 1000$ of assets from a place G to a place H, acting as initial and terminal places, with both of them being of type g. Each movement incurs a cost (eg. a tax) given by a transaction table M with values listed in the following table:

	r	g	b	y
r	0.050	0.05	-0.00525 (first time), ∞ (others)	∞
g	∞	0.40	0.001	0.40
b	0.005	∞	0.10	0.00
y	0.05	0.0001	0.40	0.05

Hard constraints, such as gr, can be conveniently represented in the table as transactions with infinite cost. Note that rb has a negative cost, ie. the agent derives a benefit; we also hard-constrain it to be applied only once (otherwise all action sequences with high utility would be only claiming benefits). The value of the asset is updated depending on the transaction as $u_{t+1} = u_t(1 - M_{i,j})$, where $i, j \in T$. The agent aims to find such a sequence of transactions that maximize the value of u_n.

(Non)compliance knowledge base and compression The data sample D forms a graph as the one illustrated in Figure 1. A vertex $x \in \Gamma$ is linked to an unlabelled vertex y if and only if $d_\Gamma(x,y) \leq \tau$, where d_Γ is Levenshtein distance (most commonly used string edit distance). The query strategy $R_D(u, \mathscr{G})$, necessary for active learning of the compressed representation of Γ, is evaluated on four different strategies by always querying either (i) **random**: a random unlabelled instance; (ii) **max utility**: an unlabelled instance with the highest utility value; (iii) **max degree**: an unlabelled instance with the highest node degree in the graph; (iv) **min degree**: an unlabelled instance with the lowest node degree in the graph.

Once the knowledge base of the possible behavioural instances is updated, the system performs a search (eg. brute force) of a regular expression with a maximal fixed length (eg. 10) over a hypothesis space given by the alphabet $\{r, g, b, y, [,], +, *\}$ to obtain a classifier expression that maximizes accuracy of classification. The accuracy is determined by oracle, that has the access to true labels of all instances.[3]

[3]The oracle here is used out of convenience, as it is possible to evaluate the true accuracy of a compression model. In practice, this evaluation would be done in a more standard way by splitting the labelled data into a training set and a test set.

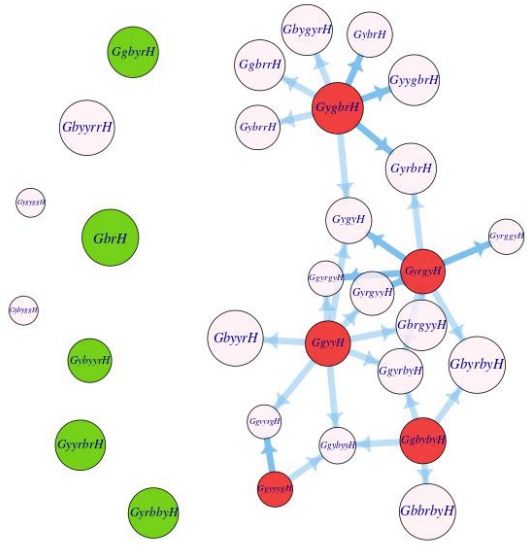


Figure 1. Vertexes coloured in green are compliant behaviours, while the red are non-compliant. The other vertexes are unlabelled. The bigger the size of the vertex, the greater the utility value of the sequence. Edge transparency depends on Levenshtein distance between the instances of behaviour.

Fraud compression results Let us assume that the fraudulent agent has generated all action sequences of length at most 7 with starting address G and ending address H that satisfy the hard constraints. Out of these 345 sequences, only one has a label at the start. This knowledge is meant only to facilitate the bootstrapping of the method. Then we can

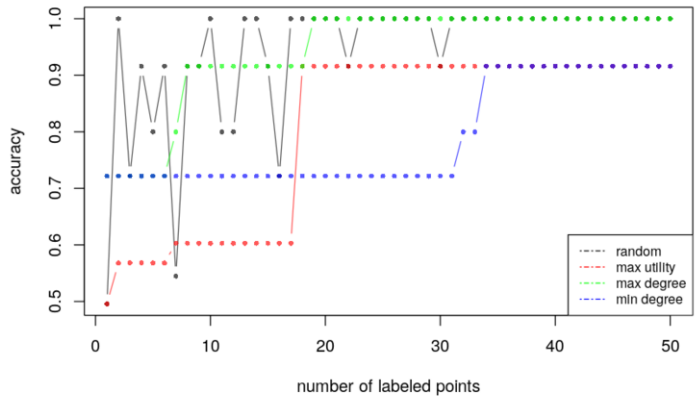


Figure 2. Min degree and **max utility** decision rules fail to converge to true expression with less than 50 oracle queries, which means the rules are biased for the less informative properties of the instance. The **random** decision rule converges, but seems to be very volatile, as instances are queried at random. The most suitable decision rule looks to be the **max degree** rule, where the agent aims to acquire labels from the oracle on instances that are most similar to already known non-compliant instances.

update the knowledge graph by querying the oracle.

Each query provides more information for the inference of the compression model. After a sufficient amount of queries, the system finds the optimal compression expression `g[by]+g`. For this compressed representation, the locally optimal form is *gbyg*. It is easy to see that any extension of this sequence that is still a member of Γ will have lower utility. For example, the non-compliant instance *gbygbyg* has the second-highest utility.

On Figure 2 one can observe how the classification accuracy of the classifier expression is separating the hard constrained space A into Γ and Γ^c. As the fraudulent agent is obtaining more knowledge by querying the oracle more times, the results converge to 100% accuracy for two out of four decision rules considered.

4. Perspectives

The theoretical considerations presented in this study provide initial foundations for a more structured approach to non-compliance detection via compression. In combination with active learning and graph-based semi-supervised learning, the framework is capable to discover compressed representations of non-compliant space, that can be later integrated into the formal legal system. The goal of the future research is to improve scalability of the framework by considering more efficient, but still explainable, compression methods over the non-compliant space, and more sophisticated models of the fraudulent agent.

References

[1] Ashley, K.D.: Case-based reasoning and its implications for legal expert systems. Artificial Intelligence and Law **1**(2-3), 113–208 (1992)
[2] Bench-Capon, T.J., Coenen, F.P.: Isomorphism and legal knowledge based systems. Artificial Intelligence and Law **1**(1), 65–86 (1992)
[3] van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Machine Learning **109**(2), 373–440 (feb 2020)
[4] Fursov, I., Morozov, M., Kaploukhaya, N., Kovtun, E., Rivera-Castro, R., Gusev, G., Babaev, D., Kireev, I., Zaytsev, A., Burnaev, E.: Adversarial Attacks on Deep Models for Financial Transaction Records. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. vol. 1, pp. 2868–2878. ACM, New York, NY, USA (aug 2021)
[5] Hemberg, E., Rosen, J., Warner, G., Wijesinghe, S., O'Reilly, U.M.: Detecting tax evasion: a co-evolutionary approach. Artificial Intelligence and Law **24**(2), 149–182 (2016)
[6] Lebichot, B., Braun, F., Caelen, O., Saerens, M.: A graph-based, semi-supervised, credit card fraud detection system. In: Int. Workshop on Complex Networks and their Applications. pp. 721–733 (2016)
[7] Merks, P.: Tax evasion, tax avoidance and tax planning. Intertax pp. 272–281 (2006)
[8] Sileno, G., Boer, A., van Engers, T.: Reading agendas between the lines, an exercise. Artificial Intelligence and Law **25**(1), 89–106 (2017)
[9] Zhu, X., Ao, X., Qin, Z., Chang, Y., Liu, Y., He, Q., Li, J.: Intelligent financial fraud detection practices in post-pandemic era. The Innovation **2**(4), 100176 (2021)

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220464

Fundamental Revisions on Constraint Hierarchies for Ethical Norms

Wachara FUNGWACHARAKORN [a] and Kanae TSUSHIMA [a] and Ken SATOH [a]
[a] *National Institute of Informatics, Sokendai University, Tokyo, Japan*

Abstract. This paper studies constraint hierarchies for *ethical norms*, which are unwritten and may be relaxed if they conflict with stronger norms. Since such ethical norms are unwritten, initial representations of ethical norms may contain errors. For correcting those errors, this paper examines fundamental revisions on constraint hierarchies for ethical norms. Although some revisions on representations for ethical norms have been suggested, revisions on constraint hierarchies for ethical norms have not been completely investigated. In this paper, we categorize two fundamental types of revisions on such constraint hierarchies, namely preference revision and content revision. We also compare effects of those revisions in the criteria of syntactic and semantic changes, which are common criteria of revisions on legal theories. From the comparison, we found that preference revision tentatively makes lower syntactic changes. However, its computation is intractable, incomplete, and potentially makes a large number of semantic changes. On the other hand, we show that content revision on constraint hierarchies can make a small number of semantic changes. However, the content revision tentatively produce a large number of syntactic changes. This comparison leads to the possibility of optimization between preference revision and content revision, which we think is an interesting future work.

Keywords. constraint hierarchies, soft constraints, belief revision

1. Introduction

This paper studies representing social norms using constraint hierarchies [1]. Constraint hierarchies consist of two types of constraints: *hard constraints* (or *required constraints*) and *soft constraints* (or *non-required constraints*). This paper takes a simplified distinction as a starting point of analysis of ethical norms along the research of [2,3,4] by distinguishing social norms into two types: *legal norms* represented as *hard constraints* and *ethical norms* represented as *soft constraints*.

Since ethical norms are unwritten, it is infeasible to represent ethical norms correctly in the first place. Therefore, this paper explores common revisions for correcting errors in representations of ethical norms. One common revision used in such representations is preference revision. Preference revision has an advantage as it does not change the content of the representations. However, it suffers from intractability of computations, incompleteness, and possibility of huge ef-

fects from the revision [5]. In contrast to preference revision, we define content revision, which can make a small number of semantic changes. However, content revision suffers from making large syntactic changes. Then, we demonstrate these behaviors in constraint hierarchies and present *DF-contraction* and *DF-expansion* computations for content revision, which are complete and make a small number of semantic changes. After that, we discuss the possibility of optimization between preference revision and content revision.

2. Constraint Hierarchies and Fundamental Revisions

In this paper, we restrict variables to Boolean variables or propositions. A valuation is a mapping θ from the set of all variables to {TRUE,FALSE} and we write Θ as the set of all valuations. Constraints in this paper are hence restricted to propositional logical formulae. For a constraint c, we define an error function $e(c, \theta)$ which returns zero iff θ satisfies c ($c\theta$ holds), and returns one iff θ does not satisfy c. A constraint hierarchy is a set of constraints with strength levels, i.e. $[k]c$ where k is a non-negative real number representing a level of constraint and c is a constraint. Conventionally, constraints with strength level 0 are called *hard constraints* or *required constraints* and other constraints are called *soft constraints* or *required constraints*, where a larger value of the level means the strength is weaker. The level is typically a non-negative integer but, in our problem, we expand the domain of strength into non-negative real numbers so that we can change the level more flexibly. However, a real-number level can still be treated as an integer in practice.

Let H be a constraint hierarchy. We write H_k be the set of all constraints in H with strength level k. We follow the definitions in [1] by defining:

$$\begin{aligned} S_0 &= \{\theta \in \Theta |\ \forall c \in H_0, c\theta\ holds\} \\ S &= \{\sigma \in S_0 |\ \forall \theta \in S_0, \neg better(\theta, \sigma, H)\} \end{aligned} \tag{1}$$

Intuitively, S_0 is a set of valuations that satisfy all constraints in the set of hard constraints H_0, and S is a valuation in S_0 that satisfies soft constraints as much as possible. A member of S is thus called a solution to H and S is called a solution set. We say H is trivial if $S = S_0$ otherwise we say it is non-trivial. *better* in (1) is an arbitrary comparator between two valuations θ and σ with respect to constraints H. In our problem, we choose the most basic one called *locally-better*. θ is *locally-better* than σ if there exists a strength $k > 0$ such that (i) for every strength stronger than k, the error after applying θ is equal to that after applying σ, i.e. $\forall i \in (0, k)\ \forall p \in H_i\ e(p, \theta) = e(p, \sigma)$, and (ii) at strength k, the error is strictly less than at least one constraint and less than or equal for all the rest, i.e. $\exists q \in H_k\ e(q, \theta) < e(q, \sigma) \wedge \forall r \in H_k\ e(r, \theta) \leq e(r, \sigma)$.

When the solution is unexpected, it means that the solution set is different from the intended set in the user's mind. In concept learning [6], a user works as a membership query that can answer correctly whether or not a queried valuation is intended. The revision task is to find a new constraint hierarchy H' such that a set of solutions to H' satisfies the results of the membership query, i.e. an intended

valuation must be a solution to H' and an unintended valuation must not be a solution to H'. Furthermore, the revision should change from H to H' as little as possible. This criterion is often called *minimal revision*. There are basically two metrics for counting the number of changes: one is based on syntax and another is based on semantics. Our syntax-based metric is adapted from Theory Distance Metric [7]. The metric is defined as follows.

Definition 1 (Syntax-based Metric)**.** *Let H, H_r, and H' be constraint hierarchies. A revision transformation r is such that $r(H) = H_r$, and H_r is obtained from H by edit operations as follows:*

1. *creating a new constraint with a strength level and one literal*
2. *adding one literal using $\vee$ (a logical or) or $\wedge$ (a logical and) to a constraint in H (adding parentheses may be needed to reduce ambiguity but it does not count as an operation)*
3. *removing one literal from a constraint in H (a constraint is deleted if the constraint has no literal left)*
4. *changing a strength of constraint (Operation 4 is specific to constraint hierarchies.)*

The syntactic changes between H and H' are determined by the smallest number of applying the revision transformation r to revise H into H', i.e. $H' = r^n(H)$ and there is no $m < n$ such that $H' = r^m(H)$.

On the other hand, our semantics-based metric is based on the change of the solutions. The metric is defined as follows.

Definition 2 (Semantics-based Metric)**.** *Let H and H' be constraint hierarchies. Let S and S' be the set of solutions to H and H' respectively. The semantic changes between H and H' are determined by the size of symmetric difference S and S', i.e. the number of valuations that belong to only one set.*

Next, we explore two fundamental types of revisions, namely preference revision and content revision. Preference revision refers to a revision that changes only the strengths of constraints but not their contents. In other words, it uses only Operation 4 in Definition 1. Changing a strength of constraint (Operation 4 in Definition 1) can be considered as an operation with the lowest cost for syntactic changes because the content of the constraints is still kept. Hence, preference revision is often considered to be the lowest cost for syntactic changes also. However, preference revision also has some limitations [5]:

1. (Intractable) Although an evaluation table is given, finding how to change a constraint to satisfy the user's intention is *NP-complete* since we need to use *better* to compare the target solution to other valuations.
2. (Incomplete) Only changing a strength of constraint cannot change the solutions in some constraint hierarchies. An obvious example is a constraint hierarchy with only one constraint.
3. (Huge Effect) Although preference revision often makes a small number of syntactic changes, it sometimes makes a large number of semantic changes.

Let us demonstrate Limitation 3 using a constraint hierarchy representing a classic ethical example of the Righteous Lies Problem.

Example 1 (Righteous Lies Problem)**.** *Suppose there is a situation where we need to tell lies to protect others and the only way to protect others is to tell lies (tell_lies* $\leftrightarrow$ *protect). From an ethical point of view, we should protect others, we should not tell lies, and it is common to prioritize protecting others over not telling lies. However, we later realize that protecting criminals could be a case of protecting others (prot_criminal* $\rightarrow$ *protect). Hence, we can represent the current setting as the following constraint hierarchy as shown on the left. For ease of exposition, we write hard, strong, weak instead of* $0, 1, 2$ *respectively.*

$[hard]\ tell\_lies \leftrightarrow protect$
$[hard]\ prot\_criminal \rightarrow protect$
$[strong]\ protect$
$[weak]\ \ \neg tell\_lies$

$[hard]\ tell\_lies \leftrightarrow protect$
$[hard]\ prot\_criminal \rightarrow protect$
$[strong]\ \neg tell\_lies$
$[weak]\ \ protect$

Table 1. Effect of preference revision on solutions to the Righteous Lies Problem

tell_lies	*protect*	*prot_criminal*	Old Solution ?	Intended ?	New Solution ?
TRUE	TRUE	TRUE	yes	no	no
TRUE	TRUE	FALSE	yes	-	no
FALSE	FALSE	FALSE	no	-	yes

Table 1 shows the effect of preference revision on solutions to the Righteous Lies Problem. The old solutions indicate that we should tell lies to protect others, regardless whether they are criminals. Then, we may not intend to protect criminals otherwise it causes more losses to society. However, the new solution from the preference revision as shown on the right gives unexpected results as it re-prioritizes not telling lies over protecting others. From a logical point of view, this revision is unexpected because it makes too many semantic changes.

In contrast to preference revision, let us define content revision as a revision that changes only the content of the constraints but not their strengths. In other words, it uses only Operation 1-3 in Definition 1. Following the definition, we can define two computations for content revision on non-trivial constraint hierarchies. The first computation is *DF-contraction*, for contracting the set of solutions to exclude an unintended solution. The second computation is *DF-expansion*, for expanding the set of solutions to include an intended valuation that satisfies all hard constraints but not yet a solution. These computations can always revise constraint hierarchies with only one semantic change as the following theorems.

Theorem 1. *Given a non-trivial constraint hierarchy* H*. Let* S *be the set of solutions to* H*,* $\theta \in S$*, clause*(θ) *be a conjunctive clause corresponding to an valuation* θ *(e.g. clause*$(a = TRUE, b = FALSE) = a \wedge \neg b$*).* DF-contraction *revises* H *into* H' *as follows.*

for every soft constraint $c \in H$ *such that* $c\theta$ *holds*

remove clause(θ) *from* c *in disjunctive form*

If H' is non-trivial, then DF-contraction *is correct (i.e. θ is not a solution to H'), complete (i.e. can always find H'), makes only one semantic change (i.e. θ is the only member of the symmetric difference between S and S').*

Theorem 2. *Given a non-trivial constraint hierarchy H. Let S be the set of solutions to H, $\theta \in S_0 \setminus S$ where S_0 is the set of all valuations that satisfy all hard constraints, as defined in (1) , clause(θ) be a conjunctive clause corresponding to an valuation θ (e.g. clause$(a = TRUE, b = FALSE) = a \wedge \neg b$).* DF-expansion *revises H into H' as follows.*

for every soft constraint $c \in H$ such that $c\theta$ does not hold

add clause(θ) with $\vee$ (a logical or) to c

DF-expansion *is correct (i.e. θ is a solution to H'), complete (i.e. can always find H'), and makes only one semantic change (i.e. θ is the only member of the symmetric difference between S and the set S' of solutions to H').*

Let us illustrate *DF-contraction* in the Righteous Lies Problem example (Example 1), we can revise the constraint by removing ($tell\_lies \wedge protect \wedge prot\_criminal$) from *protect*, which can be considered in disjunctive form as follows.

$(tell\_lies \wedge protect \wedge prot\_criminal) \vee (\neg tell\_lies \wedge protect \wedge prot\_criminal) \vee$
$(tell\_lies \wedge protect \wedge \neg prot\_criminal) \vee (\neg tell\_lies \wedge protect \wedge \neg prot\_criminal)$

Hence, *protect* is revised into

$(\neg tell\_lies \wedge protect \wedge prot\_criminal) \vee$
$(tell\_lies \wedge protect \wedge \neg prot\_criminal) \vee (\neg tell\_lies \wedge protect \wedge \neg prot\_criminal)$

or $protect \wedge (\neg tell\_lies \vee \neg prot\_criminal)$ in minimal form. As a result, *DF-contraction* gives the following constraint hierarchy.

$[hard]\ tell\_lies \leftrightarrow protect$
$[hard]\ prot\_criminal \rightarrow protect$
$[strong]\ protect \wedge (\neg tell\_lies \vee \neg prot\_criminal)$
$[weak]\quad \neg tell\_lies$

Table 2. New solutions to the content revision in the Righteous Lies Problem example

tell_lies	*protect*	*prot_criminal*	Old Solution ?	Intended ?	New Solution ?
TRUE	TRUE	TRUE	yes	no	no
TRUE	TRUE	FALSE	yes	-	yes
FALSE	FALSE	FALSE	no	-	no

Table 2 shows the new solution to this revision, which makes only one semantic change. However, the revision makes three syntactic changes. One reason for the drawback of content revision is that it does not consider interactions between constraints, as opposed to preference revision. In future work, it is interesting to propose an optimization between preference revision and content revision to adjust between syntactic changes and semantic changes from both types of revision.

3. Conclusion

This paper presents constraint hierarchies for representing ethical norms, which refer to norms that can be conflicted hence they could not be all satisfied sometimes. Such norms fit well with soft constraints in constraint hierarchies as they divide constraints into hard constraints, which must be all satisfied, and soft constraints, which should be satisfied as much as possible. A solution to a constraint hierarchy is intuitively a valuation that satisfies all the hard constraints and no other valuations that satisfy a larger set of soft constraints. This paper also investigates two fundamental types of revision on constraint hierarchies to cover the changes of ethical norms. The first type is preference revision, which changes only the strengths of the constraints but not their contents. Hence, it benefits from making fewer changes in the syntactic sense. However, it cannot always change the solutions as intended and can make a large number of changes in the semantic sense. The second type is content revision, which changes only the contents of constraints but not their strengths. We introduce *DF-contraction* and *DF-expansion* computations for content revisions on constraint hierarchies. They can always change the solutions as intended with only one semantic change, but they mostly make a large number of syntactic changes. Hence, an optimization between preference revision and content revision is an interesting future work.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103 and JP19H05470 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4, Japan.

References

[1] A. Borning, B. Freeman-Benson and M. Wilson, Constraint hierarchies, *LISP and symbolic computation* **5**(3) (1992), 223–270.
[2] J. Greene, F. Rossi, J. Tasioulas, K.B. Venable and B. Williams, Embedding ethical principles in collective decision support systems, in: *Thirtieth AAAI conference on artificial intelligence*, 2016.
[3] L. Dennis, M. Fisher, M. Slavkovik and M. Webster, Formal verification of ethical choices in autonomous systems, *Robotics and Autonomous Systems* **77** (2016), 1–14.
[4] K. Satoh, J.-G. Ganascia, G. Bourgne and A. Paschke, Overview of RECOMP project, in: *International Workshop on Computational Machine Ethics, International Conference on Principles of Knowledge Representation and Reasoning*, 2021.
[5] G. Governatori, F. Olivieri, S. Scannapieco and M. Cristani, Superiority based revision of defeasible theories, in: *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, Springer, 2010, pp. 104–118.
[6] D. Angluin, Queries and concept learning, *Machine learning* **2**(4) (1988), 319–342.
[7] J. Wogulis and M.J. Pazzani, A methodology for evaluating theory revision systems: Results with Audrey II, in: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Francisco,CA,USA, 1993, pp. 1128–1134.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220465

Predicting Outcomes of Italian VAT Decisions [1]

Federico GALLI [b], Giulia GRUNDLER [c], Alessia FIDELANGELI [b], Andrea GALASSI [c,2], Francesca LAGIOIA [a,b,2], Elena PALMIERI [c], Federico RUGGERI [c], Giovanni SARTOR [a,b] and Paolo TORRONI [c]

[a] *European University Institute, Law Department, Italy*
[b] *CIRSFID Alma-AI, Faculty of Law, University of Bologna, Italy*
[c] *DISI, Alma-AI, University of Bologna, Italy*

Abstract. This study aims at predicting the outcomes of legal cases based on the textual content of judicial decisions. We present a new corpus of Italian documents, consisting of 226 annotated decisions on Value Added Tax by Regional Tax law commissions. We address the task of predicting whether a request is upheld or rejected in the final decision. We employ traditional classifiers and NLP methods to assess which parts of the decision are more informative for the task.

Keywords. Predictive Justice, Machine Learning, Natural Language Processing, Case Law, Tax Law

1. Introduction

Outcome prediction has recently enjoyed renewed interest thanks to the availability of judicial data and breakthroughs in machine learning and NLP techniques [1,2,3]. Current approaches rely either on features describing aspects of the cases [4,5], which could be unrelated to their merit [6,7]; or on the textual content of the case decisions [8,9]. Our study falls under the second approach, which applies analytics techniques to automatically identify correlations between the textual content of decisions and their outcomes. In particular, we aim to determine the correlations between the requests by the parties and the uphold/rejection of such requests by the Regional Tax Commissions (second–instance Tax Courts).

Recent advances on outcome prediction include work by Aletras et al. [8], who predicted violations of some articles of the European Convention on Human Rights, using a dataset of 584 European Court of Human Rights decisions using Support Vector Machine (SVM), Bag-of-Words (n-grams) and topical features and later by [9], who expanded said dataset to obtain a higher performance. Several works focused on national case law. For example, [10] applied a linear SVM classifier trained on lexical features to predict the legal area and the outcome of cases by the French Supreme Court. [11] used logistic regression and SVM to predict the outcomes of Bavarian court decisions. Chinese case law was addressed by [12,13,14] among others. To the best of our knowledge, this

[1] This work has been partially supported by the H2020 ERC Project "CompuLaw" (G.A. 833647); the ADELE project (G.A. 101007420) under the European Union's Justice programme, the LAILA project (G.A. 2017NCPZ22) under the Italian Ministry of Education and Research's PRIN programme
[2] Corresponding authors: Francesca Lagioia: francesca.lagioia@eui.eu, Andrea Galassi: a.galassi@unibo.it

is the first study on outcome prediction of Italian decisions, and also the first one in the VAT domain. We focus on appeal (second-instance) decisions. We model this as a binary classification task, whose goal is predicting whether a given request by the parties is accepted or rejected by the appeal court. A distinctive aspect of our work consists in covering requests and decisions addressing different aspects of VAT (e.g., taxable transactions, exemptions, out-of-scope transactions) rather than a single specific issue.

2. The corpus

The source corpus consists of 226 Italian second-instance decisions on Value Added Tax (VAT) by the Regional Tax Commissions from various judicial districts.[3] The decisions, downloaded from the *Giustizia Tributaria* database,[4] range from 2010 to 2022 and concern taxable transactions, exemptions, out-of-scope transactions, and the right to obtain a deduction. They contain 303 first-instance requests, of which 84 rejected, 126 upheld, and 5 with other outcomes, and 490 second-instance requests, of which 129 rejected, 99 upheld, and 22 with other outcomes. The number of requests is higher than the number of outcomes since a decision on a particular request may imply the uphold or rejection of other requests, which thus are not explicitly addressed. We chose to focus on VAT Italian cases since: (a) though some AI applications exist within the Italian Tax Administration, they do not yet address the case law; (b) VAT is harmonised at the European level, governed by the VAT Directive (Directive 2006/112/EC);[5] (c) the CJEU case law on this matter favours the uniform and consistent interpretation of legal norms, principles and concepts; (d) VAT is a relatively narrow a self-contained branch of the law; (e) Italian VAT decisions have a rather consistent structure; (f) they affect –apart from lawyers– accountants, public servants and millions of taxpayers; (g) we have domain expertise.

Appeal VAT decisions have a standard structure consisting of the following parts:

1. *Introduction*, reporting (i) the number of the decision; (ii) the composition of the judicial panel, (iii) the parties and their lawyers (if present);
2. *Account of the Proceeding*, reporting facts related to both the pre-litigation phase and the first-instance proceedings (e.g., the parties' requests, claims and arguments as well as first-instance decisions by the Provincial Tax Commission);
3. *Parties' Requests* in second-instance proceedings, often presented with the related claims and arguments;
4. *Justification*, the statement of reasons in fact and in law supporting the decisions;
5. *Final Ruling*, by the Regional Tax Commission, including the decision on costs.

Annotation guidelines were defined through an iterative refinement process of validation, evaluation of the agreement, and discussion. The labelling was done by two VAT experts. The conflicts between annotators have been discussed and solved with a third legal expert. We focused on the identification of the following elements: (i) the parties, (ii) their first and second-instance requests, (ii) the related claims and (iii) arguments; (iv) the Provincial and Regional Tax Commissions' justifications, and (v) first and second-instance decisions, as reported in the different parts of the analysed documents. Such

[3] The corpus and our code are available at https://github.com/adele-project/italianVAT

[4] Tax Justice database accessible at: https://www.giustizia-tributaria.it/.

[5] In Italy, the EU VAT Directive has been implemented by the Presidential Decree 633/1972.

information can be of different lengths and details. Moreover, it is often enclosed within the same portion of text. For this reason, we identified hierarchical levels of annotation.

The parties to the proceeding (*part*) – i.e., taxpayers and tax authorities, being appellants or respondents in the appeal proceedings, plaintiff or defendants in first-instance proceedings – are mentioned in the introductory section and are identified through their names and residential addresses. The parties' requests, claims, and arguments concerning the first-instance proceeding are presented in the *Account of the proceeding* section, while those concerning the second-instance are reported in the *Parties' Requests* section. Claims and arguments may be missing for certain requests, especially first-instance requests. Requests, claims, and arguments are often included in the same sentence. To identify the relevant segments we relied on (a) recurrent linguistic indicators, including keywords and word patterns; and (b) context indicators, as detailed in the following.

Requests (*req*) may be distinguished in *main requests* and responses to them, i.e. *counter-requests*. They are often characterised by different linguistic indicators, which may help the annotators in correctly labelling the relevant textual fragments. In first-instance proceedings, main requests are made by taxpayers and often concern the annulment of the Tax Administration's acts. Those made in the second–instance can be presented either by taxpayers or by tax authorities and are often aimed at reversing first-instance decisions. The set of keywords and word patterns signalling the main request includes (a) verbs expressing the action of requesting or concluding with a request; (b) nouns identifying the measure requested, such as the reversal of the first-instance decision; (c) word patterns specifying these ideas. Counter request(s) are usually signalled by word patterns referring to requests for the rejection of the appellant's claim or the acceptance of the respondents' claim. Each request is denoted by (i) a unique id, (ii) the degree of judgement in which it has been made and (iii) the party making the request.

Claims (*claim*) are the ultimate reasons for grounding a request, usually supported by premises. They may concern (a) substantive facts (e.g., the lack of competence of the administrative tax office in adopting a particular pre-litigation decision), or (b) procedural facts (e.g., the violation of a procedural norm). Each claim is denoted by 3 mandatory attributes (id, degree and party making the claim), as well as 2 optional attributes used to identify whether a claim is supporting or attacking a request. Recurrent linguistic indicators include: (a) a set of terms, and in particular, certain verbal forms indicating an argumentative attitude; and (b) word patterns having the same function.

Arguments (*arg*) are statements that support or attack a claim. Arguments can be legal or factual. Each argument has the usual three mandatory attributes (id, degree and party making the argument), plus two optional attributes specifying whether an argument supports or attacks one or more claims. An argument is often denoted by word patterns referring to a grounding relation.

Justifications (*mot*) report the inferences made by the Court, leading to decisions on claims or requests raised by the parties. Each justification is characterised by: (i) a unique id, (ii) the degree of the proceeding, and (iii) its object, which can be a request or a claim. Each justification is generally delimited by a heading and includes word patterns indicating the different requests/claims raised by the parties.

First-instance decisions (*dec*) are concisely presented in the *Account of the Proceedings*. Second-instance decisions are reported in the *Final Ruling* section. Each decision is denoted by (i) a unique decision id, (ii) the degree of judgement in which the decision was taken, (iii) its object and (iv) outcome. Possible outcomes are: uphold, reject,

Table 1. Cohen's κ for each element, attribute, and link.

Element	κ
part	0.87
req	0.85
arg	0.94
claim	0.86
mot	0.97
dec	0.91
avg	0.90

Attribute	κ
instance	0.93
party	0.70
outcome	1.00
avg	0.88

Link	κ
req-claim	0.66
claim-arg	0.78
req-mot	0.73
req-dec	0.77
claim-mot	0.37
avg	0.66

or other (inadmissibility of the parties' requests, extinction of the proceeding, referral to the first–instance Court, or absent decision since implicit in other decisions).

2.1. Inter-Annotator Agreement

Agreement was measured on 10 documents tagged by 2 annotators. Because a marked element may consist of a fragment of sentence, and each fragment can be labelled with multiple tags, we modelled the task as a multi-label binary classification task at the word level. Accordingly, we separately measured the agreement for each type of element and attribute. Table 1 shows the Cohen's κ [15] of each category. An average κ of 0.90 indicates a strong agreement. To properly evaluate the agreement on the attributes, we considered only cases with an agreement on the annotation of elements. An average κ of 0.88 indicates good agreement in all attributes. To measure the agreement on the links (i.e., the presence of attributes that express a relation between two elements), we considered each pair of element types as a separate case. For a given pair of element types, we considered for each decision all the possible pairs of elements that belong to such types (e.g., the first element must be a request, the second must be a claim). We treated agreement on links as a binary classification problem with the aim of predicting whether there is a link between a pair of elements. Results are reported in Table 1. We obtained a good agreement in almost all categories with an average κ of 0.66. For the specific cases of *req-claim* and *claim-arg* links, we also computed the agreement on the type of link, by only considering pairs where the two annotators agreed on the presence of a link, reaching the perfect score of κ=1.0 for both classes.

3. Methods

This study aims at *(i)* predicting the outcomes of second-instance decisions and *(ii)* assessing the extent to which different parts of the decision are informative for this task. Given that each decision can contain multiple requests, and each request can have a separate outcome (different from the general outcome of the case), we considered each request separately. For each of them, we identified claims (*claim*) and arguments (*arg*) as the basic information needed to predict the outcome. For this reason, we filtered out requests not associated with a claim as well as those not explicitly decided. Furthermore, we excluded those few decisions which do not reject or uphold a request (*other outcomes*) due to the lower number of samples. Thus, we addressed the task as a binary classification (reject/uphold). Our final dataset is composed of 112 rejected decisions and 71 uphold decisions.

Our aim is to predict the court's decisions on the basis of the information provided by the parties before the case. Such information are partially present in the decision, which reports the parties' *request*(s), *claim*(s), and *argument*(s). Nonetheless, in this study, we are also interested in assessing which part of the document can provide a valuable contri-

Table 2. Results on the second-instance requests.

	Inputs	req + arg + claim			r + a + c + mot			r + a + c + dec			r + a + c + m + d		
Embedding	**Classifier**	**Avg**	**rej**	**uph**	**Avg**	**rej**	**uph**	**Avg**	**rej**	**uph**	**Avg**	**rej**	**uph**
-	Random	0.49	0.57	0.43									
-	Majority	0.38	0.76	0.00									
TF-IDF	Linear SVC	0.64	0.77	0.50	0.59	0.75	0.44	0.64	0.78	0.50	0.61	0.75	0.46
TF-IDF	Random Forest	0.49	**0.78**	0.20	0.51	0.77	0.25	0.54	**0.79**	0.30	0.51	0.77	0.24
TF-IDF	Gaussian NB	0.57	0.75	0.39	0.56	0.70	0.41	0.55	0.73	0.37	0.58	0.71	0.44
TF-IDF	K Neighbors	0.57	0.72	0.41	0.58	0.71	0.45	0.58	0.73	0.43	0.59	0.71	0.47
TF-IDF	SVC	0.47	**0.78**	0.16	0.47	**0.78**	0.16	0.47	0.78	0.16	0.47	**0.78**	0.16
SBERT	Linear SVC	**0.68**	0.77	**0.60**	**0.66**	0.76	**0.57**	**0.68**	0.76	**0.60**	**0.66**	0.75	**0.56**
SBERT	Random Forest	0.58	**0.78**	0.38	0.56	0.77	0.34	0.58	0.77	0.40	0.59	**0.78**	0.40
SBERT	Gaussian NB	0.61	0.73	0.50	0.62	0.72	0.52	0.64	0.73	0.54	0.60	0.71	0.50
SBERT	K Neighbors	0.59	0.68	0.50	0.58	0.67	0.49	0.61	0.69	0.53	0.57	0.66	0.48
SBERT	SVC	0.47	0.75	0.19	0.54	**0.78**	0.31	0.52	0.76	0.28	0.54	0.77	0.31

bution to predicting the outcome. Since the *justification* and *decision* sections may hold important information about the outcome of a case, we decided to include those sections in the experiments, obtaining four experimental settings. In the first one, the inputs are *req*, *args*, and *claims*; the second and third are similar, but, respectively, also *mot* and *dec* are included; the fourth uses both *mot* and *dec*.

We pre-processed the decisions by removing stopwords and punctuation symbols. For each experimental setting, we concatenated together the representation obtained for the request and the ones obtained for the other sections. We adopted two representations of the input text: **TF-IDF** vectorization, which is based on the term frequency-inverse document frequency statistic; Sentence-BERT (**SBERT**) [16], a modification of the BERT model that produces semantically meaningful sentences' embeddings, mapping sentences with similar semantic content into vectors close to each other. As classifiers, we have chosen the following set of traditional machine learning models that have low computational requirements: Linear SVC, SVC, Random Forest, Gaussian Naive Bayes and K-Neighbours.[6] Experiments were conducted using 5-fold cross-validation with folds determined at the document level so that all the requests of the same decision belong to the same fold. The folds were created manually to balance their composition with respect to the reject/uphold and the first/second-instance distinctions.

4. Results

Tables 2 shows the results obtained through each combination of embeddings and classifiers in each setting, as well as two baselines (random and majority class). We measure the F1 score obtained for each class and their macro-average. The task of determining the decision outcome based only on the *claims* and *arguments* of the parties reaches a maximum score of 0.68 with Linear SVC and SBERT. The use of SBERT embeddings instead of TF-IDF is beneficial with all classifiers, leading to results not worse than the baselines. Overall, the Linear SVC classifier seems to give the best result in almost all the settings. There is a wide gap between the scores obtained in the two classes, which we speculate may be caused by their unbalanced distribution in the dataset. The *justification* section seems to give conflicting results: it slightly worsens the performance of Linear SVC, but it improves other classifiers. The introduction of the *decision* section has a limited impact, slightly improving some classifiers but without improving the best case.

[6] We used the default hyper-parameters offered by the `sci-kit learn` library.

The results obtained by the use of both sections are unexpected: most classifiers perform worse than when adding only the *decision* information. We speculate that, since in the *justification* the Court retraces the arguments of the parties, mentioning all the clues towards each possible outcome, its use may introduce noise that lowers the performance.

5. Conclusion

Ideally, one would aim to predict the decision of the court based on the information provided by the parties before the case. Our experiments approximate the ideal setup, by focusing on outcome prediction based on fragments in the narrative provided by courts, which we captured through the requests, claims, and arguments marked elements. To this end, we built a first-of-a-kind dataset, on Italian decisions and on the VAT domain. In the future, we plan to include information provided by the parties before the case. From the machine learning viewpoint, we plan to adopt oversampling or augmentation to balance the distribution of the classes in the dataset, and we are investigating more advanced neural architecture for classification or domain-specific embeddings.

References

[1] Ashley KD. A brief history of the changing roles of case prediction in AI and law. Law Context: A Socio-Legal J. 2019;36:93.

[2] Feng Y, Li C, Ng V. Legal Judgment Prediction: A Survey of the State of the Art. In: IJCAI. ijcai.org; 2022. p. 5461-9. Available from: https://doi.org/10.24963/ijcai.2022/765.

[3] Zhong H, Xiao C, Tu C, Zhang T, Liu Z, Sun M. How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence. In: ACL. Association for Computational Linguistics; 2020. p. 5218-30.

[4] Ashley KD, Brüninghaus S. Automatically classifying case texts and predicting outcomes. Artificial Intelligence and Law. 2009;17(2):125-65.

[5] Ashley KD, Keefer M. Ethical reasoning strategies and their relation to case-based instruction: some preliminary results. In: Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society. Routledge; 2019. p. 483-8.

[6] Surdeanu M, Nallapati R, Gregory G, Walker J, Manning C. Risk analysis for intellectual property litigation. In: ICAIL-2011. ACM; 2011. p. 116-20.

[7] Katz DM, Bommarito MJ, Blackman J. A general approach for predicting the behavior of the Supreme Court of the United States. PloS one. 2017;12(4):e0174698.

[8] Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. PeerJ Computer Science. 2016;2:e93.

[9] Medvedeva M, Vols M, Wieling M. Using machine learning to predict decisions of the European Court of Human Rights. Artificial Intelligence and Law. 2020;28(2):237-66.

[10] Sulea OM, Zampieri M, Malmasi S, Vela M, Dinu LP, Van Genabith J. Exploring the use of text classification in the legal domain. arXiv preprint arXiv:171009306. 2017.

[11] Urchs S, Mitrovic J, Granitzer M. Design and Implementation of German Legal Decision Corpora. In: ICAART (2); 2021. p. 515-21.

[12] Xiao C, Zhong H, Guo Z, Tu C, Liu Z, Sun M, et al. Cail2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:180702478. 2018.

[13] Luo B, Feng Y, Xu J, Zhang X, Zhao D. Learning to predict charges for criminal cases with legal basis. arXiv preprint arXiv:170709168. 2017.

[14] Long S, Tu C, Liu Z, Sun M. Automatic judgment prediction via legal reading comprehension. In: China National Conference on Chinese Computational Linguistics. Springer; 2019. p. 558-72.

[15] Cohen J. A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement. 1960;20:37 46.

[16] Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: EMNLP/IJCNLP (1). Association for Computational Linguistics; 2019. p. 3980-90.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220466

The Effectiveness of Bidirectional Generative Patent Language Models

Jieh-Sheng LEE [1]
National Yang Ming Chiao Tung University School of Law
ORCiD ID: Jieh-Sheng Lee https://orcid.org/0000-0002-0990-6170

Abstract. Generative patent language models can assist humans to write patent text more effectively. The question is how to measure effectiveness from a human-centric perspective and how to improve effectiveness. In this manuscript, a simplified design of the autocomplete function is proposed to increase effectiveness by more than 10%. With the simplified design, the effectiveness of autocomplete can reach more than 60%, which means that more than 60% of keystrokes can be saved by autocomplete. Since writing patent text does not necessarily start from the beginning to the end, a question is whether the generative model can assist a user no matter where to start writing. To answer the question, the generative models in this manuscript are pre-trained with training data in both directions. The generative models become bidirectional. Since text generation is bidirectional, the calculation of autocomplete effectiveness can be bidirectional and starts from anywhere in the text. After thorough experiments, a key finding is that the autocomplete effectiveness of a model for the same text remains similar no matter where the calculation starts. The finding indicates that such bidirectional models can assist a user at a similar level, no matter where the user starts to write.

Keywords. Patent, Natural Language Generation, Natural Language Processing, Deep Learning, Artificial Intelligence

1. Introduction

Large language models have achieved notable success in natural language generation (NLG) tasks in recent years. Until now, very few language models have been dedicated to the patent domain. Furthermore, most language models are autoregressive by predicting the *next* token after having read all the previous ones. Very few language models work backward by predicting the *previous* token. In this manuscript, the author pre-trained and fine-tuned large language models dedicated to the patent domain. The model can predict the previous token, in addition to predicting the next token. Predicting the previous token is implemented by reversing the tokens for both inputs and outputs. The training dataset also contains tokenized sequences in both forward and backward directions. By doing so, the patent-specific language models in this manuscript can generate patent text in both forward and backward directions. The motivation is to make patent text generation flexible because human writing does not necessarily start from the beginning to the

[1]Assistant Professor of Law. Ph.D. in Computer Science. Admitted in New York and passed the USPTO registration exam. Email: jasonlee@nycu.edu.tw

end. Assuming that the thought process in human's mind is a back-and-forth process, a bidirectional generative language model should assist humans more flexibly.

To evaluate the performance of generative patent language models, this manuscript follows the Autocomplete Effectiveness (AE) ratio proposed in [1]. The ratio is used to measure how many keystrokes can be saved for a user if an autocomplete function is provided and based on the generative model. The higher the AE ratio, the more the keystrokes are saved. The contributions of this manuscript include: (1) proposed a simplified version of the autocomplete function to reach higher AE ratios, (2) making the calculation of AE ratios bidirectional, and (3) observed similar AE ratios when using different starting positions of the text for calculation.

2. Related Work

This manuscript is the follow-up work of [1], which proposed the AE ratio to evaluate generative language models. In [1], the patent language model is called PatentGPT-J. The PatentGPT-J models are based on the GPT-J-6B [2] models and pre-trained with a patent corpus from scratch. The use of a Transformer [3] language model for patent text generation was first proposed in [4] by fine-tuning a GPT-2 [5] model with patent corpus. A Transformer model is a deep learning model that adopts the mechanism of self-attention and learns context by tracking relationships in sequential data. The idea of generating patent text backward was introduced in [6]. The research in [6] focuses on controlling patent text generation by structural metadata. However, the effective way to generate text backward and how to measure effectiveness were not addressed in [6]. Except for these works, patent text generation remains a niche research topic less explored.

3. Implementation

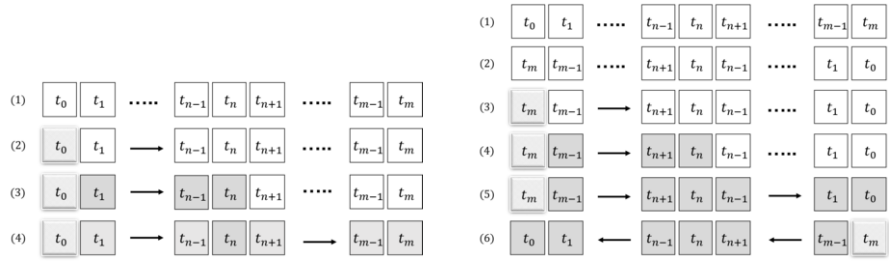
3.1. Simplified Autocomplete Function

The purpose of the AE ratio is to evaluate a language model from a human-centric perspective: how many manual keystrokes can be saved by the autocomplete function based on the generative language model? A higher AE ratio means that the autocomplete function works more effectively and more manual keystrokes are saved. This section describes how the autocomplete function can be simplified to obtain a higher AE ratio. The original AE ratio and the implementations of the PatentGPT-J models are described in [1]. The improvement in AE ratio is achieved by simplifying the user interface (UI). In the conceptual UI design in [1], a user has to press the "downarrow" (↓) key and the "tab" key to complete the autocomplete function. The user selects a preferred token in the top 10 tokens predicted by the model in this way. In this manuscript, a simplified UI is proposed using keys 0∼9 to represent the top 10 tokens. Therefore, the user can select a token by pressing only one key instead of multiple keystrokes. For example, in the previous design in [1], in order to select the 6th token, the user has to press the "downarrow" (↓) key five times and the "tab" key once. Six keystrokes are required. In this manuscript, the user can press the "5" key to select the 6th token of the top 10 tokens. Five keystrokes are saved ("0" key to represent the first token).

3.2. Back and Forth

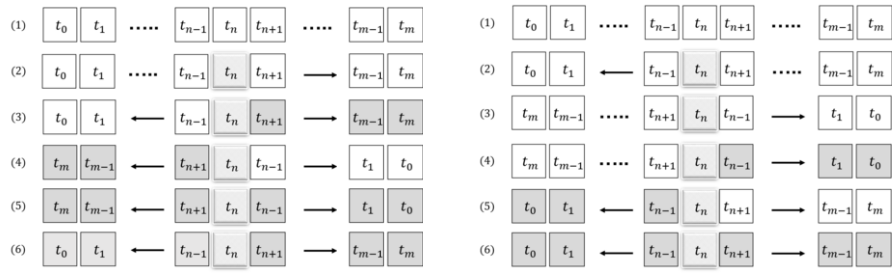
The calculation of the previous AE ratio in [1] is defined as calculating forward to reach the end. Fig. 1a shows how it works. In Fig. 1a, row (1) shows the tokens of the input text after tokenization. Row (2) indicates that the calculation starts from t_0. Row (3) indicates that the calculation moves forward. Row (4) shows that all tokens are calculated after reaching the end. The previous AE ratio in [1] addresses this use case only.

Fig. 1b shows a new use case in this manuscript: calculating backward and starting from the end of the input text. In Fig. 1b, row (1) is the same, showing the tokens of the input text. Row (2) shows the reversed sequence of row (1). Row (3) indicates that the calculation starts from t_m (the end of the original input text). Row (4) indicates that the calculation moves forward for the reversed tokens (backward in effect for the original tokens). Row (5) shows that all tokens are calculated. Row (6) shows the reversed sequence of row (5) to represent the calculation backward from the end of the original input text.



(a) Forward & From the Beginning (b) Backward & From the End

Figure 1. Text Generation



(a) Forward & In the Middle (b) Backward & In the Middle

Figure 2. Text Generation

Fig. 2a and Fig. 2b show two more use cases in this manuscript. In Fig. 2a, row (1) shows the same tokens of the input text. Row (2) indicates that the calculation starts from t_n in the middle (n can be any number between 0 and m) and is about to move forward. Row (3) indicates that the calculation moves forward from t_n, reaches t_m (the end), and is about to move backward. For moving backward, row (3) is reversed as row

(4). The calculation now moves forward starting from t_n again. Given t_m, t_{m-1}, ..., t_{n+1}, t_n as the context, row (5) indicates that the calculation moves forward and reaches t_0 (the beginning). Row (6) shows the reversed sequence of row (5) after moving forward and then backward. For brevity, the description of Fig. 2b is omitted because it is similar to how Fig. 2a works.

3.3. Dataset and Model sizes

The dataset in this manuscript is the USPTO TACD dataset in [1] which includes titles, abstracts, claims, and descriptions in granted patents. The coverage of the dataset ranges from 1976 to 2021 (1976∼2020 for training and 2021 for validation). For further details, please refer to [1]. The model sizes experimented in this manuscript are 6B, 1.6B, and 456M. These three model sizes outperform others (279M, 191M, 128M, and 115M) in [1]. After publication, the pre-trained and fine-tuned models in this manuscript will be released. The model configuration in this manuscript reuses [1]. In terms of training from scratch, this manuscript uses the original GPT-J-6B model and its source code available at [2].

4. Experiments

4.1. Experiment 1

This experiment aims to compare the algorithm in this manuscript with the previous one in [1]. The increase in effectiveness in this manuscript is more than 10%, as shown in the fifth column of Table 1. The first column shows the three models in this experiment: 456M, 1.6B, and 6B. The second column shows the direction for calculating the AE ratio: forward and backward. The third column shows the AE ratios in [1]. The fourth column shows the new AE ratios. The ratio of more than 62% in the fourth column means that more than 62% of keystrokes can be saved by autocomplete in this experiment. The effectiveness of the simplified autocomplete function using keys 0∼9 is validated. The previous combination of the "downarrow" (↓) key and the "tab" key in [1] is eliminated. In Table 1, the "Test Data A" means the patent data used in the first experiment of [1] and reused here. The patent data cover 500 independent claims randomly selected from patents in 2022.

Model	Direction	Previous Ratio (↑)	New Ratio (↑)	Increase (↑)
456M	forward	56.5%	62.7%	10.9%
456M	backward	55.7%	62.1%	11.4%
1.6B	forward	57.0%	63.1%	10.7%
1.6B	backward	56.2%	62.5%	11.2%
6B	forward	57.0%	63.1%	10.7%
6B	backward	56.5%	62.7%	10.9%

Table 1. The improvement of the AE ratio. Target: Test Data A.

4.2. Experiment 2

This experiment implements the mechanism of moving back and forth described in section 3.2. Table 2 shows the experiment results. In this table, the third column "Q1" means the position of the 25%-th token of the input text. The fourth column "Q2" means the position of the 50%-th token of the input text. The fifth column "Q3" means the position of the 75%-th token. The second column defines the direction for calculating the AE ratios. For example, if the input text is tokenized and has 100 tokens, the position "Q1" means that the calculation starts from the 25th token. The "forward" direction in the second column means that the calculation of the AE ratio moves forward to the 26th, 27th, ..., 100th tokens. The 100th token is the end of the input text. After reaching the end, the calculation starts from the 25th token again. The calculation then moves in effect backward to the 24th, 23rd, ..., 1st tokens. The forth-and-back calculation in this example runs over all 100 tokens. In the second column, if the direction is "backward," the calculation will move back first and forth later to run over all 100 tokens. According to Table 2, the AE ratios are similar to one another no matter where the starting position is and no matter which direction to go first. Such a finding indicates that, no matter where a user starts to write, the autocomplete function based on PatentGPT-J models can assist the user to a similar degree and save a similar number of keystrokes. This manuscript hypothesizes that such a finding is not specific to the patent domain and may apply to other generative language models with different training data. This hypothesis can be validated in the future. The "Test Data B" in this experiment contains 1,000 patent claims of CPC Subclass G06N for fine-tuning. These patents are not used in [1].

Model	Direction	Q1	Q2	Q3
456M (fine-tuned)	forward	62.2%	61.7%	61.3%
456M (fine-tuned)	backward	62.4%	62.1%	61.7%
1.6B (fine-tuned)	forward	61.8%	60.9%	60.1%
1.6B (fine-tuned)	backward	61.6%	61.4%	60.9%

Table 2. The AE ratios from different starting positions. Target: Test Data B.

5. Conclusion

Generative language models have great potential to assist humans in writing patent text more effectively. In this manuscript, the way to measure effectiveness is to calculate the ratio of keystrokes saved by the autocomplete function based on model predictions. A higher ratio means more saved keystrokes and fewer manual typing. The ratio was proposed in previous work as the AE ratio. After using a simplified design in this manuscript, the AE ratio increases by more than 10% and reached more than 60%. This means that more than 60% of keystrokes can be saved by the autocomplete function. Furthermore, the models are bidirectional and make it possible to calculate the AE ratio in both directions. The calculation can start anywhere in the text. A key finding is that the AE ratio for the same text remains similar regardless of where the calculation starts. This finding indicates that such bidirectional models can assist a user at a similar level, no matter where

the user starts to write. In addition to the patent domain, the research in this manuscript can be applied to other legal domains in the future because the Transformer architecture in generative models is domain-agnostic.

6. Acknowledgments

The research reported in this manuscript has been funded by the Ministry of Science and Technology (MOST) in Taiwan (Project ID: 111-2222-E-A49-005). In addition, the author would like to thank TensorFlow Research Cloud (TRC) greatly for providing powerful computational resources to make things happen.

References

[1] Lee JS. Evaluating Generative Patent Language Models. arXiv preprint arXiv:220614578. 2022. Available from: https://arxiv.org/abs/2206.14578.

[2] Wang B, Komatsuzaki A. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model; 2021. https://github.com/kingoflolz/mesh-transformer-jax.

[3] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All You Need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17. Red Hook, NY, USA: Curran Associates Inc.; 2017. p. 6000–6010.

[4] Lee JS, Hsiang J. Patent claim generation by fine-tuning OpenAI GPT-2. World Patent Information. 2020;62:101983. Available from: https://www.sciencedirect.com/science/article/pii/S0172219019300766.

[5] Radrof A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language Models are Unsupervised Multitask Learners; 2018.

[6] Lee JS. Controlling Patent Text Generation by Structural Metadata. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management. CIKM '20. New York, NY, USA: Association for Computing Machinery; 2020. p. 3241–3244. Available from: https://doi.org/10.1145/3340531.3418503.

[7] Lee JS. PatentTransformer: A framework for personalized patent claim generation. In: Seventh Doctoral Consortium of JURIX 2019. Madrid, Spain; 2019. Available from: http://ceur-ws.org/Vol-2598/paper-06.pdf.

[8] Lee JS, Hsiang J. Prior Art Search and Reranking for Generated Patent Text. In: Proceedings of the 2nd Workshop on Patent Text Mining and Semantic Technologies (PatentSemTech) 2021, co-located with the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021). PatentSemTech 2021; 2021. p. 18-24. Available from: http://ceur-ws.org/Vol-2909/paper2.pdf.

[9] EleutherAI. pile-uspto; 2020. https://github.com/EleutherAI/pile-uspto.

[10] BigScienceCorpus. pile-uspto; 2022. https://huggingface.co/spaces/bigscience/BigScienceCorpus?source=the_pile_uspto.

[11] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language Models are Few-Shot Learners. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, editors. Advances in Neural Information Processing Systems. vol. 33. Curran Associates, Inc.; 2020. p. 1877-901. Available from: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[12] Lee JS. PatentGPT-J Repository; 2022. https://github.com/jiehsheng/PatentGPT-J.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220467

Transfer Learning for Deontic Rule Classification: The Case Study of the GDPR

Davide LIGA [a,b,1], Monica PALMIRANI [b]
[a] *University of Luxembourg*
[b] *Alma Human-AI, University of Bologna*
ORCiD ID: Davide Liga https://orcid.org/0000-0003-1124-0299, Monica Palmirani https://orcid.org/0000-0002-8557-8084

Abstract. This work focuses on the automatic classification of deontic sentences. It presents a novel Machine Learning approach which combines the power of Transfer Learning with the information provided by two famous LegalXML formats. In particular, different BERT-like neural architectures have been fine-tuned on the downstream task of classifying rules from the European General Data Protection Regulation (GDPR) encoded in Akoma Ntoso and LegalRuleML. This work shows that fine-tuned language models can leverage the information provided in LegalXML documents to achieve automatic classification of deontic sentences and rules.

Keywords. Rule classification, Norms, Legal Knowledge Representation, AI&Law

The ability to automatically detect deontic rules directly from natural language sentences is a crucial long-term goal in the field of Artificial Intelligence and Law (AI&Law), and in legal argumentation [1,2]. One of the obstacles of this kind of tasks is the lack of available data designed *ad hoc* for the classification of deontic rules. Since the annotation of this kind of datasets is time-consuming and requires experts of domain, the process of creating datasets to automatically recognize deontic rules can be costly. Another obstacle, related to the first one, is that datasets might be too small to train Machine Learning classifiers, especially when dealing with deep neural architectures.

To tackle these issues, a groundbreaking methodology has recently been employed in AI (and NLP), namely Transfer Learning, an approach where huge pre-trained neural architectures are employed in downstream tasks. In this regard, BERT is one of the most famous examples, used in many downstream tasks even with very small datasets [3,4].

On the one side, this work shows the potential of using LegalXML documents as source of data. On the other side, it exploits the ability of Transfer Learning to have good performances on downstream tasks even when dealing with small datasets. Furthermore, this work tackles the automatic classification of deontic rules directly from natural language, an AI&Law task which has been approached by the community only marginally. This task consists in classifying single legal sentences or single legal provisions as containing deontic modalities such as Obligations, Prohibitions and Permissions.

[1] Corresponding Author: Davide Liga, davide.liga@uni.lu.

1. Methodology

We consider two methodological aspects: the data extraction method (how we retrieved our data) and the classification method (what Machine Learning approach we used). To extract labelled data for the classification of rules and deontic modalities, we combined Akoma Ntoso and LegalRuleML as suggested in [5]. While LegalRuleML is an optimal representation of the legal logical sphere, Akoma Ntoso is an optimal representation of the structure of legal document, including their natural language. This can facilitate the reconstruction of the atomic legal provisions from natural language sentences, especially in those cases where the deontic information is split in different structural portions within the legal source. In this work, these two formats have been used to create a dataset, where atomic legal provisions are taken (and sometimes reconstructed) from Akoma Ntoso, while the logical/deontic classes are extracted from LegalRuleML.

Regarding the classification methodology, we employed Transfer Learning, which consists in the downstream use of language models (i.e., neural architectures that have been pre-trained on a huge amount of data). Importantly, there are two major ways of performing Transfer Learning: a famous approach is to use the pre-trained language model to extract embeddings to represent our data (as described in [4]). Another approach is that of fine-tuning the pre-trained neural architecture on a downstream task. In this work, we used this second approach.

2. Related Works

The first studies which tackled the classification of deontic elements focused on the deontic elements as parts of a wider range of targets [6,7,8] or were strongly based on symbolic approaches of Artificial Intelligence [9,10]. Perhaps, the first studies which mainly focused on the deontic sphere are [11,12], which are also among the first which employed sub-symbolic methods such as Bi-LSTM and self-attention in the field of AI&Law.

Since the publication of BERT [3], a growing number of studies employed Transfer Learning methods. To the best of our knowledge, the first study which employed BERT for the classification of deontic sentences is [13]. While [12] focused on just prohibitions and obligations, [13] also focused on permissions, achieving an average precision and recall of 90% and 89.66% respectively. Another recent work is [14], which used four pre-trained architectures (BERT, DistilBERT, RoBERTa, and ALBERT) but focused just on the binary detection duties vs non-duties.

Also our work presents a Transfer Learning approach based on BERT (and other similar models), which leverage the symbolic information of LegalXML formats (see also [5]) exploiting the sub-symbolic power provided by different pre-trained language models. The novelty and the power of Transfer Learning methodologies jointly with the combined use of Akoma Ntoso and LegalRuleML are two major contributions of our study, along with the design of the experimental settings in 4 different classification scenarios: (1) Rule vs Non-rule, (2) Deontic vs Non-deontic, (3) Obligation vs Permission vs None, (4) Obligation vs Permission vs Constitutive Rule vs None. Another point which is worth mentioning is that LegalXML formats such as Akoma Ntoso and LegalRuleML are documents which are written by legal experts, providing the machine learning algorithm with high quality data.

3. Data

The data used in this study consists of 707 atomic normative provisions[2] extracted from the European General Data Protection Regulation (GDPR). To extrapolate this dataset, we used the *DAta Protection REgulation COmpliance* (DAPRECO) Knowledge Base[15], which is the LegalRuleML representation of the GDPR and the the biggest knowledge base in LegalRuleML [16], as well as the biggest knowledge base formalized in Input/Output Logic [17]. The current version of the DAPRECO[3] includes 966 formulæ in reified Input/Output logic: 271 obligations, 76 permissions, and 619 constitutive rules. As explained in [15], the number of constitutive rules is much higher than permissions and obligations because constitutive rules are needed to trigger special inferences for the modelled rules. This means that constitutive rules are an indicator of the existence of a rule, without giving information about deontic modalities.

Importantly, DAPRECO also contains the connections between each formula and the corresponding structural element (paragraphs, point, etc) in the Akoma Ntoso representation of the GDPR[4]. In other words, using a LegalRuleML knowledge base like DAPRECO and the corresponding Akoma Ntoso representation, it is possible to connect the logical-deontic sphere of legal documents (in this case the 966 Input/Output formulæ provided by DAPRECO) to the natural language statements in the legal text (provided by the Akoma Ntoso representation of the GDPR).

Importantly, this combination of Akoma Ntoso and LegalRuleML facilitate also the reconstruction of the exact target in terms of natural language. For example, many obligations of legal texts are split into lists, and Akoma Ntoso is useful to reconstruct those pieces of natural language into a unique sentence.

4. Experiment settings and results

At the end of the process of extraction, we achieved a total of 707 labelled provisions, which have been reconstructed whenever they were split into lists (thanks to the structural information provided by Akoma Ntoso). The labels of these sentences are the same as those provided by DAPRECO with the addition of a "none" category. We abbreviated "obligationRule", "permissionRule", "constitutiveRule" in "obligation", "permission" and "constitutive" respectively.

The class "obligation" is referred to those sentences which have at least one obligation rule in their related formulæ. The class "permission" is referred to those sentences which have at least one permission rule in their related formulæ. The class "constitutive" is referred to those sentences which just constitutive rules in their related formulæ. The class "none" is referred to all sentences which have no rule at all. These labels allowed 4 different experimental settings, as shown in Table 1:

Scenario 1 is a binary classification and aims at discriminating between rule and non-rule instances. In this scenario, all labels other than "none" are considered rule, while

[2]These provisions belong to the body of the GDPR (preamble and conclusions were excluded), and are generally paragraphs or list points, which may sometimes consist of multiple sentences.

[3]The DAPRECO knowledge base can be freely downloaded from: https://github.com/dapreco/daprecokb/blob/master/gdpr/rioKB_GDPR.xml.

[4]The Akoma Ntoso representation of the GDPR can be currently accessed from https://github.com/guerret/lu.uni.dapreco.parser/blob/master/resources/akn-act-gdpr-full.xml.

Table 1. Number of instances per class per scenario.

	Classes	Instances
Scenario 1	rule	260
	non-rule	447

	Classes	Instances
Scenario 2	deontic	204
	non-deontic	503

	Classes	Instances
Scenario 3	obligation	156
	permission	44
	none	503

	Classes	Instances
Scenario 4	obligation	156
	permission	44
	constitutive	56
	none	447

Table 2. Results for the two stratified baselines applied to scenario 1, 2, 3 and 4. Within the brackets, the number of instances is reported. P = precision; R = recall, F1 = F1-score; Acc = Accuracy; Mcr = Macro F1.

Baseline Scenario 1

	P	R	F1	Acc	Mcr
rule(39)	.37	.33	.35	.55	.50
non-rule(67)	.63	.67	.65		

Baseline Scenario 2

	P	R	F1	Acc	Mcr
deontic(30)	.23	.23	.23	.57	.47
non-deontic(76)	.70	.70	.70		

Baseline Scenario 3

	P	R	F1	Acc	Mcr
obligation(24)	.24	.17	.20	.60	.40
permission(6)	.20	.33	.25		
none(75)	.73	.76	.75		

Baseline Scenario 4

	P	R	F1	Acc	Mcr
obligation(23)	.16	.13	.14	.44	.23
permission(7)	.20	.14	.17		
constitutive(8)	.00	.00	.00		
none(67)	.58	.63	.60		

"non-rule" is just an alias for "none". Scenario 2 focus on a binary classification between deontic instances (i.e., any sentence labelled as either "obligation" or "permission") and non-deontic instances (i.e., all instances which are labelled neither as "obligation" nor as "permission"). Scenario 3 is a multiclassification which considers the classes "obligation", "permission" and "none" (with "constitutive" considered as part of the latter). Scenario 4 is a multiclassification which considers the classes "obligation", "permission", "constitutive" and "none". For the multi-classifications (i.e. Scenario 3 and 4) four statements have been removed, since the classes "obligation" and "permission" overlapped.

To assess the non-triviality of our experiments, we employed different kinds of baseline, showing their difficulty in performing each classification task. In Table 2, we reported the results of a "stratified" baseline, which was the better performing among the baseline methods[5]. As can be seen from Table 2, we applied the "stratified" baseline on all the scenarios achieving quite low performances on all of them.

As far as the experimental settings are concerned, the dataset was divided into 70% for the training phase, 15% for the test and 15% for the validation; and for all instances, a max length of 30 was applied.

Regarding the Transfer Learning architecture, 3 pre-trained language model have been fine-tuned, namely BERT [3], DistilBERT [18], and the LegalBert trained on EurLex [19]. These three neural architectures were fine-tuned by adding two linear layers with a ReLu activation function and with a dropout of 0.2 after each activation, and a final output layer was added for the classification, through a softmax activation function. The

[5]The so-called "stratified" baselines can reach higher scores because they reflect the class distribution.

Table 3. Results of the 4 scenarios. P = precision; R = recall, F1 = F1-score, S/T = Support/Total ratio.

Scen.	Classes	BERT			DistilBERT			LegalBERT			S/T
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	
1	rule	.74	.95	.83	.77	.95	.85	.75	.77	.76	39/260
	non-rule	.96	.81	.88	.97	.84	.90	.87	.85	.86	68/447
		Accuracy .86			Accuracy .88			Accuracy .82			Total: 107/707
		Macro avg .86			Macro avg .87			Macro avg .81			
		Weight. avg .86			Weight. avg .88			Weight. avg .82			
2	deontic	.74	.90	.81	.82	.90	.86	.80	.77	.79	31/200
	non-deontic	.96	.87	.91	.96	.92	.94	.91	.92	.92	76/507
		Accuracy .88			Accuracy .92			Accuracy .88			Total: 107/707
		Macro avg .86			Macro avg .90			Macro avg .85			
		Weight. avg .88			Weight. avg .92			Weight. avg .88			
3	obligationRule	.74	.83	.78	.74	.83	.78	.63	.92	.75	24/156
	permissionRule	.50	.83	.62	.36	.67	.47	.56	.83	.67	6/44
	none	.97	.88	.92	.94	.84	.89	1.0	.82	.90	76/503
		Accuracy .87			Accuracy .83			Accuracy .84			Total: 106/703
		Macro avg .78			Macro avg .71			Macro avg .77			
		Weight. avg .88			Weight. avg .84			Weight. avg .85			
4	obligationRule	.70	.79	.75	.80	.83	.82	.84	.67	.74	24/156
	permissionRule	.60	.50	.55	.40	.67	.50	.17	.67	.28	6/44
	constitutiveRule	.36	1.0	.53	.47	.89	.62	.89	.89	.89	9/56
	none	1.0	.73	.84	.94	.76	.84	.96	.79	.87	67/447
		Accuracy .75			Accuracy .78			Accuracy .76			Total: 106/703
		Macro avg .67			Macro avg .69			Macro avg .69			
		Weight. avg .78			Weight. avg .80			Weight. avg .81			

fine-tuning process of these 3 neural architectures was performed in 10 epochs (learning rate: 1e-3; batch size: 32).

The final results on the validation set are reported in Table 3, where it can be seen that DistilBERT outperforms the other classifiers in the binary classifications, with an average score reaching .88 in the first scenario and .92 in the second one.

The results for the third and fourth scenarios are less straightforward and show that BERT slightly outperforms other classifiers in the third scenario, while LegalBERT outperformed the other models in the fourth scenario. The main problem for the multiclassifications, is the class unbalance and the restricted amount of instances for some classes. In spite of this, scores are encouraging, especially considering the small amount of data.

5. Conclusions

The contribution of this work is showing how Transfer Learning methods can leverage the information provided in LegalXML to train classifiers capable of automatically classifying deontic sentences and rules. In the future, we would like to create a stronger connection with the ontological sphere by using PrOnto [20], strengthening this hybrid AI approach, which combines symbolic knowledge with sub-symbolic methods.

In this work, we were not targeting in the internal elements of the logical formulæ, we just addressed the ontological classes of each rule. However, in the future we want to create classifiers that directly address the internal components of each rule, trying to find a match between portions of natural language and portions of rules. In general, the ability to connect each internal component (or at least some) of the deontic formulæ contained in DAPRECO directly to the portion of natural language where the component is communicated or expressed is a crucial future direction, and an important step towards the long-term goal of filling the gap between natural language and the logical-inferential sphere, which would generate a more reliable and explainable Artificial Intelligence.

References

[1] Ashley KD. Artificial intelligence and legal analytics: new tools for law practice in the digital age. 2017.
[2] Wyner A, Peters W. On rule extraction from regulations. 2011:113-22.
[3] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.
[4] Liga D, Palmirani M; 20. Transfer Learning with Sentence Embeddings for Argumentative Evidence Classification. Proceedings of the 20th Workshop on Computational Models of Natural Argument. 2020.
[5] Liga D, Palmirani M. Deontic Sentence Classification Using Tree Kernel Classifiers. 2023:54–73.
[6] Kiyavitskaya N, Zeni N, Breaux TD, Antón AI, Cordy JR, Mich L, et al.; Springer. Automating the extraction of rights and obligations for regulatory compliance. 2008:154-68.
[7] Waltl B, Muhr J, Glaser I, Bonczek G, Scepankova E, Matthes F. Classifying Legal Norms with Active Machine Learning. 2017:11-20.
[8] Gao X, Singh MP. Extracting normative relationships from business contracts. 2014:101-8.
[9] Dragoni M, Villata S, Rizzi W, Governatori G. Combining Natural Language Processing Approaches for Rule Extraction from Legal Documents. 2018:287–300.
[10] de Maat E, Winkels R. Automated Classification of Norms in Sources of Law. 2010:170–191.
[11] Neill JO, Buitelaar P, Robin C, Brien LO. Classifying sentential modality in legal language: a use case in financial regulations, acts and directives. 2017:159-68.
[12] Chalkidis I, Androutsopoulos I, Michos A. Obligation and prohibition extraction using hierarchical rnns. arXiv preprint arXiv:180503871. 2018.
[13] Joshi V, Anish PR, Ghaisas S. Domain adaptation for an automated classification of deontic modalities in software engineering contracts. 2021:1275-80.
[14] Shaghaghian S, Feng LY, Jafarpour B, Pogrebnyakov N; IEEE. Customizing Contextualized Language Models for Legal Document Reviews. 2020:2139-48.
[15] Robaldo L, Bartolini C, Palmirani M, Rossi A, Martoni M, Lenzini G. Formalizing GDPR provisions in reified I/O logic: the DAPRECO knowledge base. Journal of Logic, Language and Information. 2020;29(4):401-49.
[16] Athan T, Boley H, Governatori G, Palmirani M, Paschke A, Wyner A. Oasis legalruleml. In: Proceedings of the Fourteenth International Conference on AI and Law; 2013. p. 3-12.
[17] Makinson D, Van Der Torre L. Input/output logics. Journal of philosophical logic. 2000;29(4):383-408.
[18] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:191001108. 2019.
[19] Chalkidis I, Fergadiotis M, Malakasiotis P, Aletras N, Androutsopoulos I. LEGAL-BERT: The muppets straight out of law school. arXiv preprint arXiv:201002559. 2020.
[20] Palmirani M, Martoni M, Rossi A, Bartolini C, Robaldo L. Pronto: Privacy ontology for legal compliance. 2018:142-51.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220468

Functional Classification of Statements of Chinese Judgment Documents of Civil Cases

Chao-Lin Liu[†] Hong-Ren Lin[‡] Wei-Zhi Liu[⁋] Chieh Yang
National Chengchi University, Taiwan
{chaolin[†],109753156[‡],109753157[⁋]}@nccu.edu.tw

Abstract. Enabling the inference systems for assisting legal decisions to identify the functions of sentences and paragraphs in documents of legal judgments can enhance the justifiability of their algorithmic recommendations. The information about the functions of larger linguistic constituents complements the information at the word level like NER, and provides more clues about the arguments for the legal decisions. We explore this venue for the civil cases, which is a relatively uncommon choice in legal informatics and more challenging than working on the criminal cases. Current experimental results are promising.

Keywords: civil cases, functional classification, semantic classification, machine learning, justification

1 Introduction

There is a long history of the applications of artificial intelligence (AI) to the legal domain, and the first AI and law conference was held in 1987. In contrast, the field has attracted a large-scale attention of the Chinese community only until recently (e.g., [7] [13]). In this short manuscript, we are unable to review the field, but to focus on our research topic.

Offering both the recommendations and supportive justifications for the legal queries is important for people to accept the algorithmic recommendations. Take the task of the legal judgment prediction (LJP) for example. The LJP task aims at predicting the outcomes of lawsuits (e.g., [4][8]). The majority of the current work is about criminal cases, and the goals are to predict the penalties against the defendants. So far, work on civil cases like [6] is relatively uncommon. When addressing the applications of machine learning techniques to the prediction tasks, Mumford et al. [9] commented that "*without an explanation of why the case was so classified, the adjudicator has no reason to follow*."

Evidences also show that enabling computers to read and understand some details in judgment documents appears to be necessary for improving the quality of legal judgment predictions [3]. It may be relatively easy to determine the types of charges and the citing statutes for criminal cases, but the quality of the predictions for the penalties remains to be expected [4].

Hence, techniques for strengthening computers' competence to understand and explain details in judgment documents are needed for both the acceptance and the quality of the prediction systems [1].

Detailed information can refer to such word level information as named entities [3]. In our work, we focus on the functions of the sentences in legal documents. The concept is similar and related to the research that hopes to find the legal arguments in judgement documents ([2][11][12]).

Information about the functions of sentences in judgment documents complements the information of the word or phrase levels, and may be useful for explaining the algorithmic recommendations of AI systems.

2 Problem Definition

The judgment documents that were published by the Judicial Yuan in Taiwan do not follow a very strict format, although the documents typically contain some common sections. Some sections provide the meta data of a case, such as the court name, the time of the judgement, and the summary of the judgments of the case. The central part of a judgment document is the section that records the facts and the information about the reasoning for reaching the judgments, e.g., the criminal activities for criminal cases, the conflicts in interests for the involving parties for civil cases, the opinions of the court, and various considerations for reaching judgments (subsuming).

We are working on the alimony problems, which is a specific type of the civil cases. Our first goal is to classify the sentences of the central part of a judgment document into **four categories** of functions, namely, the pleadings of the applicants, the responses of the opposite parties, the opinions of the court, and the subsuming part. We denote these four categories as **C1**, **C2**, **C3**, and **C4**, respectively. These are the most central factors for judging civil cases.

We will extend our classifiers to categorize the sentences into **five categories**, and add the category of "conflicts", denoted by **C5**, to indicate the disagreeing items between the applicants and the opposite parties.

Since sentences of C5 are about the conflicts of the two sides of the disputes, it was easy to confuse sentences of C1 and C2 with C5. In addition, the courts often mention the conflicts in reaching decisions, so it may not be easy to tell specific sentences of C4 and C5 apart.

3 Data Source and Preprocessing

The main source of data for this research is the open data websites of the Judicial Yuan of Taiwan. The websites provide judgment documents of three levels of courts, i.e., the district, high, and supreme courts, and some special types of courts, e.g., for business and intellectual issues, for juvenile and family issues, and for administrative issues. Due to certain privacy and protection considerations, not all of the judgements of the courts are released, and few of the previously published documents may be retracted due to a wide variety of legal reasons. Therefore, the total number of published documents is stable only within a reasonable range.

We may access the data via the batch site or the interactive site.[1] For computational efficiency, we downloaded the documents from the batch site and will refer to it as TWJY, henceforth. TWJY updates the corpus monthly. At the time of this writing, we can find about 17.6 million documents for all cases that were judged since January 1996 in TWJY.

Although the total number of published documents in TWJY is huge, these public documents belong to a myriad of types of lawsuits. The number of cases that is related to a specific category may not be large, and we have to search the TWJY data to identify the documents that are at least seemingly relevant to our research needs. We extracted 6679 cases of which the "cause of judgment" (裁判案由) were for "the issues of alimony"

[1] https://opendata.judicial.gov.tw/ and https://law.judicial.gov.tw/FJUD/default_AD.aspx

(給付扶養費). These cases were judged between January 2000 and December 2021, and Figure 1 depicts the trends of the annual numbers of the extracted cases.

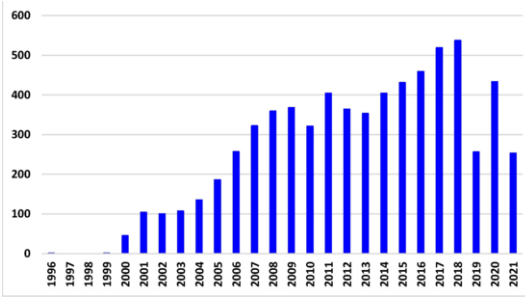


Figure 1. Annual number of cases.

We further filter the files for more detailed reasons. We counted the number of main paragraphs in the judgment documents of these 6679 cases, and Table 1 offers the counts of the number of main paragraphs in the documents. We inspected the paragraphs, and found that the cases with only one, two, or three main paragraphs are related to relatively simple and stereotypical problems, and are not relevant to our current studies. The cases with only three main paragraphs are commonly related to specific types of legal issues, such as that the applicants did not pay for the legal fees or that the litigants requested to transfer the cases to other courts. There were 11 cases which had more than 10 main paragraphs. We did not use these rare cases in our experiments.

Par.	Counts	Par.	Counts
1	2021	6	452
2	500	7	226
3	1541	8	106
4	1073	9	40
5	677	10	32

Table 1. Dist. of the number of paragraphs.

We use the cases with three or four main paragraphs in most of our experiments, and will use cases with six paragraphs also. When we examined the contents of the judgment documents more closely, we could find cases that are not appropriate for our study due to the main issues of the cases, so we could use only 820 cases. This small number of cases can be surprising and even frustrating, particularly when one compares this number with the size of TWJY. However, this is a fact that one can verify with professors of law [5][14].

We offer two possible reasons for this relative few number of cases. The first is that, in Taiwan and in Asian cultures, resolving family problems in courts can be a shameful problem for families, so, unless the problems are really not resolvable in private, the number of lawsuits for issues of alimony is suppressed for cultural reasons. The second is that, even after the cases have been submitted to the courts, the legal procedure encourages family members to resolve their problems privately via the mediation process that is assisted by the courts. This face-saving alternative can contribute to the reduction of the number of cases that have to be judged by the courts.

The definition of "sentence" in Chinese is much vaguer than that in English, although there is a symbol for sentence period in Chinese punctuation. In short, we segmented texts by the comma ("，"), the period ("。"), the semicolon ("；"), and the quotation marks. From these cases, we had 114204 sentences.

4 Labeling the Data

We can label the categories of our data in two different ways. The most common way is to ask domain professionals to categorize the individual statements. We refer to this dataset as **DE**, where "D" and "E" denote "**D**omain" and "**E**xpert", respectively. We used 80% and 20% of DE for training and testing, respectively. We refer to these subsets as **DEA** and **DEE**, respectively, where the ending "A" and "E" are for "tr**A**in" and "t**E**st", respectively.

The main annotator owns a college degree in law, and we have only limited data for inter-rater agreement at this stage.

An intriguing alternative is to take advantage of the regularities of the writing styles in the judgment documents. It is possible to find common patterns of collocations that are indicative of the high-level functions of the paragraphs. Hence, we relied on specific keywords, phrases, and collocations to algorithmically label the paragraphs to bootstrap our classification tasks, and treated all sentences in a labeled paragraph to belong to the category of the paragraph.

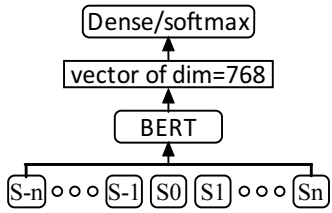


Figure 2. An intuitive classifier with BERT.

Paragraphs for categories C1 and C2 usually begin with "**聲請人**" and "**相對人**", respectively. It is very common for the courts to use "按" in the beginning statements in paragraphs of category C3. The collocations that follow the regular expression "**(本|法)院 .* (判斷|心證|經查|據)**" strongly suggests paragraphs of category C4. We refer to this algorithmically labeled dataset as **KP**, where "K" and "P" denote "**K**eywords" and "**P**atterns", respectively. Again, we used 80% and 20% of KP for training and testing, respectively. We will refer to these subsets of data as **KPA** and **KPE**, respectively, for analogous naming principle.

5 The Classifiers and Settings

In this section, we report results of the applications of machine learning methods to legal informatics. We applied established tools, including those offered by scikit-learn[2] and TensorFlow[3]. More specifically, we employed the tools for TFIDF vectorizer, logistic regression, and support vector machines of scikit-learn. We used the pretrained BERT for Chinese, including its tuning, that was demonstrated in the interface in TensorFlow Hub. Unless stated explicitly, we used the default parameters for the tools in the experiments reported in this short manuscript.

We have explored many different network structures for this study, including some intuitive ones and more academic options (e.g., [10]). We chose to report the results of the intuitive network that is shown in Figure 2. When training this model, we used the class label of a target sentence, S0, for the context of S0, where the context contains ***n*** sentences on both sides of S0. We made the BERT itself trainable, set the batch size to 64, and set the patience to stop training to five. In this report, we set *n* to be 1, 2, 3, 4, and 5.

In the experiments, 80% of the training data were used for validation. Hence, we had more than 70000, 18000, and 23000 instances for training, validation, and testing, respectively, but the exact numbers of the instances may vary slightly due to the settings of the environments.

6 Sentence Classification with DE

In this section, we report results of using **DEA** and **DEE** for training and testing, respective.

[2] https://scikit-learn.org/stable/

[3] https://tfhub.dev/tensorflow/bert_zh_L-12_H-768_A-12/4

6.1 *Classification of Four Categories*

Table 2 shows the confusion matrix of a DEA-DEE experiment for a four-category classification, and n was 4. The middle four rows of Table 3 list the F_1 values for these four categories and different values of n, while the last row lists the values of the macro F_1 for different values of n.

In Table 2, we can see that it was relatively difficult to distinguish sentences of C1 and C2 because their similarity in nature. Analogously, it was not easy to differentiate sentence of C3 and C4. One might have expected that, the quality of classification might improve as we expend the widths of the contexts, and this is supported partially by the statistics in Table 3. However, as we enlarged the contexts, the classifiers might be confused by some misleading statements in far-away contexts.

	categorized			
actual	C1	C2	C3	C4
C1	3402	198	7	623
C2	460	1667	56	838
C3	6	3	4330	298
C4	322	252	369	10962

Table 2. DEA-DEE, four categories and $n = 4$.

n	1	2	3	4	5
C1	0.717	0.785	0.760	0.808	0.747
C2	0.548	0.638	0.612	0.649	0.569
C3	0.917	0.910	0.897	0.921	0.897
C4	0.858	0.879	0.872	0.890	0.881
MF	0.760	0.803	0.785	0.817	0.773

Table 3. DEA-DEE, F_1 and macro F_1

	categorized				
actual	C1	C2	C3	C4	C5
C1	3673	351	7	189	55
C2	675	1403	38	322	102
C3	6	1	4482	103	46
C4	704	594	453	6635	1022
C5	196	23	20	491	1686

Table 4. DEA-DEE, **five** categories and $n = 4$.

6.2 *Classification of Five Categories*

Table 4 shows the confusion matrix of a DEA-DEE experiment for five-category classification, and n was 4. The statistics supported our explanation and expectation which we stated at the end of Section 2. The F_1 values of C1, C2, and C4 dropped to 0.771, 0.571, and 0.774, respectively; all are smaller than their counterparts in Table 3. The F_1 value of C3 was 0.930 and relatively stable. The F_1 value of C5 was only 0.633.

7 Training the Classifiers with KPA

In this section, we report results of using KPA to train the classifiers, and test the classifiers with DEE and KPE. Recall that the dataset KP was labeled by heuristic rules, and we did not have rules for C5, so the experiments reported in this section can involve only four categories.

7.1 *Testing with DEE*

Table 5 shows the confusion matrix of a KPA-DEE experiment for a four-category classification, and n was 4. The middle four rows of Table 6 list the F_1 values for these four categories and different values of n, while the last row lists the values of the macro F_1 for different values of n.

The statistics in Table 5 indicate similar trends as those indicated by Table 2. It was relatively difficult to distinguish sentences of C1 and C2, and it was not easy to differentiate sentence of C3 and C4. Statistics in Table 6 also support that expanding the widths of contexts benefited the F_1 to an extent.

It is interesting to compare the statistics in Tables 3 and 6. Training the classifiers with the DEA, which is annotated by domain professionals, offered better classification results than training the classifiers with heuristically labeled data across the board. The average gain in macro F_1 is above 0.05.

	categorized			
actual	C1	C2	C3	C4
C1	3221	228	47	734
C2	574	1633	69	745
C3	4	57	4150	426
C4	276	433	1070	10126

Table 5. KPA-DEE, four categories and $n = 4$.

n	1	2	3	4	5
C1	0.705	0.703	0.688	0.776	0.700
C2	0.538	0.578	0.383	0.608	0.624
C3	0.834	0.810	0.797	0.832	0.802
C4	0.827	0.816	0.775	0.846	0.752
MF	0.726	0.727	0.661	0.765	0.719

Table 6. KPA-DEE, F_1 and macro F_1.

n	1	2	3	4	5
DEA	0.675	0.710	0.697	0.720	0.679
KPA	0.675	0.697	0.621	0.721	0.662

Table 7. Testing with KPE, macro F_1.

7.2 *Testing with KPE*

Table 7 shows only the values of the macro F_1 when we tested two classifiers that were trained by DEA and by KPA and test both classifiers with KPE. Since we already had human annotated data, this comparison is only of theoretical interest. Even if the KPA and the KPE datasets were labeled with the same principles, the statistics in Table 7 indicate that using DEA to train can still help us achieve slightly better qualities than when we train the classifiers with KPA.

8 Concluding Remarks

Justifications are required for algorithmic recommendations for legal decisions. We compared effects of different ways to annotate the raw data, and evaluated a few classification models for categorizing the legal functions of the sentences and paragraphs in judgment documents of the civil cases, and the current results are promising. We have conducted more experiments, and can report their results during the Conference.

Acknowledgements

This research was supported in part by the project 110-2221-E-004-008-MY3 of the National Science and Technology Council of Taiwan. We are very thankful to the reviewers' comments.

References

[1] Atkinson K, Bench-Capon T, and Bollegala D. Explanation in AI and law: Past, present and future. *Artificial Intelligence*. 2020; 289:103387.

[2] Aumiller D, Almasian S, Lackner S, and Gertz M. Structural text segmentation of legal documents. *Proc. of the 18th Int'l Conf. on Artificial Intelligence and Law*. 2021; p. 2–11.

[3] Chen Y, Sun Y, Yang Z, and Lin H. Join entity and relation extraction for legal documents with legal feature enhancement. *Proc. of the 28th Int'l Conf. on Computational Linguistics*. 2020; p. 1561–1571.

[4] Feng Y, Li C, and Ng V. Legal judgment prediction via event extraction with constraints. *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics (vol. 1)*. 2022. p. 648–664.

[5] Huang S-C. An empirical study of the legal judgments of alimony for elderly parents in Taiwan. presented in the Conf. on Technologies for Law and Access to Justice. 25 April 2022.

[6] Huang S-C, Shao H-L, and Leflar RB. Applying decision tree analysis to family court decisions: factors determining child custody in Taiwan. *Proc. of the 18th Int'l Conf. on Artificial Intelligence and Law*. 2021. p. 258–259.
[7] Liu C-L, Chang C-T, and Ho J-H. Classification and clustering for case-based criminal summary judgments, *Proc. of the Ninth Int'l Conf. on Artificial Intelligence and Law*. 2003; p. 252–261.
[8] Long S, Tu C, Liu Z, and Sun M. Automatic judgment prediction via legal reading comprehension. *Proc. of the 2019 China National Conf. on Chinese Computational Linguistics*. 2019; p. 558–572.
[9] Mumford J, Atkinson K, and Bench-Capon T. Machine learning and legal argument. *Proc. of the 21st Workshop on Computational Models of Natural Argument*. 2021; p. 47–56.
[10] Rao G, Huang W, Feng Z, and Cong Q. LSTM with sentence representations for document-level sentiment classification. *Neurocomputing*. 2018; 308:49–57.
[11] Stab C and Gurevych I. Parsing argumentation structures in persuasive essays. *Computational Linguistics*. 2017; 43(3):619–659.
[12] Wyner A, Mochales-Palau R, Moens M-F, and Milward D. Approaches to text mining arguments from legal cases. In *Semantic Processing of Legal Texts*. 2010; p. 60–79.
[13] Xiao C, Zhong H, Guo Z, Tu C, Liu Z, Sun M, Feng Y, Han X, Hu Z, Wang H, and Xu J. CAIL2018: A large-scale legal dataset for judgment prediction. 2018; arXiv:1807.02478.
[14] Xu H, Savelka J, and Ashley KD. Accounting for sentence position and legal domain sentence embedding in learning to classify case sentences. *Proc. of the 34th Int'l Conf. on Legal Knowledge and Information Systems*. 2021; p. 33–42.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220469

An Argumentation and Ontology Based Legal Support System for AI Vehicle Design

Yiwei LU [a,1,2], Zhe YU [b,3], Yuhui LIN [c], Burkhard SCHAFER [a], Andrew IRELAND [c], Lachlan URQUHART [a]

[a] *School of Law, Old College, University of Edinburgh*
[b] *Institute of Logic and Cognition, Department of Philosophy, Sun Yat-sen University*
[c] *School of Mathematical and Computer Sciences, Heriot-Watt University*
e-mail: Y.Lu-104@sms.ed.ac.uk, zheyusep@foxmail.com, {B.Schafer, lachlan.urquhart}@ed.ac.uk, {y.lin,ceeai}@hw.ac.uk

Abstract. As AI products continue to evolve, increasingly legal problems are emerging for the engineers that design them. Current laws are often ambiguous, inconsistent or undefined when it comes to technologies that make use of AI. Engineers would benefit from decision support tools that provide engineer's with legal advice and guidance on their design decisions. This research aims at exploring a new representation of legal ontology by importing argumentation theory and constructing a trustworthy legal decision system. While the ideas are generally applicable to AI products, our initial focus has been on Autonomous Vehicles (AVs).

Keywords. Legal ontology, Autonomous vehicle, Legal detection, Argumentation theory, Explainable AI

1. Introduction

Concerns about the safety of AVs, and a recognition that widespread lack of trust in them will impede their uptake, have resulted in a plethora of legislative and regulatory activity such as the EU AI Act [1], or, specifically for AVs, the proposal by the Law Societies of England and Wales. Our contention is that legal AI, and more specifically a combination of legal ontologies and argumentation systems, can help alleviate these compliance burdens by providing intelligent design environments that help the engineer to reason through the legal implications of their design choices.

A similar approach has been developed as part of the Smarter Privacy project that aims at assisting developers of smart grids to comply with data protection law [2]. Their approach modelled the subject domain using the Sumo ontology, enriched with concepts from data protection law, and combined it with a rule-based reasoner about the relevant

[1]Work supported by UKRI Research Node on Trustworthy Autonomous Systems Governance and Regulation (EP/V026607/1, 2020-2024).

[3]Corresponding Author.

legal domain. While sympathetic to this approach, our proposal differs in a crucial jurisprudential assumption that we think implicitly underlies their solution: there the law and its categories are taken as a given, and the reasoner then merely subsumes new facts under the old categories. The result is a "Dworkinian one-right-answer" [3] approach. By contrast, we argue that the legal analysis of new technologies takes place under uncertainty not just of the facts but also the law, whose categories can become unstable in response to external change, contested and open to revision.

Our approach, in a nutshell, is this: the introduction of AVs and other autonomous systems creates fundamental challenges to the legal system that can no longer be resolved by mere analogous application of existing categories to these new objects. Rather, they potentially "break" the underlying ontology and conceptual divisions of the law, creating systematic inconsistencies and gaps, which are then in need of "ontology repair". Because law, like language, is self-reflective, this process of ontology repair in turn uses legal arguments in one and the same decision, the judge may e.g. propose an interpretation that subsumes the facts under an existing legal category, while also making an argument that some higher-order legal principle requires to be amended, deleted or added to the existing categories. This ability of lawyers to reason *about* legal categories in addition to *using* them is particularly visible when more fundamental changes in the external world create gaps, inconsistencies and ambiguities when old categories are applied to new realities. These exercises in ontology repair and ontology evolution inevitably create legal uncertainty. As we will see in the examples below, this can create an unmet need for engineers (or other members of society) to make legally informed decisions under uncertainty. We aim to show how building on existing approaches to legal ontologies and legal reasoning that make ontology repair explicit can help to address this.

A simple example may help to explain this more abstract notion. The United Kingdom Department of Transport 2015 report *The Pathway to Driverless* Cars stated that - testing of automated technologies is legally possible, provided that *the vehicle can be used compatibly with road traffic law*. In other words, the AV must observe the same rules originally addressed to human drivers. How can a developer of an AV make sense of this requirement? A starting point would be to consult the relevant road traffic rules, and treat the AV as the new norm addressee that "inherits" the legal obligations of the human driver. For some of these, this change is unproblematic and merely reinterprets the old category of "driver" as including "autonomous vehicles". For other rules, however, this strategy is less convincing. A candidate could be the rule that "the driver must not be drunk". Here the engineer can continue to treat the AV as "the driver", and as cars are never drunk, the conditional norm: "If drunk, don't drive" is trivially true all the time, and the car is trivially compliant with this provision. Alternatively, "the driver" in this context may refer to some human inside the car who may have been assigned specific legal duties like healing the injury in accidents. This means the concept of "driver" has now been subdivided to "heal" the counter-intuitive outcome. If this interpretation is taken, a number of follow-up questions need to be answered. In one interpretation, this human "non-driver" is responsible for being sober and faces sanctions when drunk but this is not a concern of the car developer. However, another interpretation is also possible: here, the duty to ensure that the vehicle is operated lawfully transfers more fully to the AV, which now has to monitor if there is at least one sober passenger, e.g. [4].

The underlying problem that leads to these three different interpretations is that AVs share some properties with the category "driver" and some properties with the disjoint

category "car", creating systematic ambiguities when interpreting laws whose semantics reflect the old ontology. Even more fundamentally, the reason AI regulation is difficult is that they seem to violate some of the most basic ontological distinctions that structure the law, in particular the distinction between persons and objects. This was a central point made by the joint report of the Law Commissions of England and Scotland. They note that "Existing law reflects a division between rules governing vehicle design on the one hand and the behaviour of drivers on the other." What the Commissions ask for in response is a new conceptual scheme that bridges these two regulatory spheres: the automated driving system is at the same time equipment fitted in a vehicle (and object) but it also determines the behaviour of the vehicle (an agent). We can repurpose research in ontology repair to model not only the reasoning of the Commission, but also how that document can in turn be made into a legal argument that informs design decisions. Ideally, at every decision that they have to document, the engineers need a system that

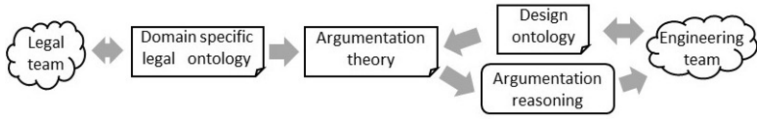


Figure 1. Overview of the process of legal support system

(1) Gives feedback about whether a design draft is in compliance with current or possible laws, depending on which of several competing interpretations is chosen, (2) Answers what happens for the legal analysis if a single functionality is added, deleted or modified in current design draft; (3) Supports reasoning based on how conflicting preferences and values have been resolved; (4) Gives an understandable explanation of the legal results for auditing purposes. Here, we present a legal support system for autonomous vehicles (*LeSAC*), as shown in Figure 1. It is built on top of legal ontology and a legal reasoning based legal argumentation framework, i.e. *L-ASPIC* [5], adapted from [6].

Legal ontologies have proved to be very strong tool for law as legal expert systems, legal databases and documents management: There are different functions that have been proposed, e.g. the Legal Knowledge Interchange Format (LKIF) Core ontology builds on the Web Ontology Language (OWL) and rules [7]; LRI-Core aimed at the legal domain grounded in common-sense [8] and UOL [9]. And there are legal ontology models constructed for specific legal domains like ALLOT[10] etc.

However, a legal ontology alone is insufficient for legal reasoning. The main description language of the current legal ontology is the Web Ontology Language (OWL) or OWL2, whose semantics are based on DL [11] and they cannot support inconsistent reasoning as a subset of first-order logic, which is quite important in legal reasoning. This is also a reason that most of the existing legal ontology focuses more on capturing abstract legal concepts, playing the role of document management or legal dictionary. To address this problem, there are works focused on detecting and repairing inconsistent parts [12] or extending classical logic by adding true values [13]. However, these works weaken the reasoning strength of DL [13] and require guidance outside the ontology [12]. Also, they lack explainability which is a desirable feature when inconsistency happens.

There are other works taking a formal argumentation approach. Formal argumentation has been noticed as an approach to dealing with reasoning under inconsistent and uncertain contexts [14,15]. Formal argumentation has the merits of computational efficiency and explainability [16,17]. For handling reasoning with inconsistent ontologies,

several previous studies have considered applying structured argumentation systems in this field, e.g. [15,18]. These works support inconsistent reasoning but they cannot describe more complicated interactions or agents' different attitudes. And they also do not discuss the explanation of reasoning results or design the legal semantics and functions in their reasoning processes.

The rest of this paper is organized as follows. §2 firstly introduce a running example, as well as an introduction of DL and argumentation theory. Then it briefly describes how the *LeSAC* is deigned and shows some of its important functions. §3 concludes this paper.

2. *LeSAC* and a case study

To help clarification, we start with a very possible scenario in the future, of which the coding could be found in [19].

Example V1. *Currently, the law stipulates a number of behaviours that a human driver has to observe after an accident has happened. This includes a duty to stay at the scene of an accident and to provide first aid if necessary and feasible. Let's assume there is one passenger in the car, but he is (illegally) too drunk to do anything. In such a "contrary to duty" scenario, how should the AV car react now when somebody is hit?*

To handle this possible situation, we refer to current and relevant legal rules. We extract and select some most relevant information from traffic law and criminal law:

(1) It is illegal to drive a motor vehicle while intoxicated. People who drive while intoxicated shall lose their driving license and may be prosecuted in criminal law.

(2) A person who commits a hit-and-run accident will be criminal responsibility, especially when the escape causes the death or the driver is intoxicated.

(3) When an accident happens, the driver should take the responsibility to transfer the injured party to a safe place and provide aid if the situation is urgent.

(4) It's illegal to let a drunk passenger leave the car alone during the trip.

DL is the basic semantic of OWL or OWL2, which are the main logic languages of current legal ontologies. The basic notions of DL systems are *concepts* and *roles*. A DL system contains two disjoint parts: the TBox and the ABox. TBox introduces the terminology, while ABox contains facts about individuals in the application domain. There are many DLs and this paper is based upon the *ALC* expression [20,11].

2.1. Legal support system for autonomous vehicles

In [5] we constructed a structured argumentation framework *L-ASPIC* for reasoning based on an inconsistent legal ontology. Then given a legal ontology, particularly for AVs design, we can construct a *LeSAC* system based on *L-ASPIC* . Based on *LeSAC* , an argumentation framework for example V1 will be like:

Example V2.

$$\mathscr{N} = \left\{ \begin{array}{l} r_1 : Driver(x) \Rightarrow Sober(x); \\ r_2 : Intoxicated(x) \Rightarrow \neg LeaveCar(x); \\ r_3 : Driver(x), Intoxicated(x) \Rightarrow BeRevokedDrivingLicense(x); \\ r_4 : Driver(x), Intoxicated(x) \Rightarrow TakeCriminalResposibility(x); \\ r_5 : hitAndRun(x,y) \Rightarrow TakeCriminalResposibility(x); \\ r_6 : hitAndRun(x,y), causeDeath(x,y) \Rightarrow AggravatedPunishment(x); \\ r_7 : hitAndRun(x,y), Driver(x), Intoxicated(x) \Rightarrow AggravatedPunishment(x); \\ r_8 : CauseAccident(x), Injury(y) \Rightarrow transferToSafePlace(x,y); \\ r_9 : CauseAccident(x), Injury(y), NeedEmergencyAid(y) \Rightarrow doNecessaryAid(x,y) \end{array} \right\}$$

$$\mathscr{R}_s{}^{4} = \left\{ \begin{array}{l} r_{10} : Sober(x) \rightarrow \neg Intoxicated(x); \\ r'_{10} : Intoxicated(x) \rightarrow \neg Sober(x); \\ r_{11} : transferToSafePlace(x,y) \rightarrow LeaveCar(x); \\ r'_{11} : \neg LeaveCar(x) \rightarrow \neg transferToSafePlace(x,y); \\ r_{12} : doNecessaryAid(x,y) \rightarrow LeaveCar(x); \\ r'_{12} : \neg LeaveCar(x) \rightarrow \neg doNecessaryAid(x,y) \end{array} \right\} \quad \mathscr{K}^A = \left\{ \begin{array}{l} Driver(PS1); Intoxicated(PS1); \\ hitAndRun(PS1, Injury1); \\ Injury(Injury1); \\ causeDeath(PS1, Injury1); \\ CauseAccident(PS1); \\ NeedEmergencyAid(Injury1) \end{array} \right\}$$

$$\mathscr{P} = \left\{ \begin{array}{l} p_1 : Human\ lives\ should\ be\ protected\ as\ a\ priority; \\ p_2 : AI\ products\ should\ avoid\ extra\ risk\ about\ safety\ for\ their\ users; \\ p_3 : People\ should\ avoid\ putting\ others\ into\ dangerous\ by\ his\ own\ behaviours, \\ \quad\ and\ should\ bear\ corresponding\ responsibility. \end{array} \right\}$$

$prin(r_1) = p_3$; $prin(r_2) = p_2$; $prin(r_3) = p_3$; $prin(r_4) = p_3$; $prin(r_5) = p_3$;
$prin(r_6) = p_3$; $prin(r_7) = p_3$; $prin(r_8) = p_1$; $prin(r_9) = p_1$

We now present *LeSAC* 's reasoning functions combing the case study.

Legal compliance detection When engineers complete a whole design draft, they could use the consistency checking function to check whether this design is fully compliant with given laws and where conflicts are. If a design is consistent after reasoning, it means it is fully compliant with given laws. Otherwise, it is not. And by tracing where arguments conflict, we could know which part of the design needs modification. Based on the *LeSAC* in Example 2, we can at least construct the following two arguments.

Example (Example V2 cont.)**.**
$\alpha = (CauseAccident(PS1), Injury(Injury1) \Rightarrow transferToSafePlace(PS1, Injury1)) \rightarrow LeaveCar(PS1)$ *and*
$\beta = Intoxicated(PS1) \Rightarrow \neg LeaveCar(PS1)$.

α *and* β *attack each other, therefore the design is not complaint with given laws.*

Feedback for single change If the AV engineers want to keep the main design of an AV and only do some minimal changes, *LeSAC* can provide possible further legal consequences with these new details by instance checking. According to *LeSAC* , assertions are the conclusions of arguments. So based on the extension of arguments, we can decide whether an assertion is accepted.

To determine whether a certain modification is consistent with the current design and given laws, we translate this problem into whether a legal assertion about this AV can be accepted as a conclusion of an accepted/justified argument. Considering arguments α and β in Example 2.1, assuming that based on the priority ordering on legal principles ($\mathscr{P}$), α is preferred. Then α can defeat β, but not vice versa. Based on the *LeSAC* in Example 2, no other arguments are conflicting with α. As a consequence, α is sceptically justified as well as the assertion "$LeaveCar(PS1)$" .

Giving legal explanations Considering the needs of AV engineers, the explanation of reasoning results from *LeSAC* should show how a legal conclusion is obtained and which content in this situation makes it accepted or not. For any agent y, we can provide a formal explanation of why a legal conclusion X is accepted for certain design requirement consisting of two parts. One explains how X is reached by presenting all the premises and legal rules contained in $\mathscr{K}^A$ and $\mathscr{N} \cup \mathscr{R}_s$. The other explains why this legal conclusion is accepted by presenting all the legal information and principles applied to construct the arguments that defend α. Considering our running example, for the acceptance of the assertion "$LeaveCar(PS1)$", the explanation is:

[4]Rules r'_{10}, r'_{11} and r'_{12} are the transposed rules of rule r_{10}, r_{11} and r_{12}, respectively.

$$(\{Injury(Injury1), CauseAccident(PS1), NeedEmergencyAid(Injury1)\} \cup \{r_8, r_9\}) \cup \{p_2 < p_1\}$$

and for the acceptance of the assertion "$\neg LeaveCar(PS1)$", it is:

$$(\{Intoxicated(PS1)\} \cup \{r_2\}) \cup (\{Intoxicated(PS1)\} \cup \{r'_{10}\}) \cup \{p_1 < p_2\}$$

3. Conclusion and future work

This paper constructed a legal support system to help engineers of AVs improve legal compliance of the design by importing argumentation theory into legal ontology. In future, we will explore legal representation for importing machine learning, e.g. representation learning. We also plan to integrate it into a conventional engineering workflow.

References

[1] Act AI. Proposal for a regulation of the European Parliament and the Council laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union legislative acts. EUR-Lex-52021PC0206. 2021.

[2] Oberle D. Ontologies and reasoning in enterprise service ecosystems. Informatik Spektrum. 2014;37(4):318-28.

[3] Rosenfeld M. Dworkin and the One Law Principle: A Pluralist Critique. Revue Internationale de Philosophie. 2005;59(233 (3)):363-92.

[4] Zaouk AK, Wills M, Traube E, Strassburger R. Driver alcohol detection system for safety (DADSS)A Status update. In: 24th Enhanced Safety of Vehicles Conference. Gothenburg, Sweden; 2015. .

[5] Yu Z, Lu Y. An Argumentation-Based Legal Reasoning Approach for DL-Ontology. arXiv preprint arXiv:220903070. 2022. Available from: https://arxiv.org/abs/2209.03070.

[6] Prakken H. An abstract framework for argumentation with structured arguments. Argument & Computation. 2010;1(2):93-124.

[7] Hoekstra R, Breuker J, Di Bello M, Boer A, et al. The LKIF Core Ontology of Basic Legal Concepts. LOAIT. 2007;321:43-63.

[8] Breuker J, Valente A, Winkels R. Use and reuse of legal ontologies in knowledge engineering and information management. In: Law and the Semantic Web. Springer; 2005. p. 36-64.

[9] Griffo C, Almeida JPA, Guizzardi G. Towards a legal core ontology based on Alexys theory of fundamental rights. In: Multilingual Workshop on Artificial Intelligence and Law, ICAIL; 2015. .

[10] Barabucci G, Di Iorio A, Poggi F. Bridging legal documents, external entities and heterogeneous KBs: from meta-model to implementation. Semantic Web Journal. 2012.

[11] Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF, editors. The description logic handbook: Theory, implementation, and applications. Cambridge University Press; 2003.

[12] Schlobach S, Cornet R, et al. Non-standard reasoning services for the debugging of description logic terminologies. In: Ijcai. vol. 3; 2003. p. 355-62.

[13] Zhang X, Lin Z, Wang K. Towards a paradoxical description logic for the semantic web. In: International Symposium on Foundations of Information and Knowledge Systems. Springer; 2010. p. 306-25.

[14] Dung PM, Kowalski RA, Toni F. Assumption-Based Argumentation. In: Simari G, Rahwan I, editors. Argumentation in Artificial Intelligence. Boston, MA: Springer US; 2009. p. 100-218.

[15] Gómez SA, Chesñevar CI, Simari GR. Reasoning with Inconsistent Ontologies through Argumentation. Applied Artificial Intelligence. 2010;24(1&2):102-48.

[16] Borg A, Bex F. A Basic Framework for Explanations in Argumentation. IEEE Intelligent Systems. 2021;36(2):25-35.

[17] Čyras K, Rago A, Albini E, Baroni P, Toni F. Argumentative XAI: A Survey. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21); 2021. p. 4392-9.

[18] Bouzeghoub A, Jabbour S, Ma Y, Raddaoui B. Handling conflicts in uncertain ontologies using deductive argumentation. In: WI'17;. p. 65-72.

[19] JURIX-22 paper resource webpage;. Accessed: 2022-09-30. https://colab.research.google.com/drive/1k7KLeDs0RPvWBLOg7ySU8NsY9rHSVSMp?usp=sharing.

[20] Schmidt-Schauß M, Smolka G. Attributive concept descriptions with complements. Artificial Intelligence. 1991;48(1):1-26.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220470

WhenTheFact: Extracting Events from European Legal Decisions

María NAVAS-LORO [1] and Víctor RODRÍGUEZ-DONCEL
Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain

Abstract. This paper presents WhenTheFact, a tool that identifies relevant events from European judgments. It is able to extract the structure of the document, as well as when the event happened and who carried it out. WhenTheFact builds then a timeline that allows the user to navigate through the annotations in the document.

Keywords. event extraction, visualization, NLP, legal domain, timeline generation

1. Introduction

Events and their logical sequence are key to understanding legal decisions, being the storyline of pivotal importance. We therefore assume that a judgment can be described as a series of time-marked happenings (*events*) instead of focusing on the other entities (things), and to this aim we must be able to extract these events in an automatic fashion.

Before undertaking the event extraction task itself, discourse extraction is required; since legal decisions are long and complex, where the event is detected within the document is crucial regarding its relevance. Once the relevant parts of the document are determined, the next step involves training a system using documents annotated manually with relevant events, as well as the semantic resources available. Finally, the system is able to annotate different documents, allowing to visualize the relevant events in it. Additionally, in the online demo[2], a timeline with these relevant events is generated, easing navigation through the document.

The paper is organized as follows. Section 2 explores previous related work in literature. Section 3 introduces the system created to extract relevant events from legal decisions, explaining its different stages: document structure extraction, training strategies and extraction itself. Section 4 presents the evaluation of the system. Finally, Section 5 summarizes the main contributions and the future research lines to explore.

2. Related work

Beside generic efforts in event extraction such as the carried out by temporal taggers following TimeML [1,2] or related tasks such as frame-semantic parsing [3,4], semantic

[1]Corresponding Author: María Navas-Loro, Ontology Engineering Group, Universidad Politécnica de Madrid, Spain; E-mail: mnavas@fi.upm.es. This work was funded partially by the project Knowledge Spaces: Técnicas y herramientas para la gestión de grafos de conocimientos para dar soporte a espacios de datos (Grant PID2020-118274RB-I00, funded by MCIN/AEI/10.13039/501100011033) and by H2020 MSCA PROTECT (813497).

[2]https://whenthefact.oeg.fi.upm.es/

role labeling (SRL) [5,6] or open information extraction[3], some proposals have been made specifically in the legal domain. These works often involve *ad hoc* definitions of events, ignoring general event annotation schemes.

In the context of legal information retrieval, events can be considered as temporally bounded objects that have entities important as participants that played a significant role in a case. To this aim, Lagos et al. [7] propose an NLP semi automatic approach to enable the use of entity related information corresponding to the relations among the key players of a case, extracted in the form of events. They are interested in the topic, the roles, the location and the time, and consider different types of events. On the other hand, Maxwell et al. [8] reviewed 150 events extracted from 18 sentences from the Canadian Supreme court and compared them with automatic extraction using SRL on two cases. Another approach was done for Spanish [9], looking for patterns in documents that help them identify legal events and related information (*who, what, to whom* and *where*), and analyzing the verbs that occur in the texts. In order to improve information retrieval in Brazilian courts, similar work was performed for Portuguese [10].

In summary, legislation systems consist still of semiautomatic or even manual approaches. Most of the proposals within the legal domain tend to be supported by patterns, using manually crafted rules or semantic role labelling techniques [8,7].

3. Event Extraction

Based on a previous works about temporal expressions in the legal domain [11], first step for building a knowledge graph was to decide the source of the documents, since there are important differences among jurisdictions, even when they share the language. Due to the ease of importing and reusing judgments from their respective repositories, as well as the multilingual challenge it offers and the possible associated documents that could eventually be added to a knowledge graph, we decided to work with decisions from European courts, namely the European Court of Human Rights (ECHR) and European Court of Justice (ECJ). Choosing a specific source also allowed us to analyze the structure of the documents, which improves the ability to extract relevant events [12]. Regarding the format of the annotations, we will use the one specified in the EventsMatter corpus [13][4].

The remaining of this section will present the structure extractor of the judgments (Section 3.1, the different training strategies used (Section 3.2) and the pipeline of the event extraction algorithm (Section 3.3), that applies the two previous techniques.

3.1. Structure Extraction

From an analysis performed in the EventsMatter [13], the only available corpus of judgments annotated with events (to the best of our knowledge), we can confirm the importance of the sections in identifying which events are relevant and which are not. To this end, we have developed a Structure Extractor that

1. Detects the structure of the document and divides it into parts with a title, a type, a parent and the begin and end offsets.

[3] https://stanfordnlp.github.io/CoreNLP/openie.html

[4] A very preliminary version of this work was briefly introduced in the paper describing this corpus.

2. Looks for the most relevant sections and sends the sentences within to the algorithm that extracts the events, ignoring sections such as references to laws.

This Structure Extractor is currently able to handle the structure of the ECHR and ECJ documents, but in such a way that a new document type can be easily added. Additionally, if for any reason the processed documents did not adhere to the expected structure (for example, with very old cases that followed a different format), it would simply return all sections.

3.2. Training Strategies

Regarding the training strategy of the event extraction system, we used both semantic and syntactic considerations. On the one hand, we collected all the events and attached arguments annotated in the training set of the EventsMatter corpus [13]. The EventsMatter corpus is a collection of 30 legal decisions manually annotated with events and their arguments (namely, *who*, *when* and *what*, called *core*). Once collected, we stored both the core of the events and the relations among their different parts. On the other hand, we also used an external semantic resource, FrameNet, to enrich the keywords we use to identify legal events. Subsequent sections provide a detailed description of both approaches.

3.2.1. EventsMatter Training Set

The first step of the training phase was to collect all the event mentions in the corpus training set. We isolated then the parts of the sentences annotated as core and generated a sentence just with it, adding has generic subject "They" in order to make them simple to parse and grammatically correct. Thereupon we iterated over all these *simple* sentences, creating a frame for each of the main verbs of the sentences that stored the information of all the mentions of this verbs along the corpus. This is, that for instance the verb "lodge" (that is to some extent a *light verb*[5] in the legal domain) can appear in several sentences carrying different meaning depending on the object attached. Some examples of its use would be the constructions "lodge a complaint", "lodge a request", "lodge an appeal", "lodge an objection" or "lodge an action". It should be noted that most of these cases could be simplified using a single semantic-carrying verb, such as "to complain" or "to request", but that the legal domain tends to recur to these paraphrasing in texts, since they usually imply not just an action but also a formal procedure (usually administrative).

Each of verbs found in this phase constitutes a *frame* that will be used to identify and classify future mentions of each of the verbs in new texts. Finally, it must be noted that we distinguish between passive and active voice when searching for the dependency parsing relations among the members of the core of an event. This is a consideration that might not be important in general kind of texts, but the legal domain tends to present a high rate of passive verbs. Among the events in the training set, for instance, we find that the 14% of the mentions were expressed as passive sentences.

Two couples of text files containing (1) the *simple* version of each sentence with a relevant event mention and (2) the type of events of each of the mention are available

[5] *Light verbs* are those verbs that have little semantic meaning, needing therefore more words to constitute a full predicate. This is for instance the case of the verbs "make" or "take" in English. For more information on this linguistic phenomenon, please check the work by Butt [14].

within the system – a couple for all the sentences of the corpus (named *all*) and another for just the training part (*train*). The collection of events can be easily extended by adding to the files new sentences and their respective types, and a detailed example of this Frame structure can be found in the website.

3.2.2. FrameNet training

It is straightforward that some events not present in the training set of the EventsMatter corpus should be detected in other documents, and even that events considered not relevant in those documents can be relevant in other cases.

This is why, in addition to the events gathered from the training set explained previously, we decided to enrich the system with frames from FrameNet [15]. FrameNet is a database that contains semantic frames together with the words that represent them in text, as well as additional information such as the arguments this frame can present. Since frames represent situations, they can be understood as events to some extent, and incorporating a selection of them to our target events would help to generalize our approach. Since not all the frames in FrameNet are of interest, we manually inspected the database using the FrameGrapher tool[6], that allowed us to navigate through it and find the most relevant frames to our task. After examining the different relations among the frames, we found the most general ones, as well as their children, and imported their information. These most legally representative parent frames were namely "Committing_crime", "Crime_scenario", "Law", "Obligation_scenario", and "Misdeed". The frames collected from them, together with the lexical units associated to them (that is what we will look for in the text), are detailed in the webpage. A text file containing this information is available in the system. In order to add more frames, it is only necessary to add them to the file maintaining the same format.

3.3. Event Extraction

Regarding the event extraction itself, Fig. 1 depicts the pipeline of the tool. We detail the different stages of the processing below.

First step consists of finding the relevant parts of the text to annotate, using for this the Structure Extractor detailed in Section 3.1. If the structure is not recognized, the whole text will be annotated, what obviously impacts in a negative way in the amount and quality of the events. Otherwise, just the relevant parts of the document are processed subsequently.

Next step is to find the sentences involving temporal expressions. To this aim we adapt and integrate the functionality of Añotador [16], a temporal tagger able to recognize temporal expressions. If there is at least one temporal expression in a sentence, we check if it is a special case (namely the application lodgement, that always follows the same syntactic structure). If so, we annotate the arguments and go to the next sentence. If not, we check if the sentence contains any of the events gathered from the training corpus. If so, we do the dependency parsing (*deppar*) of the sentence (using CoreNLP [17]) and check if it is valid and look for the arguments (see (1) below). If not, we check again for the legal frames specifically selected from FrameNet. If this is the case, we check them similarly that in the events case (see (2)). Once we detected the main event

[6] https://framenet.icsi.berkeley.edu/fndrupal/FrameGrapher

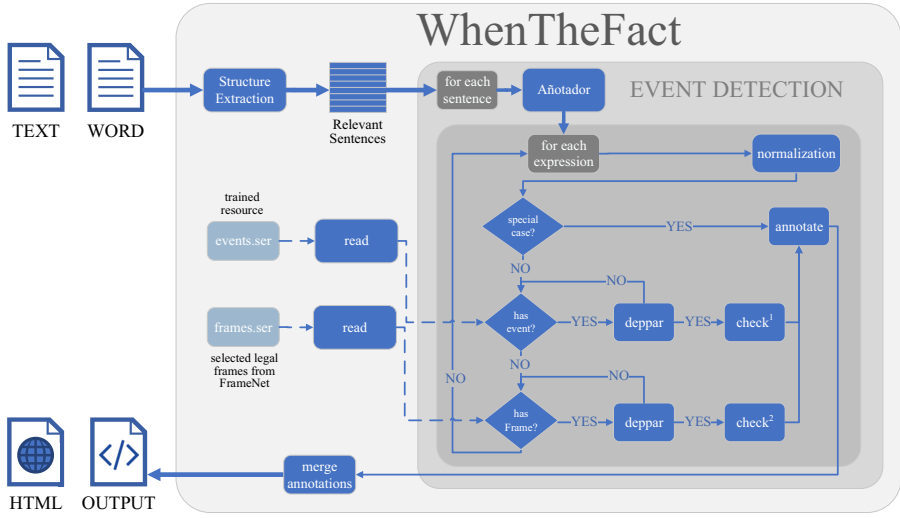


Figure 1. Pipeline of WhenTheFact.

in the sentence, if there was more than one temporal expression in it, we will select the temporal expression that is the closest to the core of the event.

(1) For the events, we check if it is not an auxiliary verb nor in the gerund form. Then we check if it is in passive or active voice. Depending on this, we will look either for the relations gathered from passive training cases or from active ones.
(2) For the frames, the check function is similar to the events' one, but there are no specific relations stored for each frame, so the argument "who" and the extent of the core are therefore detected using default relations.

Once all the sentences have been explored, we merge all the annotations and produce the output. This output consists of an annotated XML and as a visual HTML that also includes a timeline built from the retrieved events.

4. Evaluation

Regarding evaluation, we have compared WhenTheFact's results against the EventsMatter corpus and checked it has improved. The evaluation is depicted in Table 1.

5. Conclusions

In this paper we have presented WhenTheFact, an event extractor able to annotate relevant legal events taking into account the structure of a legal judgment. Next steps include solving correference, currently uncovered, since for now we just get the textual mention, that can consist of pronouns. Once this is achieved, queries will be able to retrieve for instance the timeline of one actor's involvement in a case.

Table 1. Comparison between the previous implementation of the WhenTheFact event extractor (OLD) and the new implementation (NEW).

		Event				Event Components					
		Identification		Type		What		When		Who	
		Len	Str	Len	Str	Len	Str	Len	Str	Len	Str
OLD	P	85.71	80.00	47.14	42.86	80.26	23.68	77.59	72.41	75.00	68.75
	R	77.92	72.73	42.86	38.96	69.32	20.45	63.38	59.15	63.16	57.89
	F	81.63	76.19	44.90	40.82	74.39	21.95	69.77	65.12	68.57	62.86
NEW	P	86.49	81.08	54.05	51.35	83.75	82.50	79.03	74.19	81.43	74.29
	R	83.12	77.92	51.95	49.35	76.14	29.55	69.01	64.79	75.00	68.42
	F	84.77	79.47.19	52.98	50.33	79.76	30.95	73.68	69.17	78.08	71.23

Also multilinguality is currently being explored. Although several approaches have been tested already, none of them has been good enough to guarantee acceptable results for all the languages. Finally, the tool can be used not just for visualization, but also to populate legal knowledge graphs to be used in different contexts.

References

[1] Verhagen M, et al. Automating Temporal Annotation with TARSQI. In: Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions. ACL; 2005. p. 81-4.

[2] Chambers N, et al. Dense event ordering with a multi-pass architecture. Transactions of ACL. 2014;2:273-84.

[3] Swayamdipta S, et al. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. arXiv preprint arXiv:170609528. 2017.

[4] Alam M, et al. Semantic role labeling for knowledge graph extraction from text. Progress in Artificial Intelligence. 2021:1-12.

[5] Gardner M, et al. AllenNLP: A Deep Semantic Natural Language Processing Platform; 2017. .

[6] Agerri R, et al. IXA pipeline: Efficient and Ready to Use Multilingual NLP tools. In: LREC. vol. 2014; 2014. p. 3823-8.

[7] Lagos N, et al. Event extraction for legal case building and reasoning. In: International Conference on Intelligent Information Processing. Springer; 2010. p. 92-101.

[8] Maxwell KT, et al. Evaluation of semantic events for legal case retrieval. In: Proceedings of the WSDM'09 Workshop. ACM; 2009. p. 39-41.

[9] Sierra G, et al. Event Extraction from Legal Documents in Spanish. In: Workshop on Language Resources and Technologies for the Legal KG, LREC; 2018. .

[10] Bertoldi A, et al. Cognitive Linguistic Representation of Legal Events. Towards a semantic-based legal information retrieval. In: COGNITIVE 2014; 2014. .

[11] Navas-Loro M, et al. TempCourt: evaluation of temporal taggers on a new corpus of court decisions. The Knowledge Engineering Review. 2019;34:e24.

[12] Navas-Loro M, Santos C. Events in the legal domain: first impressions. In: Proceedings of the 2nd Workshop TeReCom (JURIX 2018); 2018. p. 45-57.

[13] Filtz E, et al. Events Matter: Extraction of Events from Court Decisions. In: Proceedings of JURIX. 336; 2020. p. 33-42.

[14] Butt M. In: The light verb jungle: still hacking away. Cambridge University Press; 2010. p. 48–78.

[15] Baker CF, Fillmore CJ, Lowe JB. The Berkeley FrameNet Project. In: Proceedings of ACL/COLING - Volume 1. ACL; 1998. p. 86–90.

[16] Navas-Loro M, Rodríguez-Doncel V. Annotador: a temporal tagger for Spanish. Journal of Intelligent & Fuzzy Systems. 2020;39:1979-91. 2.

[17] Manning CD, et al. The Stanford CoreNLP Natural Language Processing Toolkit. In: Proceedings of the 52nd Annual Meeting of the ACL, System Demonstrations; 2014. p. 55-60.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220471

Autosuggestion of Relevant Cases and Statutes

Saran PANDIAN [a,1], and Shubham JOSHI [a,2]
[a] *Lawnics Technologies, India*

Abstract. In this paper, we describe a method to help legal practitioners in citing the relevant case laws and statute laws for the specified legal issue. In this method, we consider the cited case and statute law as single tokens where we try to find the relevant tokens based on the words around these tokens. We observed that context-based representations outperformed lexical-based representations and distributional representations. Also, we observed that the method works better for statute law retrieval compared to case law retrieval.

Keywords. Precedents, Statutes, Masked Language Model, Citation Recommendation, Distributional Representation, Context-Based Representation

1. Introduction

To understand which statutes or/and cases are the most relevant to the legal issue, we need to check what laws have been frequently cited in solving contextually similar legal issues or queries in past. In this paper, we try to investigate novel approaches to create an autosuggestion tool to predict the most relevant cases and statutes for similar contextual legal issues/queries. When a legal citation can be represented based on the context around it, it can be easily retrieved using a search engine when words with similar context are given as queries. Moreover, we also study the effect of the frequency of citations impacts the performance of the model.

2. Related Work

Traditional techniques for citation recommendation usually include BM25[1], Indri[2], etc. Among these BM25 is considered to be a strong baseline when it comes to the legal domain[3]. However, BM25 considers only lexical matching and not semantic matching. With the advent of deep learning techniques, research in direction of including the semantics of the documents for citation prediction tasks is widely done[4]. BM25+BERT[5] has shown better results than BM25. Our challenge was to come up with an approach for citation recommendation that finds relevant citations from a larger pool of candidate citations and recommends citations regardless of the length of the cited documents.

[1] Saran Pandian, saran@lawnics.com
[2] Corresponding Author: Shubham Joshi, sshubham@lawnics.com

3. Dataset Creation

The key requirement of the dataset was to find legal texts that was rich with the citations. Judges who wants to use laws as a reference in their judgements generally cites cases and statutes around the legal issues that the judgements deals with, hence we decided to include paragraphs of judgements that has citations in it. Since the Supreme Court of India is considered to be the highest authority and is cited multiple times through out all the courts we decided to include only judgments passed by this court. To prepare the dataset, we considered more than 55000 supreme court cases[3] from the year 1947 to 2021. Despite India having many statutory laws at central and state level, we have included only 1260 statutes as these were passed by central government having effect all over India. Data used from these statutes[4] were also included along with SC cases to create the legal corpus.

All these cases and statutes consist of multiple paragraphs. It was found that a total of 16,37,897 paragraphs were available. Using regular expressions we found paragraphs having case citations (paras containing words vs., Vs, Vs.) and statute citations (paras containing words Section, sec.,s., Article, Art., art., Act, act, etc.). 54541 paragraphs were found to have case citations and 242484 had statute citations. These paragraphs were then manually reviewed for additional annotation(especially for acts corresponding to sections) using 10 volunteers from the legal industry. Legal Volunteers were provided with proper guidelines and tools[5] to find out the citations and annotate them for higher accuracy. Also, each volunteer was then asked to review the annotations of other volunteers to confirm the quality of the dataset. We decided not to go for an Inter-annotator agreement as the task of finding citations is of elementary level for legal volunteers. Then using our proprietary database, each cited case and statute is given its ID using pattern matching with help of the name, year of judgment (in case of precedent law), and enactment(in case of statute law).

The citations are not spelled uniformly throughout the corpus; for example, the cited case could be *Maneka Gandhi v. Union of India*. But the annotated title of this cited case is *Maneka Gandhi vs UOI*. To normalize the text, we use fuzzywuzzy[6] string matching algorithm to match the annotated citation with the actual document name from a lookup table and replace it with the corresponding ID. Scores for matches were given on a scale of 100, based on the number of overlapping tokens and the order of tokens. In order to avoid string mismatches, the threshold of score 70 was set. The final dataset contains preprocessed legal text with citations being replaced with IDs.

The total number of unique citations were found to be 29010 out of which 7457 were statute citation and 21553 were case citations. The highest statute citation was found to be 11935 and the case citation was 220. We would like to convey that the dataset is part of a proprietary dataset for a commercial application. Hence the dataset cannot be published.

For final preprocessing we converted raw text to lowercase, followed by the removal of stopwords, numerals, and punctuations. It is critical to note that no preprocessing is needed to be done on citation IDs. For our approach, we used 16,37,897 paragraphs

[3] https://main.sci.gov.in/judgments
[4] https://indiacode.nic.in/
[5] https://ubiai.tools/
[6] https://pypi.org/project/fuzzywuzzy/

scraped from Indian Supreme court cases and statutes for this experiment. These paragraphs consist of text where we replaced the title of cited law with corresponding citation IDs. Further, we decided to split paragraphs in an 80:20 ratio randomly for training and testing respectively.

4. Methodology and Experiments

For a given context around a citation, the task is to find the corresponding citation c_i from citation set C. We try to Autosuggest the token based on the context. For this experiment, we consider only citations that have been cited more than 3 times in the corpus. this paper, we provide techniques for the recommendation of citation thus auto-suggesting statutes and cases based on textual context. In legal literature, citations are based on text. The whole purpose of the experiment is to make AI models understand the context and recommend citations for similar contextual queries by going through the text of cited law. In NLP, Predict Distributional Semantics Model(DSM) and context-based Distributional Semantics Model use co-occurring words around a word to arrive at a representation of the semantics of a word. Words occurring in similar environments tend to be close w.r.t semantic representation[6]. Predict DSM representation is unique for each word whereas context-based DSM representations differ with respect to sentences. Each citation is considered to be a single token, to arrive at a semantic-based representation for each citation. Two models namely LawCite2vec (Predict DSM method) and Bert4LawCite (Context-based DSM method) were trained and each of these DSM models is compared with BM25 and BM25+BERT.

4.1. LawCite2vec and Bert4LawCite

4.1.1. Training

Mikolov et al[7] used the skip-gram technique in their paper to get vector representation for words. Rather than training the LawCite2Vec model from scratch, the existing Law2Vec model[8][7], which is pre-trained on a substantial legal corpus was further fine-tuned on our training data, where we considered each citation to be a single token. We completed text preprocessing as mentioned in 3. To carry out the task, we considered a window size of 10 to train the model for 30 epochs, as some citations are scarcely cited. We trained the model with the help of 12 GB RAM processing power.

Lately, BERT[9] has become a prominent state-of-the-art model for all downstream NLP tasks. We also decided to use BERT for token prediction tasks where we try to predict tokens consisting of citation Ids. For each query consisting of the masked token, we are trying to predict the citation Ids as an output. We decided to use pre-trained LegalBERT[10], as this model has already been trained on a large corpus of legal texts[8]. We loaded the pre-trained weights to finetune it for our downstream task as it is a context-based DSM and can change the representation of the masked token based on the context around it. For training Bert4LawCite, We need to pick sentences from paragraphs that have citations within them. 298,946 sentences from train paragraphs with statute cita-

[7]https://archive.org/details/Law2Vec

[8]https://huggingface.co/nlpaueb/legal-bert-base-uncased

tions and 16377 sentences with case citations were picked with citations replaced with [MASK] token. Two separate models one for statute recommendation and the other for case recommendation were trained in order to avoid bias. We trained the model with the help of a K80 GPU with 12 GB RAM processing power.

4.1.2. Testing

Firstly candidate citations need to be indexed. For recommendation using LawCite2Vec embeddings, the candidate citation embeddings were indexed in Elasticsearch[9] service. For testing purpose we decided to index the embeddings of citations that have been cited more than 3 times as the citation is relevant to legal context only when it has been cited multiple times. It was found that 3133 statute citations and 5174 case citations (denoted by Cs and Cc respectively) had been cited more than 3 times. To come up with queries, around we can considered 39000 sentences from 20 percent of paragraphs chosen for testing that contains 4800 citations. Then these citations were removed for the queries. Queries preprocessed as the mentioned in 3 .The query representation is done by taking the mean average of the LawCite2Vec representation of words in the sentence. The citations with the closest vector representation to this query representation are retrieved through Elasticsearch. The experiments are done for statute retrieval and case retrieval separately.

To test Bert4LawCite, we used similar test data to test the LawCite2Vec model. We decided to remove citation tokens and place a [MASK] token in the middle of the sentence. As with every other prediction model, BERT also considers each word in the whole dictionary and provides the most relevant scores. However, for our citation recommendation task, we want to restrict the vocabulary to the statute citation set Cs and case citation set Cc. After feeding the query that consists of a masked token, we get the score for each citation ID from C_s and C_c sets and consider the highest ranked as the final output of the model.

5. Baseline

We decided to use BM25 as a baseline after going through previous research papers in the legal domain[3]. To make a fair comparison of BM25 output with the other two approaches, we indexed data of cases, sections, and acts in Elasticsearch that were masked and cited at least three times in the test documents. Moreover, we also preprocessed data and queries as per the steps mentioned in 3. We further retrieved the relevant data as per queries and ranked the same based on the BM25 score. On top of BM25, we also tried using BERT for re-ranking by taking the mean of LegalBERT[10] representation of every paragraph used to represent the whole document. The documents were ranked based on the distance between the LegalBERT representations of the query and the documents.

6. Results and Analysis

A citation that has been cited multiple times for a similar legal context can also be considered a trustworthy and relevant candidate for a query with a similar context. As we

[9]https://www.elastic.co/

are only considering a frequently cited candidate as the most relevant candidate for the given mask or BM25 score, we understand that relevance values shall be binary only. We considered Mean Reciprocal Rank(MRR) and Hit Rate(HR) as the most suitable metrics for evaluation as they are prominent with binary outputs. MRR score is based on how far the first relevant item is present in the recommended list (In our case only consider one document is the most relevant). Reciprocal Rank(RR) is the reciprocal of the rank at which the first relevant document was retrieved. 1 is the best RR score which means the relevant document is retrieved in the first place. HR score is based on the ratio of relevant recommended items to the total number of relevant items for a query. 1 is the best HR score which means all relevant documents are present in the recommended list. Further, we evaluated the results for the statute and the case law recommendations separately as we wanted to analyse the score difference between the two thoroughly. For an in-depth analysis of the results, we conducted a citation frequency analysis.

6.1. Results

6.1.1. Statute Law Recommendation

LawCite2Vec and Bert4LawCite performed better than BM25 and BM25+BERT, as per hypothesis. As shown in 1, the statute laws performed better as the frequency of the citations was very high, allowing the models to give better representations for the statute citations. BM25+BERT gave poor as the BERT failed to have good representations for lengthy legal documents. Bert4LawCite gave better results as it considers every citation to be a single token and uses a SOTA language model with higher perplexity[9] to predict the citations.

Table 1. Evaluation Scores

	Statute Law Recommendation			Case Law Recommendation		
	HR@1	HR@10	MRR@10	HR@1	HR@10	MRR@10
BM25	0.0916	0.2571	0.1400	0.1397	**0.3303**	0.1993
BM25+BERT	0.0157	0.08773	0.03182	0.0068	0.0459	0.0148
LawCite2Vec	0.1499	0.3930	0.2147	0.0222	0.0848	0.0385
Bert4LawCite	**0.3607**	**0.6730**	**0.4620**	**0.2120**	0.2987	**0.2398**

6.1.2. Case Law Recommendation

Though the number of case IDs was more than statute IDs, each case ID's overall frequency was less than statute IDs. Bert4LawCite managed to beat BM25 with a considerable difference, but LawCite2Vec performed poorly for the auto suggestion for Cases.

6.2. Citation Frequency Analysis

To understand the effect of the Frequency of citations on the performance of the model the test data was divided into chunks of equal ranges based on the frequency of citations. It was found that frequency has a strong effect on Performance. Even for Case Law Retrieval, where LawCite2Vec was found to perform worse compared to BM25, performed better than BM25 for chunks with higher frequency ranges as shown in Fig 1. We are neglecting BM25+BERT for analysis because of poor performance.

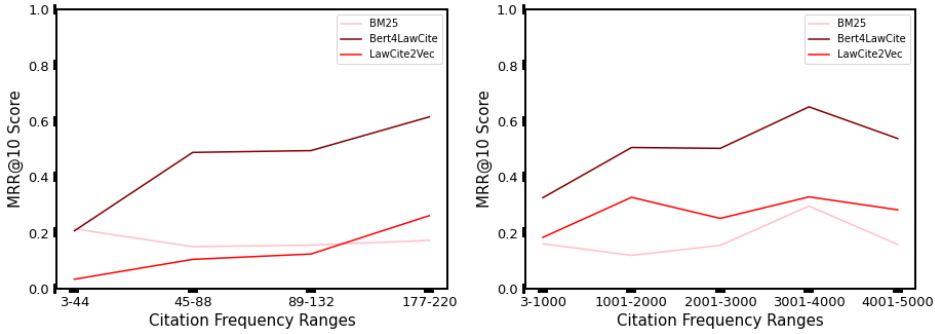


Figure 1. Citation Frequency Analysis for Statute Law and Case Law Recommendation

7. Conclusion and Future Scope

We have introduced a novel approach for citation autosuggestion/ recommendation task by turning multiple-length citations with a more straightforward single token ID and using it to train models like LawCite2Vec and Bert4LawCite. Through this approach, we found that these systems can be used for commercial applications in the field of law where it can be used as stand alone recommendation system for a legal question as a query to recommend relevant cases and statutes. Also, it can be used as re-ranking tool to improve the mean average precision of legal information retrieval tool. We believe future research directions can be conducted by including the content of the cited laws along with context to reduce the dependence on citation frequency and increase the diversity and serendipity of data to reduce the bias on highly cited citations.

References

[1] G. Amati, *BM25*. Boston, MA: Springer US, 2009, pp. 257–260. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_921

[2] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft, "Indri: A language-model based search engine for complex queries (extended version)," *Rapport technique. CIIR*, 2005.

[3] G. M. Rosa, R. C. Rodrigues, R. Lotufo, and R. Nogueira, "Yes, bm25 is a strong baseline for legal case retrieval," *arXiv preprint arXiv:2105.05686*, 2021.

[4] Z. Huang, C. Low, M. Teng, H. Zhang, D. E. Ho, M. S. Krass, and M. Grabmair, "Context-aware legal citation recommendation using deep learning," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, 2021, pp. 79–88.

[5] R. Nogueira and K. Cho, "Passage re-ranking with bert," *arXiv preprint arXiv:1901.04085*, 2019.

[6] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[8] I. Chalkidis and D. Kampas, "Deep learning in law: early adaptation and legal word embeddings trained on large corpora," *Artificial Intelligence and Law*, vol. 27, no. 2, pp. 171–198, 2019.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2898–2904. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.261

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220472

Extracting References from German Legal Texts Using Named Entity Recognition[1]

Silvio PEIKERT [a,2], Celia BIRLE [a,3], Jamal AL QUNDUS [b,4]
Le Duyen Sandra VU [a,5] Adrian PASCHKE [a,6]
[a] *Fraunhofer Institut für Offene Kommunikationssysteme, Berlin, Germany*
[b] *Middle East University, Amman, Jordan*

Abstract. Information extraction tasks are particularly challenging in specific contexts such as the legal domain. In this paper, Named Entity Recognition is used to make legal texts more accessible to domain experts and laymen. This paper focuses on extracting law references and citations of court decisions, which occur in various syntactic formats. To investigate this task a reference data set is constructed from a large collection of German court decisions and different NER-techniques are compared. Pattern matching, probabilistic sequence labeling (CRF), Deep Learning (BiLSTM) and transfer learning using a pretrained language model (BERT) are applied to extract references to laws and court decisions. The results show that the BERT based approach achieves F1 scores around 0.98 for both tasks and outperforms methods from prior work, which achieve F1 scores of 0.89 (CRF for law references) respectively 0.82 (CRF for court decisions) on the same data set.

Keywords. Named Entity Recognition, Knowledge extraction, Legal data

1. Introduction

Lawyers search for laws and past decisions in each new case, either to estimate a possible outcome or to use them as arguments or counter-arguments. “Search” implies the recognition of entities in the form of norms and decisions, either manually or automatically. Due to the large amount as well as the structure and complexity of the data available, exploring and manual recognition are complex and time consuming [1].

Named Entity Recognition (NER) systematically extracts semantic document features in order to support downstream tasks such as information retrieval. NER is not a simple task and is a research area of several AI disciplines, e.g. [2,3]. Legal texts and text documents in general contain a multitude of references to external entities, which provide important background information. These references can be used to define semantic and machine understandable representations of documents.

[1] partially funded by the German Federal Ministry of Education and Research (BMBF), grant no. 03COV03F
[2] Corresponding Author: silvio.peikert@fokus.fraunhofer.de, ORCID: 0000-0001-5716-1540
[3] ORCID:0000-0002-5652-831X
[4] ORCID:0000-0002-8848-1632
[5] ORCID:0000-0003-3307-675X
[6] ORCID:0000-0003-3156-9040

Beyond common entity types such as **person**, **location** or **organization** which can be extracted using many general purpose NER approaches, citations and references to **court decisions**, **laws**, **regulations** and **legal literature** are of particular importance in the legal domain. Although these references generally follow certain syntactic rules, in practice, a variety of deviating formats are encountered. Additionally, these references contain many special characters and abbreviations, which further complicates algorithmic NER. The following text fragment from a German court decision contains a reference to a **law** and two different *court decisions*:

> "(...) ist eine solche Entscheidung des erkennenden Gerichts gemäß
> **§ 238 Abs. 2 StPO** herbeizuführen (*BGH, Beschlüsse vom 14. Dezember 2010 - 1 StR 422/10, StV 2011, 458*; *vom 9. November 2017 - 1 StR 554/16*)."

The ability to automatically recognize and resolve such references enables more efficient information retrieval systems. These features improve accessibility by providing links and context information and more specific queries by exploiting semantic context. Furthermore, citation graphs can be used to identify documents of special importance using ranking algorithms.

The described applications require three steps: (1) the automatic recognition of references, (2) the separation of these references into fragments and identifiers and (3) the lookup of the references in a knowledge base. This paper focuses on the automatic recognition of references to **laws** and **court decisions**.

A data set for these two tasks is constructed from German court decisions and different NER approaches are compared. The approaches include pattern matching, probabilistic sequence labeling using conditional random fields (CRF) [4], deep learning using bidirectional LSTMs (BiLSTM) [5] and transfer learning based on a pretrained multitask language model (BERT) [6].

2. Related Work

NER efforts in the legal domain range from automatically identifying legal parties in court files, sometimes used to automate anonymization preceding publication in order to comply with data privacy standards [7,8], to using NER to build ontologies [9,10], to automatically annotate legal documents [11]. These applications rely on the recognition of common named entities such as **person**, **location**, **organization**.

Other types of applications, e.g. those trying to build citation graphs of legal documents in order to find legal precedent and other connected material [12,13,14,15,16] or those automating summaries of legal texts [17], consider additional entities, namely cross references to **legal norms and regulations**, **court decisions** and **legal literature**.

[18] applied different legal sentence classes to investigate the applicability of machine learning to different document types, while [19] considered legal contracts from German legal data and provided software for legal entity linking and extraction; they employed a two-stage process using NER and NED (Named Entity Disambiguation). Their best performance for NER reached an F1 score of 92%.

[20] also deals with the extraction of general legal named entities (not including citations) from German legal documents and achieved better performance by using BERT models.

[21] demonstrated on Canadian documents that the use of context might improve results further.

[12] focused on legal cross references. They used data from Luxembourg and extracted references using complex regular expressions. [22] defined a set of semantic classes and applied sequence labeling to evaluate a number of classifier models. They used approaches based on CRF and BiLSTM.

The approach described in section 3 builds on results from [22] and uses similar methods. This paper considers additional data sources and approaches based on pretrained language models.

3. Methodology

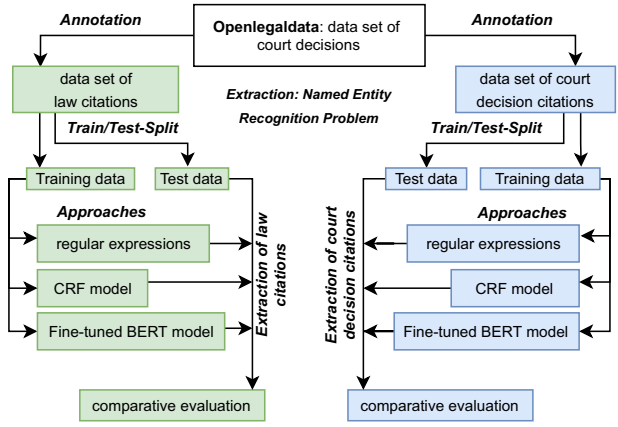


Figure 1. Overview of the Methodology

The detection and extraction of citations of law texts and court decisions can be modeled as a NER task. Two data sets, one for each NER task, were constructed. The extraction of these citations is approached using pattern matching, conditional random fields, bidirectional Long-Short Term Memory Networks and finally, transfer learning with BERT. Figure 1 summarizes the workflow used to compare these methods.

To construct a benchmark data set *Open Legal Data* [7] was used. For both reference types 100 documents from 2019 were randomly selected. and cleaned from their HTML encoding. The two data sets were then manually annotated using Docanno[8]. The resulting data sets include 3.018 law references and 1.297 citations of decisions.

The annotated data sets were converted into the CoNLL 2002 format with its BIO scheme. Each data set contains a single class - GS (*Gesetz*=law) and RS (*Rechtsprechung*=court decision) respectively. For the citations of court decisions an additional more detailed annotation was created. It includes subclasses for file reference number, date and print source, which are easily identifiable using regular expressions. Table 1 illustrates both annotations.

[7] https://static.openlegaldata.io/dumps/de/2020-12-10/cases.jsonl.gz
[8] https://github.com/doccano/doccano

Table 1. CoNLL 2002: Example of the BIO-schemes for references to court decisions

BIO	Detailed BIO	BIO	Detailed BIO
Beschluss B-RS	BGH, B-RS	StR I-RS	4 I-RS-AZ
vom I-RS	Urteil I-RS	124/16, I-RS	StR I-RS-AZ
23. I-RS	vom I-RS	JurionRS I-RS	421/00, I-RS-AZ
August I-RS	22. I-RS-DT	2016, I-RS	NJW I-RS-FS
2016 I-RS	Februar I-RS-DT	26140 I-RS	2001, I-RS-FS
- I-RS	2001 I-RS-DT	III. O	1874, I-RS-FS
2 I-RS	- I-RS	27 O	1876 I-RS-FS

Both types of references follow fairly specific rules, which make a pattern matching approach promising. In practice it is more challenging than anticipated to cover all the variations, e.g. the version of the text is quoted or not, variing levels of hierarchy of the referenced text and so on. Our attempt to define a rule set to extract law references using the language processing toolkit Spacy[9] resulted in 34 different patterns. Detecting citations of court decisions was also attempted using pattern matching. Parts of the references to court decisions (date of the decision, file reference number, print publication) follow clear syntactic patterns and can be reliably identified. However, not all citations include these features, since citations may be incomplete, referring to full citations mentioned earlier, or may be arranged differently. These are incompletely or not detected.

CRFs and BiLSTMs are machine learning approaches to perform NER. [22] has already shown that these approaches can solve the NER task studied in this paper. For the law reference extraction task, CRF and BiLSTM trained on the LER corpus [10] as used by [22] have been used. These results were compared to a CRF model trained on the data set described above. For the decision citation task, the pretrained BiLSTM model was compared to two CRF models trained on the data set described above. One was trained on the shorter BIO annotations and one on the more detailed BIO annotation.

Finally, a BERT-model was used for both tasks: The pretrained bert-base-german-cased model from huggingface[11] was fine-tuned on the data sets for both NER tasks as described in [6]. During fine-tuning the token representations generated by BERT are fed into an output layer for sequence tagging and the resulting network is trained. We used 20 epochs and a batch size of 16 for this process.

4. Results and Discussion

The results for both tasks are summarized in Table 2. A simple pattern matching approach achieves F1 of 83.24 % for the law reference detection task. Results using a pretrained CRF are in a similar range and a CRF trained on the training portion of the presented data set increases F1 to 89.44 %. This increase of performance can be attributed to better generalisation due to a better data set coverage. BiLSTM achieves worse results due to a high number of false positives. By far the best results are achieved using BERT (F1 score of 98.82 %).

[9] https://spacy.io/
[10] https://github.com/elenanereiss/Legal-Entity-Recognition
[11] https://huggingface.co/bert-base-german-cased

Table 2. Extraction Results of the compared approaches

Approach	Law References			References to Court Decisions		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Pattern Matching	83.53 %	82.94 %	83.24 %	-	-	-
Pretrained CRF	84.88 %	78.78 %	81.72 %	-	-	-
CRF	89.94 %	88.90 %	89.44 %	-	-	-
CRF: BIO	-	-	-	86.29 %	77.54 %	81.68 %
CRF: Detailed BIO	-	-	-	81.36 %	72.73 %	76.80 %
Pretrained BiLSTM	24.7 %	76 %	37.34 %	73.97 %	73.09 %	74.24 %
Fine-tuned BERT	**98.53 %**	**99.11 %**	**98.82 %**	**96.92 %**	**98.26 %**	**97.58 %**

The extraction of decision citations using pattern matching did not prove promising and was not further pursued. The pretrained BiLSTM achieved acceptable results (F1 74.24 %). A CRF trained on the data set improved F1 to 81.68 %. The use of a more detailed annotation scheme did not yield improved performance. The best results were also obtained using a BERT. The model achieved a F1 score of 96.19 %.

The data set presented covers more variations, e.g. it includes references to European court decisions, but has a lower number of samples than the data set used for the pretrained models. Despite the low number of samples CRF achieved performance comparable or superior to a pattern matching approach. By far the best performance for both tasks was achieved by BERT, since transfer learning approaches are especially suited for problems with limited but diverse data.

5. Conclusion

Legal references are specialised entities relevant to the specific domain of legal texts. Standard NER approaches, such as the work of Glaser [19], focus on more common entities like Location, Organisation and so on. The results for these common entities are not comparable to the results of domain specific entities such as references to legal documents. Using specialized NER approaches has the potential to achieve more reliable solutions.

NER helps to semantically enrich legal documents. A reliable automation of this task enables more sophisticated information systems. The obtained results suggest that a reliable solution may be accomplished by extending existing data sets and considering transfer learning methods which have proven successful for similar tasks.

References

[1] Al Qundus J, Paschke A, Gupta S, Alzouby AM, Yousef M. Exploring the impact of short-text complexity and structure on its quality in social media. Journal of Enterprise Information Management. 2020.

[2] Hoppe T, Al Qundus J, Peikert S. Ontology-based Entity Recognition and Annotation. In: Proceedings of the Conference on Digital Curation Technologies (Qurator 2020); 2020. Available from: http://ceur-ws.org/Vol-2535/paper_4.pdf.

[3] Al Qundus J. Generating trust in collaborative annotation environments. In: Proceedings of the 12th international symposium on open collaboration companion; 2016. p. 1-4.

[4] Lafferty J, McCallum A, Pereira FC. Conditional random fields: Probabilistic models for segmenting and labeling sequence data; 2001. .
[5] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks. 2005;18(5-6):602-10.
[6] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Bidirectional Encoder Representations from Transformers; 2016. .
[7] Oksanen A, Tamper M, Tuominen J, Mäkelä E, Hietanen A, Hyvönen E. Part III, Article No. 2. In: Perugunelli G, Faro S, editors. Semantic Finlex: Transforming, Publishing, and Using Finnish Legislation and Case Law As Linked Open Data on the Web. No. 317 in Frontiers in Artificial Intelligence and Applications. Netherlands: IOS PRESS; 2019. p. 212-28.
[8] Glaser I, Schamberger T, Matthes F. Anonymization of German Legal Court Rulings. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. New York, NY, USA: Association for Computing Machinery; 2021. p. 205–209. Available from: https://doi.org/10.1145/3462757.3466087.
[9] C Cardellino LAA M Teruel, Villata S. A Low-cost, High-coverage Legal Named Entity Recognizer, Classifier and Linker. In: Proc. of the 16th International Conference on Artificial Intelligence and Law (ICAIL-2017).; 2017. p. 9-18. Available from: https://hal.archives-ouvertes.fr/hal-01541446v1.
[10] Song F, De La Clergerie E. Clustering-based Automatic Construction of Legal Entity Knowledge Base from Contracts. In: Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020; 2020. p. 2149-52.
[11] Tamper M, Oksanen A, Tuominen J, Hietanen A, Hyvönen E. Automatic Annotation Service APPI: Named Entity Linking in Legal Domain. In: Harth A, Presutti V, Troncy R, Acosta M, Polleres A, Fernández JD, et al., editors. The Semantic Web: ESWC 2020 Satellite Events. Cham: Springer International Publishing; 2020. p. 208-13.
[12] Adedjouma M, Sabetzadeh M, Briand LC. Automated detection and resolution of legal cross references: Approach and a study of Luxembourg's legislation. In: 2014 IEEE 22nd International Requirements Engineering Conference (RE); 2014. p. 63-72.
[13] Shulayeva O, Siddharthan A, Wyner A. Recognizing cited facts and principles in legal judgements. Artificial Intelligence and Law. 2017;25(1):107-26. Cited By :31.
[14] Resck Domingues LE, Ponciano JR, Nonato LG, Poco J. LegalVis: Exploring and Inferring Precedent Citations in Legal Documents. IEEE Transactions on Visualization and Computer Graphics. 2022.
[15] Correia FA, Almeida AAA, Nunes JL, Santos KG, Hartmann IA, Silva FA, et al. Fine-grained legal entity annotation: A case study on the Brazilian Supreme Court. Information Processing and Management. 2022;59(1).
[16] Sadeghian A, Sundaram L, Wang DZ, Hamilton WF, Branting K, Pfeifer C. Automatic Semantic Edge Labeling over Legal Citation Graphs. Artif Intell Law. 2018 jun;26(2):127–144. Available from: https://doi.org/10.1007/s10506-018-9217-1.
[17] Galgani F, Compton P, Hoffmann A. Citation Based Summarisation of Legal Texts. In: Anthony P, Ishizuka M, Lukose D, editors. PRICAI 2012: Trends in Artificial Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012. p. 40-52.
[18] Glaser I, Scepankova E, Matthes F. Classifying semantic types of legal sentences: Portability of machine learning models. In: Legal Knowledge and Information Systems. IOS Press; 2018. p. 61-70.
[19] Glaser I, Waltl B, Matthes F. Named entity recognition, extraction, and linking in German legal contracts. In: IRIS: Internationales Rechtsinformatik Symposium; 2018. p. 325-34.
[20] Zöllner J, Sperfeld K, Wick C, Labahn R. Optimizing small BERTs trained for German NER. arXiv preprint arXiv:210411559. 2021.
[21] Donnelly J, Roegiest A. The Utility of Context When Extracting Entities From Legal Documents. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. CIKM '20. New York, NY, USA: Association for Computing Machinery; 2020. p. 2397–2404. Available from: https://doi.org/10.1145/3340531.3412746.
[22] Leitner E, Rehm G, Moreno-Schneider J. Fine-grained Named Entity Recognition in Legal Documents. In: Acosta M, Cudré-Mauroux P, Maleshkova M, Pellegrini T, Sack H, Sure-Vetter Y, editors. Proceedings of the 15th International Conference on International Conference on Semantic Systems. No. 11702 in Lecture Notes in Computer Science. Karlsruhe, Germany: Springer; 2019. p. 272-87. 10/11 September 2019.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220473

An End-to-End Pipeline from Law Text to Logical Formulas

Aarne RANTA [a,1], Inari LISTENMAA [b], Jerrold SOH [b], Meng Weng WONG [b]
[a] *Chalmers University of Technology and University of Gothenburg*
[b] *Centre for Computational Law, Singapore Management University*

Abstract. We propose a pipeline for converting natural English law texts into logical formulas via a series of structural representations. Text texts are first parsed using a formal grammar derived from light-weight annotations. An intermediate representation called assembly logic is then used for logical interpretation and supports translations to different back-end logics and visualisations. The approach, while rule-based and explainable, is also robust: it can deliver useful results from day one, but allows subsequent refinements and variations.

Keywords. legal formalisms, legal text parsing, Grammatical Framework

1. Introduction

Expressing laws computably is a classic objective of AI & Law [1] and a prerequisite to automating downstream tasks such as compliance checking [2], policy support [3], legislative simulation [4], and formal verification [3]. But faithfully translating law to logic is challenging [5], often requiring expertise in both legal and formal methods. This "natural language barrier" [6] poses a significant "knowledge bottleneck" [7] to computational law. Numerous strategies have been devised for bridging the gap. These include domain-specific ontologies [8], intermediate formalisms [6], and specialised human workflows [8,9]. Early on, [10] had already imagined automatic parsers for translating laws into logic. Several steps have been taken towards that vision. McCarty [6] used [11]'s statistical parser to extract from judicial opinions syntax trees then converted into semantic representations. [12] extract formal rules from deontically- and structurally-annotated legal texts with the standard NLP parsers, while [5] experiment with neural semantic parsing and open relation extraction.

However, whether the chosen framework accommodates the logic representation desired is not always clear [13]. This paper contributes a partially-automated

[1]Corresponding Author: Aarne Rante, aarne.ranta@cse.gu.se. This research is supported by the National Research Foundation (NRF), Singapore, under its Industry Alignment Fund — Pre-Positioning Programme, as the Research Programme in Computational Law. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

law to logic pipeline based on Grammatical Framework (GF, [14]). While prior legal GF applications [15,16] focused on Controlled Natural Languages (CNL, e.g., [17,18]), our application tackles real-world law texts, albeit still exploiting key GF features such as modularity, precision, and support for semantic back-ends via an abstract syntax. We develop a method for automatically extracting a grammar from light-weight annotations which non-experts can create. This grammar is usable as-is for a rough analysis of law texts but can also be manually improved. Our code is available open-source.[2]

2. Methodology

The initial input is a statutory text in natural language which we assume has been tokenised by some standard tool. The tokenised text is converted to abstract syntax trees (ASTs) line by line using the GF parser driven by a grammar (Section 2.1). The ASTs are converted into an intermediate representation called assembly logic (Section 2.2) using the Haskell-based methodology from [19]. Assembly logic is more abstract than ASTs from the parser, but preserves more distinctions than standard back-end logics. These distinctions are useful for deriving different output formats such as downstream logics and visualisations (Section 2.3).

2.1. From Law Text to ASTs

GF parsers are driven by grammars, such as GF's general-purpose Resource Grammar Library (RGL, [20]). However, the RGL is insufficient for law texts, which contain special constructs that are essential for the logical structure, such as itemised lists and indented paragraphs. Thus we developed a tailored grammar on top of the RGL. Figure 1 illustrates the grammar building workflow, which adopts a data-driven, top-down approach starting from the text itself. We developed a semi-automated method for grammar writing based on user annotations. The annotations are based on the RGL's grammatical categories (e.g. `NP`, `VP`, `VP2`, `CN`) which are well-known in NLP. Annotators may further specify their own categories, such as `Line`, `Item` and `Ref`, and any categories added will be added to the custom grammar. Completely novel categories *could* be created, but if they deviate too much from the RGL, the resulting grammar cannot leverage the RGL as much. Finally, a Haskell script generates GF rules from the annotated text. The rules are generated in a context-free format that GF can process.

2.2. From ASTs to Assembly Logic

Assembly logic is an intermediate representation between ASTs and standard logics. It is designed to preserve enough syntactic structure to generate representations that humans can easily relate to the original text. For example, it distinguishes between ordinary and reverse implications ("if A then B" vs. "B if A") and preserves quantified noun phrases as units (e.g. "'any organisation"). It is

[2] https://github.com/smucclaw/sandbox/tree/default/aarne#readme

A line in the raw text:

(2) without limiting subsection (1)(a), a data breach is deemed to result in significant harm to an individual —

The line annotated with marks for terminals (#) and nonterminals (*):

```
*Item (2) #without #limiting #subsection *Ref (1)(a) #,
#a *CN data breach #is #deemed #to
*VP result in significant harm to an individual #-
```

Grammar rules derived automatically by the script:

```
Line ::= Item "without" "limiting" "subsection" Ref ","
         "a" CN "is" "deemed" "to" VP "-" ;
Item ::= "(2)" ;
Ref  ::= "(1)(a)" ;
CN   ::= "data" "breach" ;
VP   ::= "result" "in" "significant"
         "harm" "to" "an" "individual" ;
```

The VP rule above refined into more general rules:

```
VP2 ::= "result" "in" NP ;
NP  ::= "significant" "harm" "to" NP ;
NP  ::= "an" CN ;
CN  ::= "individual" ;
```

Figure 1. The grammar extraction process.

intended to be sufficient for this purpose, so that when the grammar is extended (e.g., new law texts), the assembly logic and its back-ends can be kept constant.

Figure 2 shows a sample of the assembly logic implemented as a Haskell datatype `Formula`. It also shows part of an **interpretation function** [19], `iNP`, which converts ASTs of GF type `NP` (Noun Phrase) to assembly logic. These functions use pattern matching over trees. Each AST constructor may have its own pattern, such as for `NP_any_CN` in Figure 2. When the grammar is extended, new patterns can be added. But even if this is not done, the function can take care of the new constructors by the catch-all case (`_`) which treats the new expressions as atomic. Atomic expressions can then be converted to atomic formulas or constants in logics and to single cells in spreadsheets (see Section 2.3).

2.3. From Assembly Logic to Downstream Logics or Visualisations

Assembly logic is mapped into many-sorted logic and then into ordinary predicate logic in TPTP notation [21]. We use many-sorted logic as it better supports compositional translation. Quantification expressed by noun phrases (e.g. "any organisation") are compositionally interpreted as quantifiers with sorts rather than divided into unsorted quantifiers and sort predicates, whereas definite noun phrases (e.g. "that organisation") are interpreted as Russell's iota terms (we write $\iota(A)$ instead of $(\iota x)A(x)$, leaving possible variable bindings to A itself, as is customary in higher-order logic). Both sorted quantifiers and iota terms are eliminated in the conversion from many-sorted to ordinary predicate logic. Iota terms are eliminated in a pass that looks for non-iota terms in their context of use.

Some assembly logic constructors:

```
data Cat =
  CProp | CSet | CInd | ...
data Formula =
  Atomic Cat Atom
| Implication Formula Formula
| Conditional Formula Formula     -- reverse implication
| Quantification String Formula   -- quantifier + domain
```

Semantics of ASTs in the assembly logic:

```
iNP :: Env -> NP -> Formula
iNP env np = case np of
  NP_any_CN cn -> Quantification "ANY" (iCN env cn)
  NP_each_CN cn -> Quantification "EACH" (iCN env cn)
  ...
  _ -> Atomic CInd (toAtom env np) -- convert to string
```

Figure 2. Data structures and conversions related to the assembly logic.

Below is a minimal example, with an existential quantifier in the antecedent and definite noun phrase referring to it in the succedent: "if a notification is a data breach, the notification is affected". Its compositional interpretation in many-sorted logic with iota terms is $(\exists x : \mathit{notification})\mathit{data\_breach}(x) \supset \mathit{affected}(\iota(\mathit{notification}))$. When converted to ordinary predicate logic, the existential quantifier is changed into a universal one with a wide scope of implication, and the iota term is interpreted as the bound variable:

```
![X]:(notification(X) => data_breach(X) => affected(X))
```

The AST can also be automatically visualised in a spreadsheet (see Figure 3) displaying the formula trees in a structured format. The spreadsheet format, which serves as the input to a low-code programming platform, is currently under development and will be more fully described in future work.

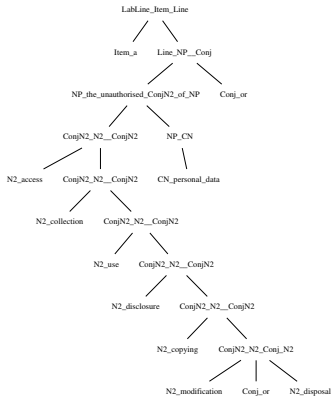
3. Formalizing the Personal Data Protection Act

We illustrate our pipeline using Part 6A of the PDPA, which comprises 47 lines, 1053 tokens, 228 unique tokens. Figure 3 below illustrates the pipeline as applied to one paragraph. The PDPA is Singapore's primary data protection statute and Part 6A governs data breach notifications. While the PDPA has not been examined in AI & Law literature, its subject matter connects it to prior work on the General Data Protection Regulation [8,2]. Part 6A is also complex enough to demonstrate the utility of a computational law approach. Modelling these rules surfaced a race condition in the PDPA: an organisation which promptly notifies both the regulator and the affected individuals of a data breach, as s 26D PDPA generally requires, might violate s 26D(6) which provides that organisations should *not* inform affected individuals if the regulator so directs. A more complete formalism of Part 6A can be found on our code repository.

A paragraph in the raw text:

> "data breach", in relation to personal data, means —
>
> (*a*) the unauthorised access, collection, use, disclosure, copying, modification or disposal of personal data; or
>
> (*b*) the loss of any storage medium or device on which personal data is stored in circumstances where the unauthorised access, collection, use, disclosure, copying, modification or disposal of the personal data is likely to occur.

AST of line (a) and spreadsheet visualization of the paragraph



Logical formula in TPTP notation:

```
![X]:(data_breach(X) & ?[Y]:(personal_data(Y) & IN_RELATION_TO(X,Y)) <=>
(personal_data(X) & ?[Y]:((access(Y,X) | collection(Y,X) | use(Y,X) | disclosure(Y,X) |
copying(Y,X) | modification(Y,X) | disposal(Y,X)) & unauthorized(Y))) | (((storage_medium(X) |
device(X)) & (personal_data(X) & ?[Y]:((circumstances(Y) & (((unauthorized(Y) & (access(Y) |
collection(Y) | use(Y) | disclosure(Y) | copying(Y) | modification(Y) | disposal(Y))) &
is_likely_to_occur(Y))) & is_stored_in(X,Y)))) & loss(X))))
```

Figure 3. An example through the pipeline

4. Conclusion

This paper proposed a pipeline which parses legal text into ASTs using the GF grammar formalism, an intermediate assembly logic, and finally predicate logic. Some pipeline steps can work out of the box when the input scope is extended. The main things to be added are text annotations for extending the grammar, and the conversion of the new grammar rules to the assembly logic. These steps are light-weight enough to make the system feasible to apply to new texts. Further, since GF's mapping between ASTs and natural language is fully reversible, the pipeline can be extended to support natural language generation. Once parsed into GF trees, the source text can be converted into novel forms: declarative sentences can become questions, negations, hypotheticals, etc. That said, this work is a proof of concept and has a some limitations. Importantly, we have not evaluated the accuracy of our PDPA formalisation and aim to do so in future work. A proper evaluation would implicate gold standards developed by human legal and technical experts and vetted by the relevant regulatory body. The legal

language barrier is far from solved, but we hope to have taken one more step towards realising that vision.

References

[1] Sergot MJ, Sadri F, Kowalski RA, Kriwaczek F, Hammond P, Cory HT. The British Nationality Act as a logic program. Communications of the ACM. 1986 May;29(5):370-86.

[2] Hickey D, Brennan R. A GDPR International Transfer Compliance Framework Based on an Extended Data Privacy Vocabulary (DPV). In: Proceedings of JURIX. IOS Press; 2021. p. 161-70.

[3] Haan ND. TRACS: A Support Tool for Drafting and Testing Law. In: Proceedings of JURIX; 1992. p. 63-70.

[4] Bench-Capon TJM. Support for Policy Makers: Prospects for Knowledge Based Systems. In: Proceedings of JURIX; 1992. p. 41-50.

[5] Ferraro G, Lam HP, Tosatto SC, Olivieri F, Islam MB, Beest Nv, et al. Automatic extraction of legal norms: Evaluation of natural language processing tools. In: JSAI International Symposium on Artificial Intelligence. Springer; 2019. p. 64-81.

[6] McCarty LT. Deep semantic interpretations of legal texts. In: Proceedings of ICAIL; 2007. p. 217-24.

[7] Nazarenko A, Lévy F, Wyner A. A Pragmatic Approach to Semantic Annotation for Search of Legal Texts – An Experiment on GDPR. In: Proceedings of JURIX. IOS Press; 2021. p. 23-32.

[8] Palmirani M, Martoni M, Rossi A, Robaldo L. Legal Ontology for Modelling GDPR Concepts and Norms. In: Proceedings of JURIX; 2018. p. 91-100.

[9] Witt A, Huggins A, Governatori G, Buckley J. Converting copyright legislation into machine-executable code: interpretation, coding validation and legal alignment. In: Proceedings of ICAIL. São Paulo Brazil: ACM; 2021. p. 139-48.

[10] Bing J. Designing text retrieval systems for conceptual searching. In: Proceedings of ICAIL. Boston, Massachusetts, United States: ACM Press; 1987. p. 43-51.

[11] Collins M. Head-Driven Statistical Models for Natural Language Parsing. Computational Linguistics. 2003 Dec;29(4):589-637.

[12] Dragoni M, Villata S, Rizzi W, Governatori G. Combining natural language processing approaches for rule extraction from legal documents. In: AI Approaches to the Complexity of Legal Systems. Springer; 2015. p. 287-300.

[13] Wyner A, Governatori G. A Study on Translating Regulatory Rules from Natural Language to Defeasible Logic. In: Proceedings of the 7th International Web Rule Symposium; 2013. p. 16.1-16.8.

[14] Ranta A. Grammatical Framework: Programming with Multilingual Grammars. Stanford: CSLI Publications; 2011.

[15] Angelov K, Camilleri J, Schneider G. A Framework for Conflict Analysis of Normative Texts Written in Controlled Natural Language. The Journal of Logic and Algebraic Programming. 2013;82:216-40.

[16] Digital Grammars, Signatu. GDPR Lexicon; 2018. https://gdprlexicon.com/.

[17] Fuchs NE, Kaljurand K, Kuhn T. Attempto Controlled English for Knowledge Representation. In: Reasoning Web, Fourth International Summer School 2008. 5224. Springer; 2008. p. 104-24.

[18] Ranta A, Angelov K. Implementing Controlled Languages in GF. In: Proceedings of CNL-2009, Marettimo. vol. 5972 of LNCS; 2010. p. 82-101.

[19] Ranta A. Translating between Language and Logic: What Is Easy and What Is Difficult. In: Automated Deduction – CADE-23. Springer Berlin Heidelberg; 2011. p. 5-25.

[20] Ranta A. The GF Resource Grammar Library. Linguistics in Language Technology. 2009;2.

[21] Sutcliffe G. The TPTP problem library and associated infrastructure. Journal of Automated Reasoning. 2009;43(4):337-62.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220474

Legal Text Summarization Using Argumentative Structures

Bianca STEFFES [a,1], Piotr RATAJ [a,b]
[a] *Saarland University, Saarland Informatics Campus, Saarbrücken, Germany*
[b] *Zentrum für Recht und Digitalisierung, Saarbrücken, Germany*

Abstract. Legal text summarization focuses on the automated creation of summaries for legal texts. We show that the argumentative structure of judgments can improve the selection of guiding principles as a specific kind of summary using judgments of the German Federal Court of Justice as measured by the ROUGE metric. We evaluate our first results and put them in the context of our ongoing work.

Keywords. legal text summarization, German Federal Court of Justice, guiding principles, extractive summarization

1. Introduction

Text summarization algorithms allow automated creation of summaries for arbitrary texts. Especially the legal domain, in which long and complex documents are ubiquitous, might benefit from such algorithms on a large scale. Yet the summarization of legal documents is still confronted with some unsolved problems: the domain specific knowledge and structure seems to hinder simple porting of domain independent summarization algorithms to the legal field (e.g. [1], [2]) and the sheer length of documents and sentences challenges neural models (e.g. [3]). In this work we tackle the summarization of judgments delivered by the German Federal Court of Justice (Bundesgerichtshof, BGH) by automatically selecting guiding principles of the judgments.

1.1. Character of Guiding Principles

Guiding principles (*Leitsätze*) are, roughly speaking, very short formulations of or at least introductions to the main and "important" normative statement(s) that a court finds when deciding a particular case. Such a principle could be: "A will is invalid when written on a computer.". Their function is to quickly brief the reader and provide some orientation regarding the judgment. Although guiding principles are issued by many (German) courts, we only take into account judgments of the BGH in civil matters that contain such guiding principles (which is not the case for all of its judgments). As the BGH is a court of final instance and, thus, judicial review is limited to (important) issues concerning the

[1] Corresponding Author: Bianca Steffes, bianca.steffes@uni-saarland.de.

interpretation of the law (versus the facts) we assume that, in most cases, a large part of the court´s reasoning will be somehow reflected in its guiding principles.

Note that guiding principles, at least the ones that we pick, are issued directly by the court's body that decides the case. This guarantees a high quality of the data. With respect to their content in detail, however, there are no formal rules. In practice, we observe two basic forms in which guiding principles are stated by the court and label them as *topical* and *propositional* respectively: while the former simply introduce the decision's main legal topic (e.g., "On the conditions of a valid will when written on a computer."), the latter contain a normative statement (as is the case with our example from above: "A will is invalid when written on a computer."). We only take into account propositional ones.

Since guiding principles contain the main statement(s) regarding the court´s interpretation of the law they reflect the normative conclusion of its argumentation. On a structural level, this implies that only a specific part of the—rigorously structured—judgment needs to be considered (see further *infra* 4.1.). Thematically, this means that the guiding principles concern a different order than the argumentation justifying them. Furthermore, the guiding principles as well as the argumentation refer to the facts of the case mostly indirectly, allowing us to focus on the normative / legal domain and its language. With this in mind we can think of guiding principles as a specific kind of legal summary of a judgment.

1.2. Related Work

Existing algorithms for text summarization can be roughly grouped into two classes: abstractive and extractive approaches. Abstraction-based algorithms create summaries by paraphrasing the content of a text, while extraction-based methods select sentences from the original document as a summary. In the legal domain, extractive algorithms are most common.

LetSum [4] and *DelSumm* [5] are extractive algorithms that create summaries by, firstly, mapping all sentences to structural parts of a judgment (e.g., facts, reasoning) with the aim of representing each of these parts in their summaries. The sentences which will later constitute the summary are then selected by a tf*idf based ranking and a specific scoring. Both of these categorizations are not applicable to our problem as, for one, they categorize the sentences of a whole judgments while German judgments are published in a similar categorization already, and on the other hand, we intend to work on only one of those categories (reasoning) which they do not provide a structuring for. The scoring of *DelSumm* is highly based on the mentioned structuring, thus, it does not fit our problem.

The *MMR algorithm* [6] selects the sentences for the summary based on how predictive they are for the outcome of the case, e.g, which party wins. It uses an iterative selection process to pre-select particularly predictive sentences and creates the final summary based on this subset using Maximum Marginal Relevance. As guiding principles do not, as such, give an indication to the outcome of a case, such a selection would not be helpful.

A simpler approach is implemented by *CaseSummarizer* [7] which selects sentences mostly based on tf*idf, the information on whether a sentence is at the beginning of a paragraph and the occurrences of dates, and known entities in the sentences. Using neural networks for summarization tasks allows extractive summaries, as shown, e.g., by the Chinese *GIST* [8] using different ensemble models, as well as abstractive results by, e.g.,

fine-tuning pre-trained language models like *BERT2BERT* or *BART* [3]. Unfortunately, such approaches are in need of a high amount of data which is hardly available in the German legal field.

Compared to existing work, our approach allows us to work with the limited judgment data available in Germany and operates on a completely different level of structuring of judicial decisions than previous work. Thus we contribute to the current research by showing preliminary results on text summarization of judgments passed by the BGH. To our knowledge, little work on summarization tasks has been done on German judgments so far. As previous work may only be partially applicable, we show that the integration of the argumentative structure achieves significantly higher ROUGE-scores than our baseline on judgments of the BGH.

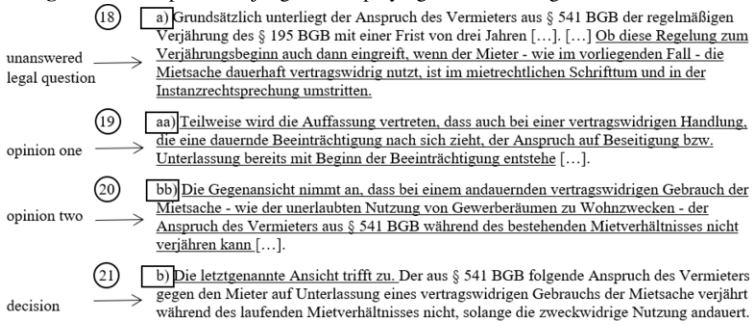
2. Working with the Argumentative Structure

To present our results we will first give an introduction to how Section II of the legal grounds of a judgment (*Entscheidungsgründe*) of the BGH is structured. Then, we will explain the data that we worked on, present our approach, and, finally, evaluate our preliminary results.

2.1. Argumentative Structure

Legal documents and especially judgments are highly structured texts. The reasoning concerning the legal grounds of a decision basically follows a treelike structure: The relevant points of law are each addressed and then discussed in more detail, one point after the other. Deeper levels contain further elaboration on the respective legal aspect, discuss sub-questions or give differing opinions of lower courts and literature. An example can be found in Figure 1.[2]

Figure 1. Excerpt from a judgment displaying the treelike argumentative structure



The example already shows a variety of different structuring elements: The numbers in circles at the left hand side depict the consecutive numbering of paragraphs (*Randnummern*) of the judgment. The corresponding paragraphs contain logical units of the text. The listing in the rectangles at the beginning of the paragraphs indicate the argumentative

[2]The judgment can be found at juris.bundesgerichtshof.de/cgi-bin/rechtsprechung/document.py?Gericht=bgh&nr=91902 and an English image version at legalinf.de/jurix22.

structuring level; elements of the listing might contain several paragraphs. Elements of the same structuring level are of equal abstraction level and oftentimes continue the line of reasoning. In case of our example above, an—from the BGH's perspective—open legal question is stated at the end of paragraph 18. The following paragraphs 19 and 20 (of the lower structuring level) present different opinions on this question as derived from lower court decisions. In paragraph 21 we return to the higher structuring level again and the BGH decrees which of these opinions is, according to them, correct. This example already shows how the reasoning and the argumentative structure can give an indication for finding the guiding principles: The last sentence in paragraph 18 gives the explicit concluding decision of the court and is the guiding principle of this particular judgment.

2.2. Data

We investigate whether the argumentative structure of a judgment allows us to select the sentences for the guiding principles. Therefore, we inspected existing judgments of the BGH and gathered 100 judgments with extractive guiding principles already formulated by the court.[3] For all of these 100 judgments we determined the exact positions of the respective guiding principles with respect to the argumentative structuring. To avoid overfitting, we used another set of 100 judgments as an unknown test set for the validation of our results. The judgments in this test set did not contain extractive guiding principles but abstractive ones. The reason we chose such a dataset as a test set is that most existing judgments contain abstractive guiding principles. Note that we measure our resulting summaries with the ROUGE metric. As the judgments used for our analysis contain the exact sentences of the guiding principles, a perfect summarization algorithm may reach a ROUGE-score of 1. This is impossible as regards the remaining test set as it does not contain sentences that are syntactically identical to the guiding principles. Therefore, it is only natural that the ROUGE-scores of the results on these judgments are lower than on the other 100 cases.

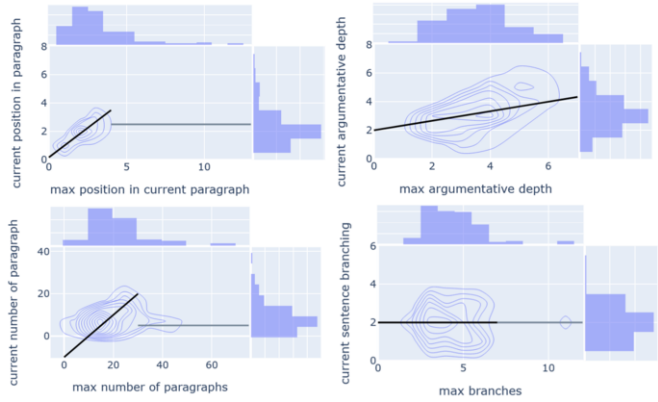
2.3. Ranking Sentences based on Aspects of the Argumentative Structure

In regards to the argumentative structuring as a treelike structure, we focused on the position of a sentence in the paragraph, the argumentative depth of the sentence in the tree, the number of the paragraph of the sentence (counted from the beginning of Section II of the legal grounds), and the number of the branch with respect to one parent node. To get further insight into the impact of these features on the guiding principles, we considered the extractive guiding principles and their corresponding values of these features in relation to their maximum values in these judgments. An illustration can be found in Figure 2 (density plots) which shows, e.g., that the guiding principles in our judgments were mostly found in a branch number close to 2 (density is highest, lower right image). The histograms give an indication of the data distribution of the features (e.g. branch number, at the right) and the maximum of that feature in the judgment (e.g. maximum branch number in the judgment, at the top).

Based on these insights we derived linear functions to approximate the optimal values for a sentence to be chosen as a guiding principle (lines in the images). Except for the feature *depth in the argumentative structure*, we had a long tail in the distributions of the

[3]The cases were accessed at www.rechtsprechung-im-internet.de.

Figure 2. Parameter distributions and approximation functions for the 100 extractive guiding principles



maximum values. Therefore, we decided to use different partially defined approximation functions (gray lines) for the tails as the knowledge derived from this data is much less reliable. In case of the branch numbers, this resulted in the same function for both parts.

For our actual ranking of sentences, we first pre-processed the judgments by removing stopwords and sentences without at least one verb in present tense (as guiding principles are always written in present tense), as well as lemmatizing and case normalizing the words. Similar to *CaseSummarizer* [7], we then calculated a ranking for each sentence as the sum of tf*idf values of its words and normalizing the score by the length of the sentence ($rank_{tfidf}$). Then, we calculated the final ranking by the following formula: $rank_{final} = rank_{tfidf} + \sigma * (p_d * f_d(d, max_d) + p_p * f_p(p, max_p) + p_n * f_n(n, max_n) + p_b * f_b(b, max_b))$ with σ the standard deviation of $rank_{tfidf}$ and p_d, p_p, p_n and p_b tunable parameters for the argumentative depth (d), position in paragraph (p), number of paragraph (n) and the number of branching (b). The functions f_d, f_p, f_n and f_b calculate the shortest distance of the current value (e.g., of the argumentative depth of a sentence) to the derived functions of the most likely values. The final selection of the ranked sentences is done by first selecting the sentence with the highest score and then adding as many sentences from the top of the ranking until the selection has a length of approx. 2.47% of the original judgment, which was the average length of guiding principles compared to the original judgment in a set of 5000 judgments.

2.4. Evaluation

For our evaluation, we compared our results to a random selection of sentences, a ranking using only tf*idf values and *CasesSummarizer* (Table 1). We distinguished between the results concerning the judgments containing abstractive and extractive guiding principles and optimized the parameters of our approach separately for each of these datasets.

As expected, the ROUGE values on the abstractive judgments were significantly lower than in the extractive judgments and the optimized formula for the extractive version is by no means optimal for the abstractive judgments and vice versa. Compared to a random ranking and a simple tf*df based ranking we could significantly increase the ROUGE score of the results. Especially in comparison to *CaseSummarizer* we constantly achieved high ROUGE-L scores, which indicates that we are (more) successfully able to identify sentences close to the meaning of the original guiding principles.

Table 1. Evaluation results using ROUGE metric

	extractive judgments		abstractive judgments	
	ROUGE-1	ROUGE-L	ROUGE-1	ROUGE-L
random ranking	0.2259	0.1950	0.1403	0.1751
tf*idf ranking	0.2874	0.3280	0.1208	0.1775
CaseSummarizer	0.3886	0.2520	0.2310	0.1890
our work (optimized on extractive dataset)	0.4134	0.4141	0.1696	0.2163
our work (optimized on abstractive dataset)	0.3596	0.3643	0.1881	0.2232

3. Conclusion and Future Work

We found that our approach of using the argumentative structure of the judgments seems to be indeed a fruitful starting point for creating guiding principles in the case of the BGH. Integrating the argumentative structure in a ranking for extractive summarization achieves higher results than our baseline and performs especially well in the ROUGE-L metric.

In our ongoing work we intend to further compare our approach to other existing algorithms and make use of semantics to determine whether they relate to the argumentative structure. Furthermore, we plan to extend our analysis to abstractive guiding principles and their relation to the argumentative structure. Other aspects, like highly recurrent terms, might also increase the ROUGE-score of selections.

References

[1] Deroy A, Bhattacharya P, Ghosh K, Ghosh S. An Analytical Study of Algorithmic and Expert Summaries of Legal Cases. In: Legal Knowledge and Information Systems; 2021. p. 90-9.

[2] Bhattacharya P, Hiware K, Rajgaria S, Pochhi N, Ghosh K, Ghosh S. A Comparative Study of Summarization Algorithms Applied to Legal Case Judgments. In: Advances in Information Retrieval. Cham: Springer International Publishing; 2019. p. 413-28.

[3] Yoon J, Junaid M, Ali S, Lee J. Abstractive Summarization of Korean Legal Cases using Pre-trained Language Models. In: 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM); 2022. p. 1-7.

[4] Farzindar A, Lapalme G. LetSum, an automatic Legal Text Summarizing system. In: Legal Knowledge and Information Systems, Jurix 2004: The Seventeenth Annual Conference; 2004. p. 11-8.

[5] Bhattacharya P, Poddar S, Rudra K, Ghosh K, Ghosh S. Incorporating Domain Knowledge for Extractive Summarization of Legal Case Documents. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. ICAIL '21. New York, NY, USA: ACM; 2021. p. 22–31. Available from: https://doi.org/10.1145/3462757.3466092.

[6] Zhong L, Zhong Z, Zhao Z, Wang S, Ashley KD, Grabmair M. Automatic Summarization of Legal Decisions Using Iterative Masking of Predictive Sentences. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. ICAIL '19. New York, NY, USA: ACM; 2019. p. 163–172. Available from: https://doi.org/10.1145/3322640.3326728.

[7] Polsley S, Jhunjhunwala P, Huang R. CaseSummarizer: A System for Automated Summarization of Legal Texts. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations. Osaka, Japan: The COLING 2016 Organizing Committee; 2016. p. 258-62. Available from: https://aclanthology.org/C16-2054.

[8] Liu CL, Chen KC. Extracting the Gist of Chinese Judgments of the Supreme Court. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. ICAIL '19. New York, NY, USA: ACM; 2019. p. 73–82. Available from: https://doi.org/10.1145/3322640.3326715.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220475

Measuring the Complexity of Dutch Legislation

Tim VAN DEN BELT [a] and Henry PRAKKEN [b]
[a] *Faculty of Law, University of Groningen, The Netherlands*
[b] *Department of Information and Computing Sciences, Utrecht University & Faculty of Law, University of Groningen, The Netherlands & European University Institute, Florence, Italy*

Abstract. For legislation to be effective, it should not be too complex; otherwise, it cannot be sufficiently understood by those who have to apply the law or comply with it. This paper adds to the research in AI & law on developing precise mathematical complexity measures for legislation and applying these measures by computational means. The framework of Katz & Bommarito (2014) is applied to measure the complexity of Dutch legislation. The aim is twofold: first, to investigate whether this framework is meaningfully more widely applicable by applying it to a different jurisdiction and a corpus of larger size; and second, to identify possible improvements to the framework.

Keywords. legislation, complexity analysis, Dutch law

1. Introduction

For legislation to be effective, it should not be too complex; otherwise, it cannot be sufficiently understood by those who have to apply the law or comply with it. In law and politics, the desire to constrain the complexity of legislation is often discussed but these discussions could benefit from precise measures of the complexity of the legislation. Accordingly, in AI & law research exists on developing precise mathematical complexity measures for legislation and applying these measures by computational means [1,3,2,5]. The hope underlying this research is that it will aid academic and policy discussions about the complexity of the law, resulting in more accessible and understandable legislation.

Bourcier and Mazzega [1] made a distinction between *structure-based* and *content-based* measures of complexity and discussed some possible measures of these kinds. Waltl and Matthes [5] applied several quantitative metrics of these kinds for analysing the complexity of German law. Katz & Bommarito [2] refined Bourcier and Mazzega's classification into *structure-*, *language-* and *interdependence-based* complexity measures. They then proposed a comprehensive computational framework in which several such measures are combined into an overall measure of the complexity of the legislation. They then applied the framework to measure the complexity of the *United States Code*. The framework was

"motivated by the specific contours of the United States Code", but the authors hypothesised that it is more widely applicable.

Accordingly, this paper presents an application of the framework of [2] to measure the complexity of Dutch legislation. The aim of this is twofold. First, we want to investigate Katz & Bommarito's [2] hypothesis that their framework is meaningfully more widely applicable by applying it to a different jurisdiction, a different language and a corpus of larger size. A second aim is to identify possible additions to or improvements of their framework as used in [2].

To summarise our findings, we found that the framework of Katz & Bommarito can be applied both mathematically and computationally in our corpus of Dutch legislation. However, we found reasons to recommend that complexity measures that strongly correlate with the structural size of legislation are less useful since they may be beyond the legislator's control.

The rest of this paper is organised as follows. In Section 2 we describe the corpus of Dutch legislation that was our study's object and summarise the way we applied Katz & Bommarito's framework to measure its complexity. In Section 3 analyse our complexity results and compare them with the results of Katz & Bommarito. We conclude in Section 4. The full extent of our analysis goes beyond a conference paper. Therefore, we can in this paper only present a summary of the data, method and results; the full details are available on Github.[1]

2. Corpus and Method

In this section we describe the corpus of Dutch legislation that was our study's object and summarise Katz & Bommarito's method of measuring complexity and our additions to and modifications of their method.

According to Katz & Bommarito, the United States Code is only a small portion of existing US law. By contrast, our data set consists of essentially the entire corpus of Dutch legislation limited to acts. To analyse its complexity, structured data is required. Our dataset consisted of a structured XML version of the corpus made available by KOOP, the knowledge and exploitation center for official Dutch government publications

The underlying idea of Katz & Bommarito's approach is that complexity can be measured using a knowledge acquisition process where someone wants to decide whether to comply with the law. This idea is operationalised into three features: *structure*, *interdependence* and *language* of legislation.

Structure The structure of a piece of legislation is represented as a tree, where the nodes represent the elements of the act and the links capture their hierarchical relations. For the Dutch legislation we distinguished the elements 'book', 'department', 'title', 'chapter', 'paragraph', 'subparagraph', 'section', 'subsection', 'sub'. This tree is then used to define two structure-based measures. *Structural size* is the number of nodes in the tree, while *Graph depth* (by Katz & Bommarito called *Element depth distribution*) is the mean distance of all nodes to the root of the tree. In addition to Katz & Bomarrito's, we also measured the element depth dis-

[1] github.com/TimvandenBelt/Complexity-Dutch-Legislation.

tribution of only the leaf nodes. We observed little difference in this measure by comparing the correlation and results, which differ by just 0.018. When ordering the results from highest to lowest, the ranking differs minimally.

Language Katz & Bommarito define the following measures in terms of the language of legislation. *Size* is the number of tokens within the text of an element. *Average word length* is the average number of characters of words in the text of an element (disregarding 'stop' words of several kinds). It should be noted here that, all other things being equal, average word length will be lower for English than for languages like Dutch and German, which combine words into single longer words. For instance, 'word length' translates to 'woordlengte' in Dutch. Finally, Katz & Bommarito use *Word entropy*, which informally measures the amount of textual variance of an element: does it use many different words and concepts, or is it homogeneous in these respects? They measure this in terms of the information-theoretic concept of Shannon entropy [4]. All other things being equal, the higher the word entropy of an element, the more complex it is. We also applied lemmatisation through the use of natural language processing.[2] The idea was that identical verbs or nouns might be used but in different forms and thus increasing the entropy. With lemmatisation, we morphed all words to their base form, providing, in our eyes, a better representation of the homogeneity of the text. However, we observed minimal differences with regular word entropy. In addition to the framework of Katz & Bommarito, we also use a measure of *Readability* of an element. For this, we use the so-called *Flesch reading ease* measure. It rates the readability of a text on a scale from 0 to 100, based on the average sentence length and the average number of syllables per word.[3] We use this measure as we believe it provides a more accurate representation of language complexity as it considers both word complexity and sentence complexity.

Interdependence Katz & Bommarito measure the interdependence within legislation in terms of the number of citations from one element to another. The higher the number of citations, the higher the complexity. Interdependence can be both internal (within an act) and external (between acts). Citations are represented in a directed citation graph, where the nodes are in [2] sections, and citations below-section nodes are attributed to section level from all 'titles' in the corpus, while in our case, they are at section level and below. The reason for this difference is that we believe that some below-section nodes may be of similar size or larger than some section nodes. We also believe it provides a more factual representation and may yield a more accurate network analysis. The links in the citation graph are citations from one element to another. Within-element citations in a title (in [2]) or act (in our analysis) are represented by subgraphs where all nodes are from the same title, respectively, act. Katz & Bommarito distinguish between explicit citations and the use of definitions from one element by another element. Due to time constraints and limits in the data, we have only considered explicit citations, excluding definitions. We measure *internal interdependence* of an act by counting the number of citations that cite another element in the same act.

[2]For which we used Spacy: https://spacy.io/.

[3]For detecting syllables, we used Spacy along with a community package: https://spacy.io/universe/project/spacy_syllables

We then normalise this against the structural size of the act by dividing the number of citations by the number of nodes in the hierarchical graph of that act. For measuring *external interdependence* between titles, Katz & Bommarito distinguish between titles exporting information (by being cited by another title) and titles importing information (by citing another title). They then measure the numerical difference ("net flow") between the number of imports and exports of a title. They also consider a normalised version "net flow per section" relative to title size. We apply the same methods to acts and their sections.

Waltl and Matthes [5] used several of the above-discussed measures, namely, section-nodes, number of words, element depth, internal interdependence, and a variation of external interdependence. In addition, they measured language complexity in terms of indeterminacy and vocabulary variety. Vocabulary variety can be compared to word entropy. Indeterminacy was outside our scope due to time constraints. Unlike [2] and us, [5] did not use a composite complexity measure.

Composite measures Katz & Bommarito then use these measures to define two composite measures. Both choose one measure from each of the three categories structure, language and interdependence. For their *unnormalised composite measure* they choose structural size, word entropy and net flow while for their *normalised composite measure* they choose mean element depth, word entropy and net flow per section. For both composite measures they then rank each title with each of these individual measures. Finally, they combine the three rankings thus obtained by computing the average rank of each title, acknowledging that other methods might be more suitable.

We used the same unnormalised composite measure, but we replaced word entropy with Flesch readability in their normalised composite measure. The reason for this is that, in our opinion, word entropy is not suitable for a normalised composite since it correlates too strongly with the size of the legislation.

3. Results & Analysis

We gauged each measure and calculated the correlation of most in relation to the structural size of legislation. Thereafter, just as [2], we used two composites to rank the legislation, with some minor adjustments. These results can be found on Github.[4] In total, 1120 acts were analysed.

In this section we analyse our results and compare them to those of Katz & Bommarito. As regards the normalised and unnormalised rankings, it is interesting to observe that as in [2], some acts rank similarly in these two rankings while for other acts there are considerable differences in rank (although still within the same region). Apart from this, an absolute comparison between [2] and our analysis on the various criteria is not very informative, because of the differences between the Dutch and English languages and the differences in legislation style between the Dutch and US jurisdictions. We, therefore, focus on correlation analysis. While Katz & Bommarito performed two correlation analyses, we did several more. Table 1 summarises our correlation results. We in particular investigated

[4] github.com/TimvandenBelt/Complexity-Dutch-Legislation.

Table 1. Correlation results ordered by highest to lowest R value.

Correlation	P value	R value	R squared %
Size & text nodes	0.000	0.998	99.68
Size & number of words	0.000	0.978	95.59
Size & number of tokens	0.000	0.977	95.48
Size & non-text nodes	0.000	0.973	94.65
Size & below-section nodes	0.000	0.964	92.95
Size & section nodes	0.000	0.961	92.37
Size & mean depth	0.000	0.926	85.82
Size & word entropy	0.000	0.925	85.58
Size & lemmatised word entropy	0.000	0.921	84.77
Size & citations total	0.000	0.919	84.42
Size & mean leaf depth	0.000	0.908	82.39
Size & internal citations	0.000	0.894	80.00
Sections & below-section nodes	0.000	0.870	75.64
Sections & above-section nodes	0.000	0.868	75.41
Size & above-section nodes	0.000	0.867	75.24
Size & external citations	0.000	0.843	71.13
Above-section & below-section nodes	0.000	0.795	63.31
Size & tokens per section	0.000	0.580	33.66
Size & net flow	0.000	0.408	16.66
Size & word length	0.000	0.111	1.24
Size & net flow per section	0.281	-0.032	0.10
Flesch & tokens per section	0.046	-0.060	0.36
Size & Flesch	0.000	-0.105	1.11
Flesch & number of words	0.000	-0.120	1.44
Flesch & word length	0.000	-0.604	36.53

the correlation of the various other measures with the structural size of the legislation. The motivation for this is that if a measure strongly correlates with the size of legislation, the measure may be beyond the legislator's control. A legislator can, of course, attempt to lessen the size of the legislation, but this might render the legislation less effective in practice, which harms instead of improves the quality of legislation. It may therefore be argued that measures that strongly correlate with the size of legislation are less useful as measures of the complexity of legislation. After all, a practical motivation for developing complexity measures is to support legislators in making legislation more accessible and understandable.

Katz & Bommarito found that size was at best weakly correlated with mean element depth. Our results show a stronger correlation with more statistical significance. Katz & Bommarito found that size strongly correlates with the number of sections. Our results are nearly identical with more statistical significance. Additionally, we observed that the measures text nodes, number of words, number of tokens, non-text nodes, below-section nodes, section nodes, mean depth, word entropy, lemmatised word entropy, citations total, mean leaf depth, above-section nodes and external citations either strongly or decently correlate with the size of legislation. Size and tokens per section very weakly correlate with the structural size of legislation. Net flow, word length, net flow per section and Flesch do not seem to correlate with the (structural) size of legislation. [5] also found that Flesch does not correlate with the number of words.

4. Conclusion

In this paper we have reported on an experiment to investigate whether the complexity framework of Katz & Bommarito [2] can be meaningfully used to

analyse the complexity of Dutch legislation. We found that this is possible both mathematically and computationally. We also compared our results to those of Katz & Bommarito. Since an absolute comparison in terms of the complexity numbers is not very informative because of differences between the Dutch and English language and legislation style, we mainly focused on correlation analysis. By and large, our correlation results were similar to the results in [2] but with higher statistical significance because of a higher number of legislative documents.

We also did several correlation analyses not done by Katz & Bommarito, particularly to see which complexity measures correlate with the size of legislation. This was motivated by the idea that complexity measures that strongly correlate with the structural size of legislation are less useful as measures of complexity since they are largely beyond the legislator's control. This means that the underlying idea of Katz & Bommarito to measure complexity in knowledge acquisition costs requires refinement, especially since the underlying aim of their work is to support legislators in making legislation less complex. In fact, our recommendation can be generalised to any measure that strongly correlates with features of legislation that are beyond the legislator's control. In light of this, we believe that our additional correlation analyses are a vital addition to the analysis of Katz & Bommarito, who did a correlation analysis for just two of their measures, namely, size versus sections & mean element depth. The results of our correlation analysis motivated us to recommend the replacement of the word entropy measure with the Flesch readability score in the normalised ranking composite since, unlike word entropy, the Flesch readability score only negligibly correlated with legislation size.

We end by mentioning two limitations that our approach shares with that of Katz & Bommarito. First, the framework gives only relative measures of complexity and no measures of when legislation is too complex. Second, the choice of composite measures is not yet governed by clear and convincing criteria. These issues should be addressed in future research.

References

[1] D. Bourcier and P. Mazzega. Toward measures of complexity in legal systems. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Law*, pages 211–215, New York, 2007. ACM Press.

[2] D.M. Katz and M.J. Bommarito. Measuring the complexity of the law: The United States Code. *Artificial Intelligence and Law*, 22:337–374, 2014.

[3] M. Palmirani and L. Cervone. Measuring the complexity of law over time. In P. Casanovas, U. Pagallo, M. Palmirani, and G. Sartor, editors, *AI Approaches to the Complexity of Legal Systems: AICOL 2013 International Workshops, Revised Selected Papers*, number 8929 in Springer Lecture Notes in AI, pages 82–99, Berlin, 2013. Springer Verlag.

[4] C. E. Shannon. Prediction and entropy of printed English. *The Bell System Technical Journal*, 30(1):50–64, 1951.

[5] B. Waltl and F. Matthes. Towards measures of complexity: Applying structural and linguistic metrics to German laws. In R. Hoekstra, editor, *Legal Knowledge and Information Systems. JURIX 2014: The Twenty-seventh Annual Conference*, pages 153 – 162. IOS Press, Amsterdam etc., 2014.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220476

Judgment Tagging and Recommendation Using Pre-Trained Language Models and Legal Taxonomy

Tien-Hsuan WU [a], Ben KAO [a], Henry CHAN [b], Michael MK CHEUNG [b]
[a] *Department of Computer Science, The University of Hong Kong*
[b] *Faculty of Law, The University of Hong Kong*

Abstract. We study the problem of machine comprehension of court judgments and generation of descriptive tags for judgments. Our approach makes use of a legal taxonomy $\mathcal{D}$, which serves as a dictionary of canonicalized legal concepts. Given a court judgment J, our method identifies the key contents of J and then applies Word2Vec and BERT-based models to select a short list T_J of terms/phrases from the taxonomy $\mathcal{D}$ as descriptive *tags* of J. The tag set T_J suggests concepts that are relevant to or associative with J and provides a simple mechanism for readers of J to compose *associative queries* for effective judgment recommendation. Our prototype system implemented on the Hong Kong Legal Information Institute (HKLII) platform shows that our method provides a highly effective tool that assists users in exploring a judgment corpus and in obtaining relevant judgment recommendation.

Keywords. judgment recommendation, judgment tagging, keyword extraction

1. Introduction

In common law, prior judgments (a.k.a. *precedents*) serve as the body of law following which courts make decisions with similar judicial reasoning. This principle (called *stare decisis*) makes it important that legal professionals be able to effectively find relevant judgments as references. There are a number of online platforms such as the World Legal Information Institutes (WorldLII) that provide online accesses to historical court judgments. These platforms generally provide tools for users to retrieve judgments with keyword search. In practice, however, the task of finding reference judgments for legal research is often *concept-based* instead of *instance specific*. For example, one may want to find previous cases of "*teenagers trafficking illegal drugs*" instead of specifically a case in which "*an 18-year-old was found carrying 5.2g of cocaine*" as the latter is so specific that a search will unlikely net any reference judgments. Traditional keyword-search tools do not handle concept-based search well as they require stringent string matching against the textual contents of judgments, which by nature is specific to each judgment. Another characteristic of judgment search is that the exercise is often *exploratory*, with the search goal not perfectly identified at the start but incrementally refined as one progresses. A lawyer who has found a judgment J that partially matches his/her search intent may want to expand his collection with other judgments that are associative with certain

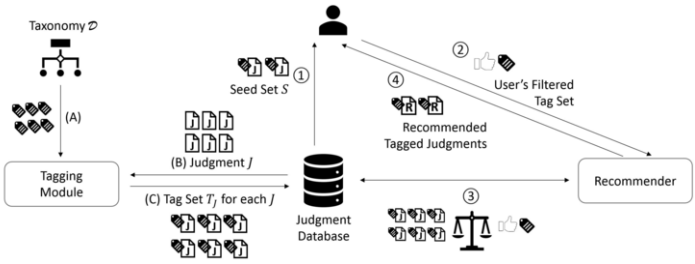


Figure 1. CAJUR architecture

details of J. For example, a lawyer researching for personal injury cases may have retrieved a judgment on *metatarsal bone* fracture because that is an injury that the lawyer's client has sustained. However, the client's injury may also involve *tibialis anterior* (a muscle attached to the metatarsal) tear. In this case, "tibialis anterior" is a concept that is *associative with* "metatarsal bone". Linkages from one judgment (e.g., on metatarsal bone) to others associative judgments (e.g., on tibialis anterior) would greatly facilitate the lawyer's reference judgment search.

In this paper we study the problem of **concept-based associative judgment recommendation**. Our goal is to design a system that assists legal professionals in effectively finding reference judgments during legal research. We assume that a user starts with a few judgments that satisfy his/her initial search goal. We call this initial set of judgments the *seed set* $\mathcal{S}$. A user can obtain $\mathcal{S}$ by any means, such as browsing or keyword searches. The problem is to recommend other judgments J's to the user with the requirement that the concepts covered in J's are either the same as or associative with those of the judgments in the seed set. Our proposed approach of concept-based associative judgment recommendation can help users in better expressing their search intents and exploring the judgment database resulting in more effective and efficient reference judgment search.

2. Method

In this section we describe our method CAJUR, which stands for *Concept-based Associative Judgment Recommendation*. CAJUR performs judgment tagging for effective recommendation. Figure 1 illustrates CAJUR's design. We start with a judgment database. CAJUR uses a legal taxonomy $\mathcal{D}$ as a dictionary of standard tag terms/phrases (A). It employs a tagging module that selects phrases from the taxonomy to create a tag set T_J for each judgment J in the database (B&C). Let $\mathcal{S}$ (①) be a seed set of judgments that are relevant to a user's search intent. Each judgment $J' \in \mathcal{S}$ is also given a tag set $T_{J'}$ by the tagging module. The user can compose a query (②) by including or excluding some tags in the tag sets $T_{J'}$'s. The recommender module (③) will then compare the query against the tag set T_J of each judgment J in the database to determine whether J should be recommended to the user (④). Next, we give technical details of the tagging module and the recommender module.

2.1. Tagging Module

Let $\mathcal{D} = \{p_1, p_2, ...p_{|\mathcal{D}|}\}$ be a legal taxonomy that consists of a collection of phrases p_i's. Given a judgment J, the tagging task is to determine a subset $T_J \subset \mathcal{D}$ such that each phrase

$p \in T_J$ is *relevant to* or *associative with* the content of J. Specifically, the tag set should consist of two parts $T_J = T_J^r \cup T_J^a$. The set T_J^r is a collection of *relevant tags* each of which describes a concept that is mentioned in the judgment. T_J^r helps the user grasp the major elements of J and to obtain recommendation of other judgments that are similar to J. On the other hand, the set T_J^a gives a collection of *associative tags*. These tags are not explicitly mentioned in the judgments but they describe concepts that are associative with those mentioned in J. The set T_J^a helps the user expand his/her search by considering related concepts that are not explicitly included in J. The tagging process consists of two steps, namely, (1) *sentence selection* and (2) *tag prediction*. To facilitate our discussion, we will illustrate our method using *personal injury compensation (PI) cases in Hong Kong* as an example.

2.1.1. Sentence Selection

Judgments are generally long and complex. To abstract a judgment, which may contain thousands of words, to a handful of concept tags requires techniques that can identify the key aspects and elements of a judgment. The first step of the tagging module is to extract a set of key sentences $KS_J = \{s_1, ..., s_{|KS_J|}\}$ from a judgment J that cover the major aspects of a court case. For example, for PI cases, key aspects include *plaintiff background*, *injury*, *loss*, *treatment* and *compensation*. We employ the technique given in [1] to identify key sentences for each given aspect in a judgment. Due to space limitation, readers are referred to [1] for details. In the following discussion, we focus on *injury* and *loss* aspects for illustrative purpose.

2.1.2. Tag Prediction

Given the taxonomy $\mathcal{D}$ and the key sentences KS_J, the next step is to determine the semantic correlation between each phrase $p \in \mathcal{D}$ and each sentence $s \in KS_J$. Those p's that are of high semantic correlation with the s's are collected in the tag set T_J. Recall that T_J consists of two subsets T_J^r and T_J^a. The tagging module thus uses two taggers, namely, a Word2Vec tagger and a BERT tagger to perform respective semantic analysis.

[Word2Vec tagger] The objective of the Word2Vec tagger is to generate the tag set T_J^r, which represents concepts that are mentioned in the judgment J. For each sentence $s \in KS_J$ that is extracted under a certain aspect A of the case, we consider the relevant section $\mathcal{D}_A$ of the taxonomy $\mathcal{D}$. For example, for the aspect *injury*, we consider a subset $\mathcal{D}_{\text{injury}} \subset \mathcal{D}$, which consists of 382 phrases that express various kinds of injuries; for the aspect *loss*, we consider another subset $\mathcal{D}_{\text{loss}}$, which consists of 41 phrases. Given a sentence s, we obtain all the noun phrases (using TEXTBLOB) contained in it. For each such noun phrase np, we compute the cosine similarity between the Word2Vec embeddings of np and each phrase $p \in \mathcal{D}_A$. If a phrase $p^* \in \mathcal{D}_A$ gives the highest score and if the score exceeds a threshold τ, we consider the phrase p^* and the sentence s semantically correlated. We thus put p^* in T_J^r. The threshold τ is determined empirically; In our prototype, we set $\tau = 0.6$. Note that the tags generated by the Word2Vec tagger express concepts that are directly relevant to the extracted sentences. Next, we describe a BERT tagger that finds tags that are associative with the sentence's contents. For example, the tag "loss of leisure" should be obtained if the judgment mentions "can't enjoy life".

[BERT tagger] We use the BERT Next Sentence Prediction (BERT-NSP) architecture to determine the semantic correlation between a phrase p and a sentence s, particularly

for phrases that express concept that are not explicitly mentioned in s. The BERT-NSP architecture has been used in sentence pair classification tasks. The model takes two text sequences as input and outputs a classification result. In our study, we train a BERT-NSP model that classifies if p and s are semantically correlated. Specifically, given a (p, s) pair, the classifier outputs a confidence score. If the score exceeds a threshold σ and if p is not already collected in T_J^r by the Word2Vec tagger, then p is considered an associative tag and is put in T_J^a. In our prototype, we set $\sigma = 0.8$. Furthermore, as there are potentially many associative concepts, we retain only the top-5 tags based on their confidence scores. To train the BERT tagger, we need a training set that consists of positive and negative samples. We collect the sentences and the phrases extracted by the Word2Vec tagger as the positive samples. For each sentence, we further sample 5 other phrases in $\mathcal{D}$ and include them as the negative samples. We trained six sub-models using different hyper-parameters, and our final BERT tagger outputs a phrase p for a sentence s if at least two of the sub-models vote for p.

2.2. Recommender Module

Judgment recommendation is made based on a user's search intent. For that, a user maintains a *focal tag set* F and an *exclusion tag set* E. Intuitively, we find judgments that mention the concepts expressed by the tags in F but not those in E. As a user explores the judgment database and reads a judgment J, the user can refine the F and E sets by including or excluding the tags given in T_J. With the F and E sets, the recommender module retrieves a set of candidate judgments R from the judgment database DB. R is given by: $R = \{X \in DB \,|\, F \subseteq T_X \text{ and } E \cap T_X = \varnothing\}$. Each judgment X in R is then ranked based on the following scores H_1 to H_3:
(Relevant tag count): $H_1 = |F \cap T_X^r|$. Recall that the relevant tags T_X^r (generated by the Word2Vec tagger) are those that express concepts directly mentioned in the judgment X. The higher the H_1 score, the more directly relevant is X to the user's search focus.
(BERT tagger votes): H_2 = total number of votes given by the 6 BERT models to the tags in F when the BERT models predict tags for judgment X. The higher this vote count, the more confident we are that X is relevant or is related to the concepts expressed in F.
(Average tag similarity): $H_3 = \text{AVG}(cs(t_a,t_b) \,|\, t_a \in F \text{ and } t_b \in T_X)$, where $cs(t_a,t_b)$ is the cosine similarity of the Word2Vec embedding vectors of tags t_a and t_b. H_3 measures the average semantic closeness of the focal tags and those of judgment X. Judgments in R are ranked first by their H_1 scores, with H_2 and H_3 serve as level 1 and level 2 tie-breakers.

3. Demonstration and Evaluation

We implemented a prototype of CAJUR on the HKLII platform. In this section we briefly describe the interface. Also, we evaluate the effectiveness of the tagging module and the recommendation module through experiments.

Figure 2 shows a screenshot of the interface where a PI judgment J is displayed in the middle. The tag set T_J, which consists of 9 tags, is shown in the right pane. Among them, the relevant tags T_J^r are displayed in darker blue while the associative tags T_J^a are in lighter blue. The taxonomy $\mathcal{D}$ from which tags are obtained provides a hierarchy of phrases with related concepts grouped under a subtree. Tags in T_J that share common

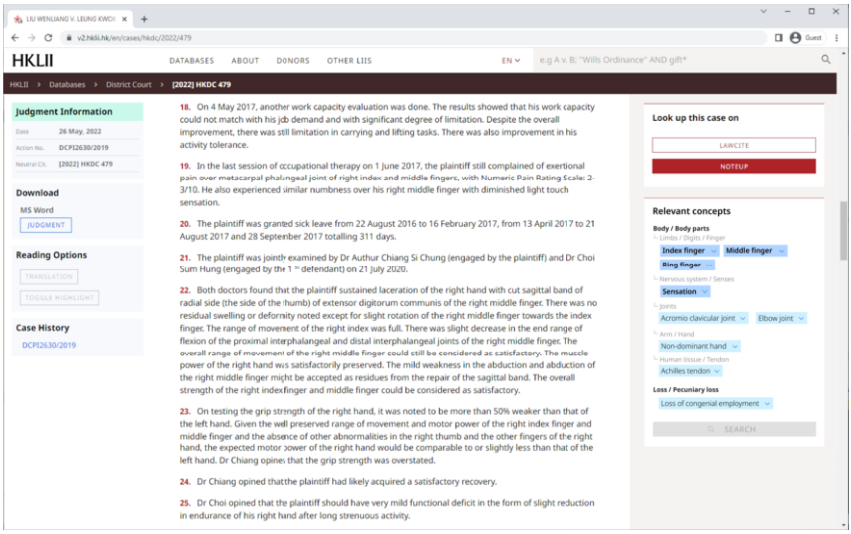


Figure 2. Interface showing a judgment and its tags

Table 1. Tag evaluation

	PS	PSLegal	CAJUR
Relevant	76 (30.8%)	75 (30.4%)	70 (28.4%)
Associative	4 (1.6%)	5 (2.0%)	49 (19.8%)
Others	167 (67.6%)	167 (67.6%)	128 (51.8%)

Table 2. Recommendation quality

	Avg. Score
Document Similarity	0.48
Tag-Doc Similarity	0.26
CAJUR	1.16

path prefix in the hierarchy are displayed as a group with the path prefix shown to provide contexts. For example, the tags *Index finger*, *Middle finger*, and *Ring finger* are child nodes of the branch *Limbs* → *Digits* → *Finger*. A user can put a tag into his focal tag set F or his exclusion tag set E by either *including* or *excluding* a tag, respectively. The user can then click the search bar and the system in response will display a short list of recommended judgments.

We conduct experiments to evaluate the effectiveness of CAJUR's tagging module and recommendation module. For tag quality, we compare CAJUR against two methods, namely, PS and PSLegal [2], which have been shown to be effective methods in identifying key phrases in legal documents. We sample 25 PI judgments from Hong Kong courts. For each judgment J, CAJUR generates a tag set T_J. We then apply PS and PSLegal to extract $|T_J|$ key phrases from J for each method. All phrases given by CAJUR, PS, and PSLegal are collected and shuffled before they are presented to human evaluators in a single list. We employ six human evaluators who are asked to classify each tag (phrase) with the following descriptors: **Relevant tag**: The tag helps the user grasp an important element/concept mentioned in the judgment; It is conceivable that the user would select the tag in expressing a search intent for similar judgments. **Associative tag**: The tag does not express elements/concepts that are directly mentioned in the judgment but is related to or associative with them; It is conceivable that the user would select the tag in extending a search intent in search of other related judgments. **Others**: It is unlikely that the user will select the tag in expressing a search intent.

Table 1 shows the total number of each tag category given by each method as evaluated by the human evaluators. We see that CAJUR generates significantly more associa-

tive tags than PS and PSLegal. CAJUR is therefore much more effective in suggesting tags to users in formulating *associative queries*. This will greatly help users in exploring the judgment database. As an illustrative example, there are a number of PI judgments on HKLII that contain the phrases "upper back" and "neck". For these judgments, CAJUR generates the tag "trapezius muscle", which is a large muscle piece that connects the neck and the upper back. This associative tag helps connect the back-and-neck judgments to those that mention trapezius muscle injuries. The ability of CAJUR in generating associative tags thus help users identify related judgments which would otherwise be missed if only straightforward keyword matching were done.

Finally, we evaluate the accuracy of CAJUR's recommendation module. For each of the 25 sampled judgments used in the tagging experiment, we select 1-2 tags into a focal tag set F. We then recommend two judgments using CAJUR and the following two baseline methods. **Document Similarity**: We embed the sampled judgment and all judgments in the corpus using Doc2Vec [3]. The two judgments that are most similar to the selected sampled judgment are selected. **Tag-Doc Similarity**: We embed the focal tags and all judgments in the corpus using Doc2Vec, and select two judgments that are most similar to the focal tags. We collect the recommended judgments from all approaches and shuffle them before presenting them to a human evaluator for evaluation. The evaluator was asked to decide whether each recommended judgment is *relevant to the focus* (2 marks), *somewhat relevant* (1 mark), or *irrelevant* (0 marks).

Table 2 shows the results. We see that CAJUR's recommendation module is significantly more effective compared with the baselines. We observe that Tag-Doc Similarity does not perform as well as Document Similarity. The reason is that the Doc2Vec embedding vectors of some words (i.e., tags) can be very different from those of entire documents (i.e., judgments) as the word distributions of the two are quite different. This shows that recommendation approach with tagging needs to be thoughtfully designed. CAJUR generates more precise recommendations by fully using the tags assigned to the judgments in the corpus as the recommendation criteria.

4. Conclusion

In this paper we study the problem of tagging and recommending legal judgments. We propose CAJUR, which achieves the tasks by using a taxonomy and the BERT model. CAJUR generates relevant tags that express concepts mentioned in judgments as well as associative tags that provide users hints on how their search could be extended with associative queries. We present a user interface that displays tags and facilitates a user to specify a search focus. The evaluation results show that CAJUR can generate high quality relevant and associative tags, as well as recommend judgments that are much more relevant to the user search focus.

References

[1] Wu TH, Kao B, Chan F, Cheung A, Cheung M, Yuan G, et al. Semantic Search and Summarization of Judgments Using Topic Modeling. In: Legal Knowledge and Information Systems. IOS Press; 2021. .

[2] Mandal A, Ghosh K, Pal A, Ghosh S. Automatic catchphrase identification from legal court case documents. In: CIKM; 2017. p. 2187-90.

[3] Le Q, Mikolov T. Distributed representations of sentences and documents. In: International conference on machine learning. PMLR; 2014. p. 1188-96.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220477

Multi-Granularity Argument Mining in Legal Texts

Huihui Xu [a,b,1], Kevin Ashley [a,b,c]
[a] *Intelligent Systems Program, University of Pittsburgh*
[b] *Learning Research and Development Center, University of Pittsburgh*
[c] *School of Law, University of Pittsburgh*

Abstract. In this paper, we explore legal argument mining using multiple levels of granularity. Argument mining has usually been conceptualized as a sentence classification problem. In this work, we conceptualize argument mining as a token-level (i.e., word-level) classification problem. We use a Longformer model to classify the tokens. Results show that token-level text classification identifies certain legal argument elements more accurately than sentence-level text classification. Token-level classification also provides greater flexibility to analyze legal texts and to gain more insight into what the model focuses on when processing a large amount of input data.

Keywords. Argument mining, Information retrieval, Natural language processing, Deep learning

1. Introduction

Argument mining is "the automatic discovery of an argumentative text portion, and the identification of the relevant components of the argument presented there." [1]. The goal is to identify and extract the structure of inference and reasoning expressed as arguments presented in natural language [2]. Legal argument mining identifies and extracts arguments in legal texts.

In previous work, we applied and demonstrated that supervised machine learning (ML) and deep learning methods can classify sentences of legal cases in terms of the roles they play in a legal argument to some extent.

In this paper, we take legal argument mining to a finer-grained level – token-level argument mining where the tokens are words. That is, we treat it as a word classification task. Token-level argument mining has several potential advantages. First, it is more robust against errors in sentence segmentation [3]. Secondly, it can efficiently handle single sentences that exhibit multiple argumentative elements. For example, as shown in Figure 1, different parts of a single sentence have been labeled as conclusion and reason. If we apply sentence-level classification methods for each label to the same sentence, we confuse the classifier and lose ordering information as compared with training on those sub-sentences. Finally, token-level argument mining can provide insights about the contributions of particular words to sentence-level classification.

[1] Corresponding Author: huihui.xu@pitt.edu

Allowing the appeal, that s. 23(1) of the Social Assistance Act places a mandatory responsibility upon social service committees to provide assistance for all persons in need as defined in s. 19(e) of the Act.

Figure 1. An example of a legal summary sentence whose parts are labeled with two argumentative elements. Green-colored text represents conclusion, and blue-colored text represents reasons.

Our contributions in this work are, first, to apply token-level argument mining to legal texts. Secondly, we show that this token-level approach improves the accuracy of classifying sentences in terms of legal argument elements. Finally, our error analysis shows new ways to understand the significance of certain tokens/words in classifying sentences by legal argumentative roles.

2. Related Work

Argument mining in the legal domain includes training classifiers on different types of extracted features to classify premises and conclusions [4,5], investigating discursive and argumentative characteristics of legal documents [6], identifying argument schemes [7] or rhetorical roles that sentences play in legal cases [8], summarizing legal cases in terms of argument elements [9,10], and accounting for sentence position and embedding in legal argument classification [11].

Some recent argument mining research has focused on a more granular level, the token-level, which means assigning labels to every word. For example, [3] showed that the token-level argument mining employed in Argument Unit Recognition and Classification (AURC) retrieves a larger number of arguments than sentence-level mining. [12] also treated the Kaggle competition "Feedback Prize - Evaluating Student Writing" as a token-level argument mining task.

Some sequential labeling techniques have been applied in this context, like BERT, Conditional Random Fields, and Bi-LSTM [3,13]. This conceptualized token-level classification resembles the classic sequence labeling tasks in NLP like Named Entity Recognition (NER). Hidden Markov Models (HMM), Maximum entropy Markov models (MEMMs) [14], and Conditional Random Fields (CRF) are the most commonly used sequential labeling techniques in the pre-neural model era. Recently, researchers have applied neural models to tackle sequence labeling problems such as convolution networks [15], bidirectional LSTM-CRF models [16], and BERT-CRF [17].

As far as we know, token-level argument mining has not yet been applied in legal argument mining. We have applied it to a corpus of expert-annotated legal cases and summaries as described below.

3. Dataset

Our dataset comprises 28,733 legal cases and summaries prepared by attorneys, members of legal societies, or law students and provided by the Canadian Legal Institute

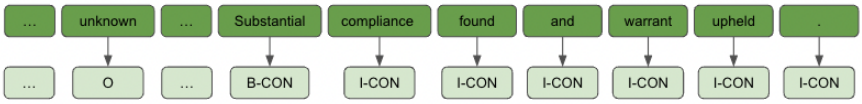


Figure 2. An example of using BIO format to tag every token in a conclusion sentence.

(CanLII).[2] As noted our IRC type system for labeling sentences in legal cases and case summaries includes: **Issue** – Legal question which a court addressed in the case; **Conclusion** – Court's decision for the corresponding issue; **Reason** – Sentences that elaborate on why the court reached the Conclusion. All un-annotated sentences are treated as non-IRC sentences.

We employed two third-year law school students to annotate sentences from the human-prepared summaries in terms of issues, reasons, and conclusions. They annotated 1049 randomly selected case/summary pairs. Cohen's κ [18] metric is used to assess the degree of agreement between two annotators. The mean of Cohen's κ coefficients across all types for summaries is 0.734; the mean for full texts of cases is 0.602. Both scores indicate substantial agreement between two annotators according to [19]. The full texts annotation agreement is lower than that of summaries since the sentences of full texts and summaries are not in a one-to-one mapping.

The BIO or IBO tagging scheme was first proposed in [20]. We adapt the BIO tagging format to our annotated summary/full text pairs. One advantage of this tagging format is it allows tokens to carry both the sentence structure and sentence type information. As shown in the Figure 2, the B-prefix of a tag indicates the beginning of an annotated conclusion sentence, the I-prefix of a tag indicates the token is inside a conclusion sentence, while the O tag indicates the token does not belong to any typed sentence.

4. Experiment

We pre-processed our dataset using the BIO format: we first tokenized all the sentences in summaries and full texts, then assigned the corresponding BIO tags to every token. Those BIO-tagged tokens were then put into the pretrained Longformer [21] model for token classification. We chose Longformer over the traditional BERT [22] model because of its ability to process longer documents. The maximum input length is 1024 tokens due to the GPU limitation.[3] We decided to segment the full text documents into multiple chunks of length 1024 to avoid information loss. We experimented with two types of Longformer: Lonformer-base-4096 and Longformer-large-4096.[4] We split our datasets of summaries and full texts into 80% training, 10% validation and 10% test sets.

[2] https://www.canlii.org/en/
[3] We use a single NVIDIA Titan X GPU with 12 GB memory.
[4] https://github.com/allenai/longformer

Table 1. Results of BIO token-level classification on summaries and full texts. All the results are reported in terms of precision, recall and F_1 scores. The scores inside parentheses are produced by Longformer-base-4096, while the scores outside of parentheses are produced by Longformer-large-4096.

	Summary						
	B-Issue	I-Issue	B-Reason	I-Reason	B-Conclusion	I-Conclusion	O
Precision	0.83 (0.79)	0.83 (0.80)	0.72 (0.67)	0.75 (0.70)	0.83 (0.77)	0.80 (0.73)	0.78 (0.77)
Recall	0.79 (0.78)	0.78 (0.81)	0.80 (0.75)	0.80 (0.76)	0.84 (0.80)	0.82 (0.72)	0.75 (0.72)
F1-score	0.81 (0.78)	0.81 (0.81)	0.75 (0.71)	0.77 (0.73)	0.83 (0.78)	0.81 (0.72)	0.77 (0.74)
	Full-texts						
	B-Issue	I-Issue	B-Reason	I-Reason	B-Conclusion	I-Conclusion	O
Precision	0.66 (0.62)	0.80 (0.75)	0.54 (0.44)	0.69 (0.64)	0.53 (0.46)	0.65 (0.61)	0.98 (0.98)
Recall	0.55 (0.52)	0.70 (0.69)	0.36 (0.36)	0.63 (0.62)	0.43 (0.44)	0.63 (0.61)	0.98 (0.98)
F1-score	0.60 (0.56)	0.74 (0.72)	0.43 (0.40)	0.66 (0.63)	0.47 (0.45)	0.64 (0.61)	0.98 (0.98)

Table 2. Results of classification on summaries and full texts. All the results are reported as F_1 scores.

	Summary				Full text			
	Issue	Reason	Conclusion	Non-IRC	Issue	Reason	Conclusion	Non-IRC
Longformer(large)-BIO	0.81	0.77	0.87	0.79	0.66	0.68	0.67	0.98
Longformer(base)-BIO	0.82	0.72	0.81	0.77	0.63	0.67	0.64	0.98
Longformer(base)-no BIO	0.75	0.73	0.80	0.75	0.49	0.30	0.49	0.95
Longformer(large)-no BIO	–	–	–	0.58	–	–	–	–
Legal-BERT	0.76	0.73	0.81	0.76	0.52	0.47	0.56	0.98
BERT	0.73	0.70	0.79	0.69	0.50	0.49	0.52	0.98

5. Results

Table 1 shows the results of token-level classification in summaries and full texts. As seen in the table, the classification results on the summaries are better than on the full texts in terms of precision, recall, and F1 score. The better results on summaries are expected because the summaries are shorter than full texts and more clearly organize the sentences. To determine the sentence type from the resulting token labels, we used the token type that appears most frequently in the sentence. Table 2 reports the results of assigning sentence type utilizing the token labels.

For purposes of comparison, we trained three techniques on sentence-level annotation: Legal-BERT [23], BERT [22] and Longformer. None of these baseline techniques employ token-level annotation. In order to compare the results across different models, we tested them on the same test set. As shown in the Table 2, Longformer(large)-BIO achieved better F_1 scores in sentence labeling across all sentence types (e.g., issues, reasons, and conclusions).

6. Discussion

We trained Longformer on annotated summary and full text sentences, respectively. It confirms that the BIO approach classifies the sentences more effectively. For token-level classification, as shown in Table 1, we can see that the F_1 scores of I-prefixed token types (i.e., inside) are higher than B-prefixed token types (i.e., beginning). Our intuition is that

I-prefixed token types benefit from more training data, because each annotated sentence has only one beginning token while I-prefixed tokens dominate the rest of the annotated sentence.

After investigating the results of token-level classification, we find that the model is more likely to assign I-Reason to a Non-IRC (O) type in both summaries and full texts. A large portion of those misclassified tokens are stop words, like 'the', 'to', 'of' etc., which are commonly used within issue sentences. Those stop words, of course, appear everywhere in a document; their type depends more on their context than their semantic meaning. The token-based classification indicates some contexts where even a stop word like 'the' appears to have an effect.

We also found that 'HELD' appears most frequently in the correctly classified B-Conclusion tokens in the summaries; 'the' appears most frequently in the correctly classified I-Issue tokens in the full texts. The human summarizers tend to make conclusions more noticeable to readers by using indicators such as 'HELD'. Those indicators are captured by the model.

For sentence-level classification, the sentence type is determined by the token type that appears most often in the sentence. We observed that conclusions in summaries are prone to be misclassified as reasons. We investigated those misclassified conclusion sentences and find most sentences were completely misclassifed on a token-level. That is, the model identified no conclusion tokens. Only one sentence had several correctly identified conclusion tokens including 'support' and 'allowed'. We have been unable to explain the token-level misclassification.

7. Conclusion

In this work, we experimented with multi-granular argument mining from legal texts. We employed two label classification tasks: token-level (i.e., word-level) classification and sentence-level classification. The sentence-level classification is based on the results of the token-level classification. Results showed that token-level classification achieved more accurate sentence classification than state-of-the-art sentence-classification models. The token-level classification not only improved the sentence classification performance but also gave insights into how the model behaves with respect to certain tokens.

In future work, we plan to use the token-based approach to more accurately classify issues, conclusions, and reasons and to use these IRC argument elements to improve automatic case summarization. We will explore using these finer-grained indicators to identify other legal argumentative units, such as factors, and to better evaluate the quality of legal summaries in terms of coverage of argument elements.

Acknowledgement

This work has been supported by grants from the Autonomy through Cyberjustice Technologies Research Partnership at the University of Montreal Cyberjustice Laboratory and the National Science Foundation, grant no. 2040490, FAI: Using AI to Increase Fairness by Improving Access to Justice. The Canadian Legal Information Institute provided the corpus of paired legal cases and summaries. This work was supported in part by the

University of Pittsburgh Center for Research Computing through the resources provided. Specifically, this work used the H2P cluster, which is supported by NSF award number OAC-2117681.

References

[1] Peldszus A, Stede M. From argument diagrams to argumentation mining in texts: A survey. International Journal of Cognitive Informatics and Natural Intelligence (IJCINI). 2013;7(1):1-31.
[2] Lawrence J, Reed C. Argument mining: A survey. Computational Linguistics. 2020;45(4):765-818.
[3] Trautmann D, Daxenberger J, Stab C, Schütze H, Gurevych I. Fine-grained argument unit recognition and classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. p. 9048-56.
[4] Moens MF, Boiy E, Palau RM, Reed C. Automatic detection of arguments in legal texts. In: Proceedings of the 11th international conference on Artificial intelligence and law; 2007. p. 225-30.
[5] Mochales-Palau R, Moens M. Study on sentence relations in the automatic detection of argumentation in legal cases. Frontiers in Artificial Intelligence and Applications. 2007;165:89.
[6] Mochales R, Moens MF. Study on the structure of argumentation in case law. In: Proceedings of the 2008 conference on legal knowledge and information systems; 2008. p. 11-20.
[7] Feng VW, Hirst G. Classifying arguments by scheme. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies; 2011. p. 987-96.
[8] Saravanan M, Ravindran B. Identification of rhetorical roles for segmentation and summarization of a legal judgment. Artificial Intelligence and Law. 2010;18(1):45-76.
[9] Xu H, Savelka J, Ashley KD. Toward summarizing case decisions via extracting argument issues, reasons, and conclusions. In: Proceedings of the eighteenth international conference on artificial intelligence and law; 2021. p. 250-4.
[10] Elaraby M, Litman D. ArgLegalSumm: Improving Abstractive Summarization of Legal Documents with Argument Mining. arXiv preprint arXiv:220901650. 2022.
[11] Xu H, Savelka J, Ashley KD. Accounting for sentence position and legal domain sentence embedding in learning to classify case sentences. In: Legal Knowledge and Information Systems. IOS Press; 2021. p. 33-42.
[12] Ding Y, Bexte M, Horbach A. Don't Drop the Topic-The Role of the Prompt in Argument Identification in Student Writing. In: Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022); 2022. p. 124-33.
[13] Ajjour Y, Chen WF, Kiesel J, Wachsmuth H, Stein B. Unit segmentation of argumentative texts. In: Proceedings of the 4th Workshop on Argument Mining; 2017. p. 118-28.
[14] Freitag D, McCallum A. Information extraction with HMM structures learned by stochastic optimization. AAAI/IAAI. 2000;2000:584-9.
[15] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. Journal of machine learning research. 2011;12(ARTICLE):2493-537.
[16] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:150801991. 2015.
[17] Souza F, Nogueira R, Lotufo R. Portuguese named entity recognition using BERT-CRF. arXiv preprint arXiv:190910649. 2019.
[18] Cohen J. A coefficient of agreement for nominal scales. Educational and psychological measurement. 1960;20(1):37-46.
[19] Landis JR, Koch GG. The measurement of observer agreement for categorical data. biometrics. 1977:159-74.
[20] Ramshaw LA, Marcus MP. Text chunking using transformation-based learning. In: Natural language processing using very large corpora. Springer; 1999. p. 157-76.
[21] Beltagy I, Peters ME, Cohan A. Longformer: The Long-Document Transformer. arXiv:200405150. 2020.
[22] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.
[23] Zheng L, Guha N, Anderson BR, Henderson P, Ho DE. When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset. arXiv preprint arXiv:210408671. 2021.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220478

On Capturing Legal Knowledge in Ontology and Process Models Combined

The Case of an Appeal Process

Melissa ZORZANELLI COSTA [a,1], Giancarlo GUIZZARDI [b] and João Paulo A. ALMEIDA [b]
[a] *Federal University of Espírito Santo, Brazil*
[b] *University of Twente, The Netherlands*

Abstract. In this paper, we explore conceptual modeling as a means to improve the explicit representation of key aspects of a legal procedure. We employ in tandem an ontology-based structural conceptual model and a behavioral process model as complementary views on a legal subject matter. We examine as a case a specific type of appeal in the Brazilian legal system and establish a correspondence between elements in the models and fragments of the specific norms on which they are grounded. These correspondences are expressed with identifiers using the Brazilian LexML identification scheme.

Keywords. Legal process, Legal ontology, OntoUML, BPMN, LexML

1. Introduction

Given the importance of legal systems for many aspects of our lives, their functioning demands a much higher level of *transparency* than that which is required of other organizational environments (such as private enterprises). While legal institutions are created mostly through documented speech acts, there are a number of barriers for transparency. These include opaque aspects of legal jargon used in legal documents (including legislation), a number of procedural or operational aspects embodied in the practice that are not captured explicitly in legal documents, the complex nature of the legal system, and the inescapable ambiguity of natural language [4]. Lack of transparency in legal systems affects the access of citizens to justice as well as the design and operations of digital legal information systems, which are key to tame the scale of current societal demands.

In this paper, we report on ongoing work to explore conceptual modeling as a means to improve the explicit representation of key aspects of a legal procedure. The approach is based on the simultaneous development of structural ontology-based conceptual models and behavioral process models. In order to capture the structural aspects of this domain, we adopt OntoUML, an ontologically well-founded UML profile whose primitives reflect ontological distinctions of an underlying foundational ontology (UFO [7]); and to capture the behavioral (or dynamic) aspects of the domain, we adopt the Business Process Model

[1]Corresponding Author: Melissa Zorzanelli Costa, Federal University of Espírito Santo, Brazil.

and Notation (BPMN). Our OntoUML and BPMN diagrams share events and other UFO prescribed ontological entities (e.g., phases, qualities and modes) providing a better integration for the models [10,15]. Finally, we propose a procedure for grounding these models by aligning the classes and assertions used therein with corresponding fragments of a suitable normative description. In this case, such alignment is supported by the LexML identification scheme[2] which is the basis of a Brazilian open data system to identify and publicize legal documents [13]. We examine the representation of an specific type of appeal in the Brazilian legal system (a *Request for Standardizing the Interpretation of a Federal Law*) that is part of a highly specialized procedure in Federal Courts, and often considered verbose and nontransparent.

Several approaches in the literature address the combination of structural and dynamic viewpoints [14,16]. In some of these approaches, there is an explicit recognition that linking laws and models can increase the traceability between laws and processes, helping the law makers to elaborate models in collaboration with software developers and process engineers, and understand the impact of law or process changes to their counterparts. A common feature of [14,16] (and also other ontology-based approaches such as [3]) is the representation of the structural aspects of the domain using OWL-DL. OWL-DL (despite the name) is a logics offering no support for real ontological analysis, as well as for explicit representation of the result of these analyses. In contrast, OntoUML supports the modeler's analysis with rich theories of types, relations, events, as well as existentially dependent and independent endurants. In addition, it offers a system of modeling primitives and constraints that can explicit represent these notions. In particular, it allows for representation of contingent (i.e., dynamic) intrinsic and relational properties and the contingent types whose dynamic classification conditions are defined by them (roles, phases, rolemixins). This allows for direct connection points between elements in the structural viewpoint, and events, conditions and decision points in the dynamic/process model. Further, the semantics of events in OntoUML models and the constraints governing their connection to enduring entities improves the ontological consistency of the overall approach (e.g., w.r.t. to semantics of object identifiers, as well as interpretation of the modal properties of the respective types [9]). OWL, in contrast, being a monotonic language and having no built in notion of temporal modality is particularly limited in modeling dynamic aspects of the domain. Finally, OntoUML modelers can count on a rich ecosystem of tools for model construction, verification, OWL generation, validation and verbalization [5,11].

2. Towards Models for an Appeal Process

With the introduction of the new civil procedure code that came into force in 2015, Brazil is rapidly evolving into a system of *stare decisis*, adopting a normative model of formally binding precedents and, thus, changing from a predominantly *civil law* system to become a country with a hybrid system of *civil and common law* [17]. In this new context, cases are then to be decided according to consistent and principled rules so that similar facts will yield similar results. In practical terms, precedents set by appellate courts—the Federal Supreme Court (STF) and the Superior Court of Justice (STJ)—should be applied as

[2] https://www.lexml.gov.br/

binding precedents in future decisions. In the scope of the Federal Special Courts in Brazil, parties may challenge unfavorable Appellate Panel decisions by pointing to deviations from established precedents. In this case, they file a so-called *Request for Standardizing the Interpretation of a Federal Law* (henceforth RS). This request is analyzed by a federal judge for admissibility before it is sent for consideration of the *National Uniformization Panel (TNU)* (according to law n° 10.259/01, article 14). The focus of our models is on this type of request, its phases and the admissibility procedure.

In OntoUML, enduring entities (continuants, endurants) are classified into a system of *kinds* that are mutually disjoint and exhaust the entities in that particular sub-portion of the domain. In other words, all enduring entities considered belong to exactly one kind. Entities of different kinds can contingently play different *roles* in the scope of relational contexts, and contingently instantiate different *phases*. Phases are then types that entities instantiate contingently due to changes in their intrinsic properties. The so-called dispersive types (non-sortals) classify entities of multiple kinds. These types include *categories* and *role mixins* (roles played by entities of multiple kinds). These different types of types can classify both *objects* (independent enduring entities) as well as *qualities* (reified intrinsic characteristics), and *relators* (relational contexts formed by reified relational characteristics) [7,8]. Finally, enduring entities change phases and start playing roles by changing their intrinsic and relational properties, respectively. This is caused by occurrence of *events* [2]. These distinctions put forth by UFO are explicitly encoded in the syntax of OntoUML (with UML stereotypes). Figure 1 shows a fragment of the OntoUML produced, focusing on the REQUEST FOR STANDARDIZING (RS) and its phases. It identifies the various roles involved in the JUDICIAL PROCESS, the APPELLATE DECISION against which the RS is filed and the ADMISSIBILITY DECISION in which an RS is analyzed and ruled on.

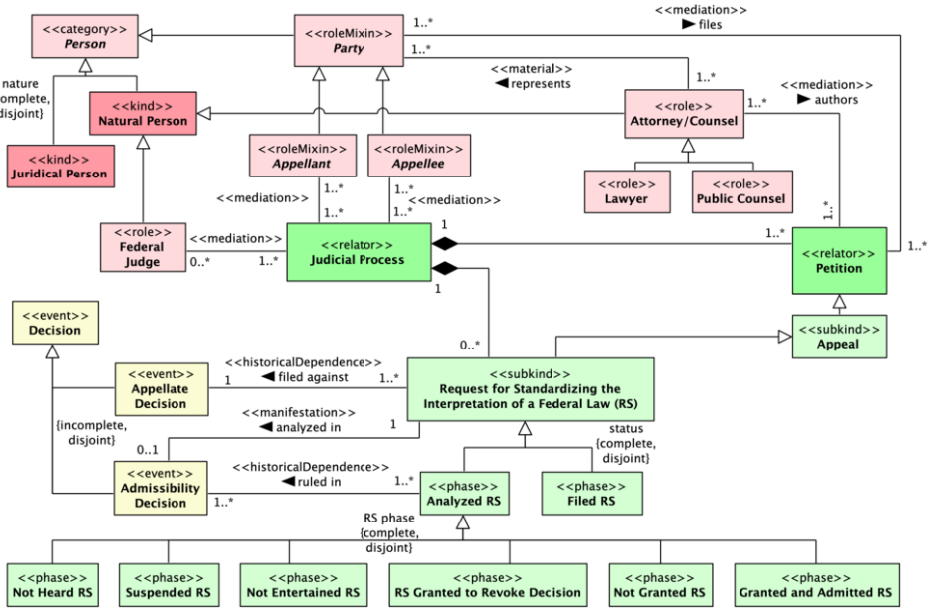


Figure 1. Request for Standardizing Legal Ontology

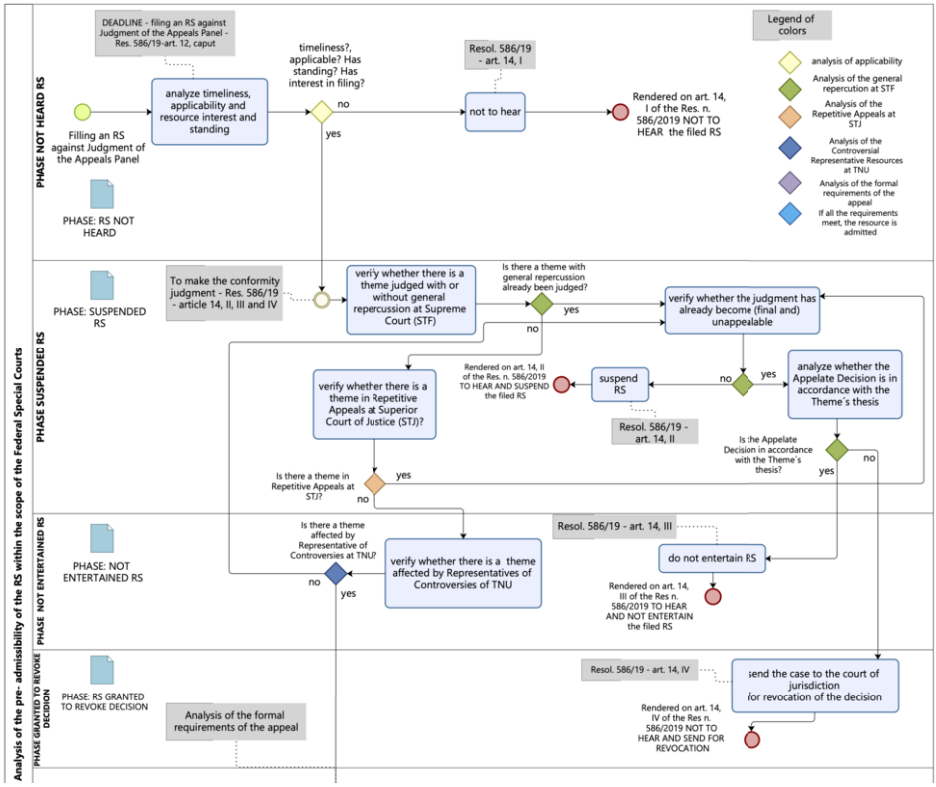


Figure 2. Analysis of the admissibility of the RS in a Federal Special Court, based on Res.n°586/19

The analysis of an RS's admissibility is made in accordance to the aforementioned internal regulations of the TNU (Res n°586/19). The RS goes through different phases according to our process model, depending on the result of the analysis procedure (represented by the BPMN diagram in Figure 2). The same phases found in the ontology are represented here in the process model. We show only a fragment of the complete diagram[3]. The Superior Courts establishes/recognizes topics ('themes') of general repercussion (at the level of the Federal Supreme Court – STF), repetitive appeals when there are multiple appeals based on the same question of law (at the Superior Court of Justice – STJ), and RSs (at the TNU) as representatives of controversies. In order to bring greater legal certainty and uniformity across cases, the judgment of these themes must consider and be considered as binding precedents. If the APPELLATE DECISION is in accordance with a thesis of a 'theme' (a biding precedent) set by the STF, STJ or TNU, the appeal is NOT ENTERTAINED. But, in contrast, if diverges from the understanding of the supreme courts under the rules of general repercussion or of the resolution of multiple appeals on the same point of law, the case must be further analyzed by the proper court of jurisdiction for the possible revocation of the decision to adapt to the binding precedent, standardizing the understandings of the Brazilian courts on the same subject. If the RS meets the legal and regimental requirements, it must be GRANTED AND ADMITTED and forwarded to the

[3]See the complete document at https://github.com/MelissaZor/JURIX2022

National Uniformization Panel (TNU) and, if there are multiple appeals based on the same issue of law, it should be classified as representative of the controversy with the other SUSPENDED RSs, until the trial and final judgment of the pilot case occurs. This aims at standardizing decisions on the same subject in Federal Special Courts throughout the country: *"the task of determining the* ratio decidendi*, or rule of decision of a case, which raises similar issues of selection, characterization, and abstraction of case facts."* [1].

2.1. Grounding Domain Classes in Norms

In order to ground the elements related to the admissibility of an RS that appear in the OntoUML and BPMN diagrams, we align these elements to the norms in force that constitute them. A small fragment of the result of this alignment is shown in Table 1. The table refers to the grounding norms including their LexML norm fragment identifiers, which can be resolved with hyperlinks to `https://normas.leg.br/`, revealing, when available in that database, the corresponding legal text. For example, we observe the definition of the APPELLATE DECISION event class in article 204 of the Civil Procedure Code, as well as the RS with a definition based on article 14 of federal law n° 10.259/01.

Model Element	Law/Norm	URN
JUDICIAL PROCESS	13.105/15-Art. 2°	urn:lex:br:federal:lei:2015-03-16;13105!art2
RS	10.259/01-Art. 14	urn:lex:br:federal:lei:2001-07-12;10259!art14
PUBLIC COUNSEL	Fed. Const.-Art. 131	urn:lex:br:fed:const:1988-10-05;1988!art131
APPELLATE DECISION	13.105/15-Art. 204	urn:lex:br:federal:lei:2015-03-16;13105!art204
JURIDICAL PERSON	10.406/02-Art. 41	urn:lex:br:federal:lei:2002-01-10;10406!art416
SUSPENDED RS	Res.n°586/19-Art. 14, III	urn:lex:br:cjf:res:2019-09-30;586!art14_inc3

Table 1. Validation of conceptual elements based on norms with LexML hyperlinks

3. Final Considerations

The transparent and expressive representation of legal systems has the potential to bring multiple benefits for these systems. In particular, representing the law in a well-founded diagrammatic manner can bring benefits in terms of the interpretability and explainability of legal texts [6]. This is particularly important for complex procedures such as those addressed in this paper. In complex modeling cases, viewpoint modeling has been used in many different areas as a mechanism for complexity management via separation of concerns [12]. In this paper, we explore some preliminary results of an approach for the multi-viewpoint conceptual modeling of the law that combines: a structural perspective modeled with an ontology-driven conceptual modeling language (OntoUML); (ii) with a dynamic perspective modeled in the BPMN notation. Specific ontological categories present in the OntoUML structural model provide for explicit alignment points through which the two models can be combined. This approach is applied in the modeling of *Requests for Standardizing the Interpretation of a Federal Law* in the Brazilian system.

In future work, we intend to investigate the cognitive benefits of the combined models in supporting the stakeholders of the modeled procedures. We also intend to elaborate on guidelines and semantically-motivated syntactic rules to establish the correspondence

between the structural and the behavioral views. The OntoUML tools already provide for a transformation into OWL, which can then be further employed for reasoning about the structural aspects. An investigation into suitable semantics for the BPMN model in terms of OWL and UFO-B may further support us in reasoning on the combined models. We will also examine the benefits of the availability of the conceptual models for automation of certain decisions in the analysis of admissibility of an RS. This will require us to formalize the content of the various decisions along the process model presented in this paper. We intend to extract and semantically-annotate data from judgments directly from the TNU portal.

Acknowledgements

This research is partly funded by Brazilian funding agencies CNPq (313687/2020-0) and FAPES (281/2021).

References

[1] Ashley, K.D.: Precedent and legal analogy. Handbook of Legal Reasoning and Argumentation pp. 673–710 (2018)
[2] Benevides, A.B., Bourguet, J.R., Guizzardi, G., Peñaloza, R., Almeida, J.P.A.: Representing a reference foundational ontology of events in SROIQ. Appl. Ontology **14**(3), 293–334 (2019)
[3] Bourguet, J.R., Zorzanelli Costa, M.: About the exposition of brazilian jurisprudences. In: ONTOBRAS. vol. 1862, pp. 138–143 (2016)
[4] Bourguet, J.R., Zorzanelli Costa, M.: Scoring judicial syllabi in Portuguese. In: JURIX. vol. 302, pp. 119–124 (2017)
[5] Fonseca, C.M., Sales, T.P., Viola, V., Fonseca, L.B.R., Guizzardi, G., Almeida, J.P.A.: Ontology-driven conceptual modeling as a service. In: Proc. FOMI 2021. CEUR Workshop Proceedings (2021)
[6] Griffo, C., Almeida, J.P.A., Guizzardi, G.: Conceptual Modeling of Legal Relations. In: Conceptual Modeling - 37th International Conference, ER 2018. pp. 169–183. Springer (2018).
[7] Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. CTIT PhD thesis series, Centre for Telematics and Information Technology, Telematica Instituut (2005)
[8] Guizzardi, G., Botti Benevides, A., Fonseca, C.M., Porello, D., Almeida, J.P.A., Prince Sales, T.: UFO: Unified foundational ontology. Applied ontology **17**(1), 167–210 (2022)
[9] Guizzardi, G., Guarino, N., Almeida, J.P.A.: Ontological considerations about the representation of events and endurants in business models. In: Int. Conf. Business Process Mngmt. pp. 20–36. Springer (2016)
[10] Guizzardi, G., Wagner, G.: Conceptual simulation modeling with Onto-UML. In: Proceedings of the Winter Simulation Conference. WSC '12, Winter Simulation Conference (2012)
[11] Guizzardi, G., Wagner, G., Almeida, J.P.A., Guizzardi, R.S.S.: Towards ontological foundations for conceptual modeling: The Unified Foundational Ontology (UFO) story. Appl. Ontology **10**(3-4) (2015)
[12] Josey, A., Lankhorst, M., Band, I., Jonkers, H., Quartel, D.: An introduction to the archimate® 3.0 specification. White Paper from The Open Group (2016)
[13] Lima, J.A.O., Passos, E.: LexML – visão unificada da informação legislativa e jurídica do brasil. Cadernos de Informação Jurídica (Cajur) **6**(1), 248–259 (jun 2019)
[14] Palmirani, M., Governatori, G.: Modelling legal knowledge for GDPR compliance checking. In: JURIX. vol. 313, pp. 101–110 (2018)
[15] Suchánek, M., Pergl, R.: Mapping UFO-B to BPMN, BORM, and UML activity diagram. In: EOMAS 2019. LNBIP, vol. 366, pp. 82–98. Springer (2019)
[16] Weldemariam, K., Villafiorita, A., Siena, A., Susi, A.: Enhancing law modeling and analysis: Using BPR-based and goal-oriented frameworks. Int. Journal Advances in Security **3**(3), 80–90 (2011)
[17] Zaneti Jr, H.: O valor vinculante dos precedentes: teoria dos precedentes normativos formalmente vinculantes. JusPodivm (2021)

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220479

Can a Military Autonomous Device Follow International Humanitarian Law?

Tomasz ZUREK [a,1], Mostafa MOHAJERIPARIZI [b], Jonathan KWIK [c], Tom VAN ENGERS [b]

[a] *T.M.C. Asser Institute, The Hague*
[b] *Complex Cyber Infrastructure, Informatics Institute, University of Amsterdam*
[c] *Faculty of Law, University of Amsterdam*
ORCiD ID: Tomasz Zurek https://orcid.org/0000-0002-9129-3157, Jonathan Kwik https://orcid.org/0000-0003-0367-5655, Tom van Engers https://orcid.org/0000-0003-3699-8303

Abstract. The paper presents a formal model and an experimental verification of the system controlling the International Humanitarian Law compliance for the autonomous military device.

Keywords. military autonomous device, International Humanitarian Law, reasoning model, experimental analysis

1. Introduction

Military autonomous devices remain an object of a constant debate, with the main controversies related to moral and legal issues. In particular, it is frequently argued that incorporating many principles of international humanitarian law (IHL), such as distinction, proportionality, and precautions, into an AI is impossible [1,2]. In opposition to this, other commentators [3,4] have noted that the possibility of IHL-compliant military AI should not be immediately discarded, particularly in light of the advantages a well-functioning AI can provide in the form of better performance and increased respect for the law [5,6]. In this paper we examine the possibility of implementing an IHL-compliance controlling mechanism and perform its experimental verification. On the basis of the experiment we discuss what kinds of data are required to perform necessary legal tests and point out the main difficulties of its implementation. We develop the mechanism described in [7] by introducing a fully-fledged formal representation of IHL rules and their implementation with the use of ASC2 and eFLINT languages.

[1] Corresponding Author: Tomasz Zurek, t.zurek@asser.nl. Tomasz Zurek received funding from the Dutch Research Council (NWO) Platform for Responsible Innovation (NWO-MVI) as part of the DILEMA Project on Designing International Law and Ethics into Military Artificial Intelligence.

2. International Humanitarian Law rules

International humanitarian law (IHL) is the body of rules applicable to all military operations, including weapons release [8]. Attack decisions and weapons systems that do not comply with IHL principles are unlawful and may even entail the decision-maker's criminal liability for any harm that results [9]. These principles include guaranteeing that the weapon is sufficiently accurate so as to not be indiscriminate, that attacks are proportionate, and all necessary precautions are taken to spare the civilian population.

For our purposes, IHL principles related to targeting and weaponeering are particularly relevant, which are implemented through a series of legal tests during the targeting process [10,11,12]. The authors of [7] structured and streamlined these legal tests for implementation by a hypothetical military autonomous device. In the current paper, we will limit our discussion to the implementation of tests which are commonly described [4] as the most difficult tasks for an artificial agent to perform, namely those which involve the incidental harm (IH) and military advantage (MA) variables. The tests in question are the *proportionality rule* and the *two minimisation rules* (see Section 3.3).

3. The model

The general structure of our model has been presented in [7]. The key point of the model lies in the analysis of various relations between MA and IH. In our model they are expressed by two values: v_{MA} representing Military Advantage and v_{CIV} representing civilian well-being (inversely proportional to IH). In this section we introduce the three layers of the legal analysis conducted by the military autonomous device.

3.1. First layer: Data preparation

In this layer the system prepares the data needed to perform legal tests. In this paper we assume that necessary data have already been prepared. An initial discussion of the topic of obtaining this data was introduced in [7]. We realize that some functionalities may still be very difficult to implement in real life systems (e.g. identifying direct participation in hostilities) [2]. However, we can expect that such modules, at least for some tasks (e.g. distinguishing military from civilian aircraft), will be feasible in the near future. Below follows the list of the data necessary to reason about the legality of a military autonomous agent's behaviour:

- The set of propositions $D = \{d_x, d_y, ...\}$ representing the available decisions.
- The set of evaluations of the results of decisions in the light of two values: v_{civ} and v_{MA}. The extents to which every value is satisfied by the results of the decisions are denoted by $V = \{v_{civ}(d_x), v_{MA}(d_x), v_{civ}(d_y), v_{MA}(d_y), ...\}$. Every evaluation is expressed by a real number.
- The proportionality coefficient p, a real number declared in advance, represents the level of acceptable (from the point of view of IHL) relations between military advantage and incidental harm to fulfil the Proportionality test.

3.2. The second layer: Weighting of MA and IH

Both MA and IH are, in our model, expressed by numbers. Since the framework requires a logical representation of norms, we have to introduce an intermediate layer of the analysis of decisions. The role of the weighting layer is to examine the relations between the levels of satisfaction of MA and IH. This problem of balancing appears in three tests in particular: (1) Article 57(3) test, (2) Proportionality test, and (3) Minimisation of Incidental Harm test. The three tests mentioned above require four kinds of weighting between v_{MA} and v_{CIV}[2]:
(1) Test whether two different decisions satisfy Military Advantage to the same level. If by d_x and d_y we denote two different decisions then by $EQMA(d_x,d_y)$ we denote that both decisions satisfy MA to the same level: $(ev_{MA}(d_x) = ev_{MA}(d_y)) \rightarrow EQMA(d_x,d_y)$
(2) Test whether one of two decisions satisfy v_{CIV} to a greater extent than the other. By $LESSCIV(d_x,d_y)$ we denote that d_x satisfies value v_{CIV} to a lower extent than d_y:
$ev_{Civ}(d_x) < ev_{Civ}(d_y) \rightarrow LESSCIV(d_x,d_y)$
(3) Test whether the level of satisfaction of the well-being of civilians (v_{CIV}) by results of a given decision, multiplied by a certain coefficient, is higher than the level of satisfaction of MA by the same decision. In other words, this tests whether military advantage is proportionate to a change in the well-being of civilians. By $PROP(d_x)$ we denote that decision d_x brings about results which satisfy the test: $ev_{MA}(d_x) \leq p * ev_{Civ}(d_x) \rightarrow PROP(d_x)$
(4) Compare relations between the extents of satisfaction of MA and IH obtained by two decisions. By $MOREREL(d_x,d_y)$ we denote that the relation of MA to IH for d_x is higher than it is for d_y: $ev_{MA}(d_x) * ev_{Civ}(d_x) \geq ev_{MA}(d_y) * ev_{Civ}(d_y) \rightarrow MOREREL(d_x,d_y)$

3.3. Third layer: legal rules

On the basis of the predicates introduced in the previous section, we introduce a set of legal rules representing tests necessary to examine whether a given decision is lawful from the point of view of IHL. We use standard logical expressions to represent the above tests:

Article 57(3) test. This provision provides that if more than one target is viable and they produce comparable MA, the target with the lowest IH should be selected. We will represent it by the predicate $DT(d_x)$, where d_x is the decision which satisfies the test: $\exists_{d_x \in D} \neg \exists_{d_y \in D} (EQMA(d_x,d_y) \wedge LESSCIV(d_x,d_y)) \Rightarrow DT(d_x)$

Proportionality test. By predicate $DP(d_x)$ we denote that decision d_x passes the proportionality test: $PROP(d_x) \Rightarrow DP(d_x)$. Where p is the proportionality coefficient. $DP = \{d_x | DP(d_x)\}$

Minimisation of incidental harm. By $DMH(d_x)$ we denote that a decision d_x passes the minimisation test: $\exists_{d_x \in D} \forall_{d_y \in D} (MOREREL(d_x,d_y)) \Rightarrow DMH(d_x)$

Rule of IHL. A given targeting decision will be coherent with IHL if all the above tests are fulfilled. Therefore, on the basis of all the defined earlier predicates we can create a rule describing whether a given decision will follow IHL. By $DAV(d_x)$ we denote decision d_x fulfills the requirements: $DT(d_x) \wedge DP(d_x) \wedge DMH(d_x) \wedge DG(d_x) \Rightarrow DAV(d_x)$

On the basis of the above formulae we can distinguish a set of legal decisions, i.e. decisions which fulfill IHL rules. The decision-making system can choose one of those decisions to fulfill a military goal. A brief description of such a mechanism is presented

[2]Note that we use expected levels of satisfaction of values instead of absolute ones.

in Section 5.1 (an experimental implementation can be found on our github[3]), but full discussion of the decision making process, including the possibility of reconsideration of previous decisions, will be reserved for another paper.

4. Scenario

To test our mechanism, we compiled a scenario involving a hypothetical drone tasked with disabling an enemy's signal intelligence network, which it can achieve by targeting one of the key network points for each district. Complexity is added by introducing variables in the form of the network points' locations, added MA from neutralising enemy personnel around the target, added IH from the amount of civilian persons and objects damaged by an attack, and two types of missiles the drone can select from when engaging. We conceived three different subscenarios with different values for each of the variables, and tested whether our mechanism allows the drone to select the correct (i.e., IHL-compliant) target and means of engagement. We find that it does. For further details on this scenario and the test results, see the project's github.

5. Implementation

This section presents the basics of the implementation of the experiment. The proof of concept is implemented in two components: (1) an intentional agent that encapsulates the objectives and procedural knowledge that is implemented utilizing ASC2 framework and (2) a normative advisor that encompasses the the normative aspects i.e., rules that are implemented with ASC2 and eFLINT norms framework. The main advantage of using intentional agents and normative advisors is the separation of the analysis of legality of the decision from making the decision itself. Such a separation is important because it preserves the required level of transparency concerning the IHL compliance: in particular, it allows for clear understanding why a given decision fulfills a particular IHL rule. Since the main goal of our work is to discuss the experiments concerning the recognition whether a given decision option fulfills IHL requirements (i.e. if it is lawful an IHL perspective), we will focus on a particular element of a normative advisor (component 2), i.e. the normative reasoner, which is responsible for performing the legal tests. The normative reasoner is implemented with the use of eFLINT framework (discussed in section 5.2). Section 5.1 presents briefly the details of component 1, leaving a discussion of the complete decision process to another paper.

5.1. Intentional agents

Intentional agents are generally approached in the computational realm via the *belief-desire-intention* (BDI) model [13]. In practice, BDI agents also include concepts of *goals* and *plans*. Goals are concrete desires, plans are abstract specifications for achieving a goal, and intentions then become commitments towards plans. Our implementation was made with the use of AgentScript/ASC2 [14] language.

[3] https://github.com/mostafamohajeri/jurix2022-ihl-devices

5.2. The eFLINT norm language

The eFLINT language is a DSL designed to support the specification of (interpretations of) norms from a variety of sources (laws, regulations, contracts, system-level policies such as access control policies, etc.) [15]. The language is based on normative relations proposed by Hohfeld [16]. The type declarations introduce types of *facts*, *acts*, *duties* and *events*, which together define a transition system in which states—knowledge bases of facts—transition according to the effects of the specified actions and events.

The script defines multiple types of facts, some atomic ones like `target` and `vma`, some composite ones like `outcome` and the rest are *derived* facts. Some examples are: The fact `evciv(target,value)` which derives the expected value of civilian well-being for a target from all the possible outcomes of that target or the fact `proportionate(target)` which is derived from the proportionality formula in Section 3.3. Note that eFLINT by design includes a transition system that on every update proactively searches for all the possible facts (or acts, or duties).

6. Discussion of results

In the experiment, the list of available decisions with their evaluations is sent to the intentional agent in a sequence. After the last decision is sent, the system inspects the norms instance embedded in the advisor to see which facts are present. The results of the IHL compliance analysis are presented on the project's github. Although our eFLINT-based normative reasoning mechanism is relatively simple, the results obtained (even for controversial cases) are correct.

The problem of balancing was widely discussed in a number of AI and Law papers and legal case-based reasoning in particular. In legal CBR, the objects of comparison are either dimensions (e.g. [17]) or values (e.g. [18]). The key difference between our model and the existing ones is in the level of abstraction: both V_{MA} and V_{CIV} have a very abstract character, especially in comparison to dimensions like *number of disclosures*. An important difference also lies in the absolute representation of the level of satisfaction of values, whereas in other models of balancing, the levels of values' promotion was represented in a relative way (in comparison to other decisions, state of affairs, etc.; e.g. [19]). Moreover, in contrast to many argumentation or legal reasoning models [20,21], values in our model are not an external element of a reasoning process allowing for solving conflicts between arguments, but they are an element of a legal rule itself. The simplicity of our model, however, shows that the critical point of the reasoning process is not located in the legal reasoning, but in the calculation of the specific relations between v_{MA} and v_{CIV}. Such an observation allows us to derive a more general conclusion: the key difficulty of targeting compliance testing lies not in the legal reasoning and balancing itself, but in the process of evaluating the available options.

In practice, obtaining v_{MA} and v_{CIV} can be seen as a classification or regression task, which can be expressed as assigning numbers (representing v_{MA} and v_{CIV}) to particular decisions (represented by their specific parameters). The key question is whether the creation of such a regression mechanism is feasible at all. Answering this question will be an important topic for future research.

References

[1] Crootof R. The Killer Robots are here: Legal and Policy Implications. Cardozo Law Review. 2015;36:1837-915.

[2] Szpak A. Legality of Use and Challenges of New Technologies in Warfare – the Use of Autonomous Weapons in Contemporary or Future Wars. European Review. 2020 feb;28(1):118-31.

[3] Anderson K, Waxman MC. Law and Ethics for Autonomous Weapon Systems: Why a Ban Won't Work and How the Laws of War Can; 2013.

[4] Boothby WH. New Technologies and the Law of War and Peace. Cambridge: Cambridge University Press; 2019.

[5] Defense Innovation Board. AI Principles : Recommendations on the Ethical Use of Artificial Intelligence by the Department of Defense Defense Innovation Board. Department of Defense; 2019.

[6] Jurecic Q. Paul C. Ney Jr., General Counsel, U.S. Department of Defense, Keynote Address at the Israel Defense Forces 3rd International Conference on the Law of Armed Conflict. Lawfare. 2019 may.

[7] Kwik J, Zurek T, van Engers T. Designing International Humanitarian Law into Military Autonomous Devices; 2022. https://ssrn.com/abstract=4109286.

[8] Fleck D, editor. The Handbook of International Humanitarian Law. 3rd ed. Oxford: Oxford University Press; 2013.

[9] Additional Protocol I. Protocol Additional to the Geneva Conventions of 12 August 1949, and relating to the Protection of Victims of International Armed Conflicts (adopted 8 June 1977, entered into force 7 December 1978) 1125 UNTS 3; 1977.

[10] Ducheine P, Gill T. From Cyber Operations to Effects: Some Targeting Issues. Militair Rechtelijk Tijdschrift. 2018;111(3):37-41.

[11] von Heinegg WH. Considerations of Necessity under Article 57(2)(a)(ii), (c), and (3) and Proportionality under Article 51(5)(b) and Article 57(2)(b) of Additional Protocol I. In: Kreß C, Lawless R, editors. Necessity and Proportionality in International Peace and Security Law. Oxford: Oxford University Press; 2020. p. 325-42.

[12] Corn GS. War, law, and the oft overlooked value of process as a precautionary measure. Pepperdine Law Review. 2014;42:419-66.

[13] Rao AS, Georgeff MP. BDI Agents: From Theory to Practice. In: Proceedings of the First International Conference On Multi-Agent Systems (ICMAS-95); 1995. p. 312-9.

[14] Mohajeri Parizi M, Sileno G, van Engers T, Klous S. Run, Agent, Run! Architecture and Benchmarking of Actor-Based Agents. New York, NY, USA: Association for Computing Machinery; 2020. p. 11–20.

[15] van Binsbergen LT, Liu LC, van Doesburg R, van Engers T. EFLINT: A Domain-Specific Language for Executable Norm Specifications. In: Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences. New York, NY, USA: Association for Computing Machinery; 2020. p. 124–136.

[16] Hohfeld WN. Fundamental Legal Conceptions as Applied in Judicial Reasoning. The Yale Law Journal. 1917;26(8):710-70.

[17] Bench-Capon TJM, Atkinson K. Dimensions and Values for Legal CBR. In: Wyner AZ, Casini G, editors. Legal Knowledge and Information Systems - JURIX 2017: The Thirtieth Annual Conference, Luxembourg, 13-15 December 2017. vol. 302 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2017. p. 27-32.

[18] Bench-Capon T, Prakken H, Wyner A, Atkinson K. Argument Schemes for Reasoning with Legal Cases Using Values. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. ICAIL '13. New York, NY, USA: ACM; 2013. p. 13-22.

[19] Grabmair M. Predicting Trade Secret Case Outcomes Using Argument Schemes and Learned Quantitative Value Effect Tradeoffs. ICAIL '17. New York, NY, USA: Association for Computing Machinery; 2017. p. 89–98.

[20] Bench-Capon TJM. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. Journal of Logic and Computation. 2003 06;13(3):429-48.

[21] Zurek T, Araszkiewicz M. Modeling teleological interpretation. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. ACM; 2013. p. 160-8.

Demo Papers

This page intentionally left blank

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220481

Toward an Integrated Annotation and Inference Platform for Enhancing Justifications for Algorithmically Generated Legal Recommendations and Decisions

Yi-Tang Huang[♜] Hong-Ren Lin[‡] Chao-Lin Liu[†]
National Chengchi University, Taiwan
{108753132[♜],109753156[‡], chaolin[†]}@nccu.edu.tw

Abstract. We introduce our workflow that integrates the steps of annotation and classification, and hope that the end products are helpful for improving the justifications for legal reasoning and for recommending similar civil cases.

Keywords. technology-assisted annotation, annotation and prediction and inference, system integration, machine learning, natural language processing

1. Overview

Convincing justifications strengthen the usability of the legal recommendations and decisions that are produced by algorithmic computations [1][3]. A legal informatics system may offer similar cases for preparing cross-examinations in courts, and may even attempt to predict the sentences against the defendants. Such assistive systems, which are constructed by the machine learning (ML) approaches, typically rely on training data to learn to select the recommendations and decisions. An ML-based predictive procedure that aims to offer satisfactory recommendations and decisions would be more useful if we can associate their outputs with appropriate supporting evidences.

We believe that such supporting evidences require at least a few high-quality annotated data for training the predictive procedure. Given a collection of original judgment documents, we use existing tools for lexical, syntactical, semantic, and even pragmatic analysis to mark the texts. Human experts can verify and correct the raw annotations. Our system also allows the annotators to read, find and mark the statements for high-level legal factors for specific categories of lawsuits. The annotated data will be used to train a new generation of tools, hopefully improving the quality of future annotations.

Currently, we use the open judgment documents of the courts in Taiwan in our system. We believe that the system architecture can be adopted by systems of legal informatics in any other languages. If the proposal is accepted, we will show the current operations of the annotation system both on site and online during the Conference.

2. Architecture and Raw Data Selection

Figure 1 shows the system architecture for the integrated system. To modularize our presentation, we show the architecture in four blocks, which are indicated on the left margin in Figure 1.

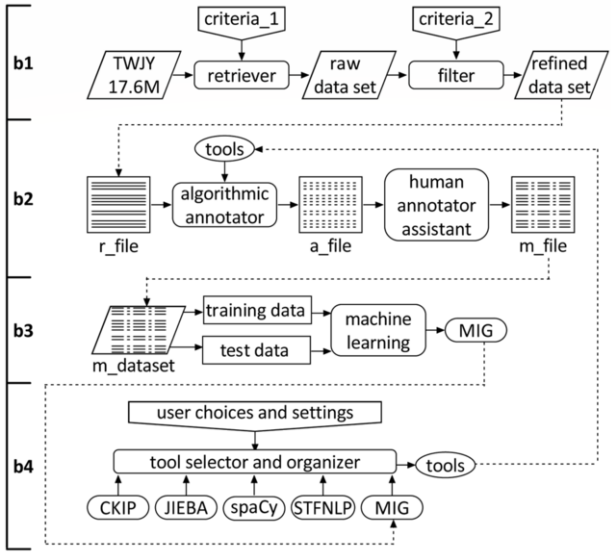


Figure 1. System architecture.

The block **b3** shows that we need an annotated dataset, **m_dataset**, to support the machine-learning approaches to train a classifier, **MIG**. We split the annotated dataset into two parts for training and test to produce the classifier.

In order to obtain the annotated dataset, we first need to extract appropriate files from a large collection of judgment documents. In our current work, we use the open data repository of the Judicial Yuan of Taiwan as the main source. We show this repository as **TWJY** in block **b1**. There are about 17.6 million documents in TWJY as of April 2022. Assuming that our research focuses on a specific category of cases, say burglary. We can use "burglary" as a keyword in **criteria_1**, and let the **retriever** extract documents that are related to burglary from TWJY. The resulting **raw data set** usually includes some documents that do not really fit the research requirements from the legal informatics perspective. Hence, we would consult the expertise of legal experts, and use the **filter** to remove some files from the raw data set based on **criteria_2** to produce the **refined data set**. After this step, our system offers an interface via which a human expert can select to read a specific document to determine whether to annotate a file in the set.

Depending on the research issues of the researchers, the size of refined data set may not be large, even though we have millions of documents in TWJY. TWJY includes documents for lawsuits of all possible criminal, civil, and other special cases of between 1996 and April 2022. Some types, e.g., burglary and gambling, can be relatively common, but some are relatively infrequent, e.g., verification of presumption of paternity.

3. The Annotation Procedure

The block **b2** shows the main steps for the annotation task. Usually, the human annotator chooses to annotate one file, **r_file** ("r" for raw and refined), from the refined data set at a time. In fact, depending on the needs of the research, an annotator may just need to examine some specific parts (sections) of the documents, e.g., the *facts and reasons*, and do not need to inspect the whole judgment document. If you may find the "**理由**" in eighth line of the center window of Figure 2, that is the Chinese word for reasons.

Figure 2. A sample of a_file and the current human annotator assistant.

Figure 2 shows a sample of **a_file** ("a" for annotated) and the current implementation of the **human annotator assistant**. In the center of Figure 2 is a sample of a_file. The selected r_file was annotated by the **algorithmic annotator**. The algorithmic annotator uses the tools that are set up in the **tools** component (to be explained shortly). In this example, the tools component is just the *named entity recognition* (NER) component of the CKIP toolset [2]. It is our software that colors the outputs of the CKIP-NER by the types of entities, and coverts the file format for the reading and annotation interface. The current CKIP-NER identifies seven categories, as is shown on the upper right corner of Figure 2, where "ORG" and "GPE" refer to organizational and geographic entities, respectively. The categories of the words in the text are indicated by their colors.

We can equip the human annotator assistant with multiple functions. The most basic ones include allowing the annotators to *create*, *update*, and *delete* the annotations that are suggested by the algorithmic annotator, whose suggestions may not be perfect.

It is evident that the seven categories of entities that the CKIP-NER attempts to identify is insufficient for the needs of legal reasoning. For the cases of burglary, it might be preferable to annotate the places of the events, the thief, the stolen objects and their values, and perhaps whether the thief is armed. An annotator can add such additional categories in the box on the upper right corner of Figure 2, where we put a "TEST" box as an example. An annotator can add a new tag, then chooses texts in the center window, and labels the chosen texts with the new tag. More than one tag can be added in a session.

The RELabel button leads the annotators to enter an interface for editing regular expressions to define patterns. A pattern can represent an established way to express a legal notion in the text, and will be assigned a tag. Defining patterns for legal notions in regular expressions allows the software to identify the patterns automatically for us. A more complex system for annotation could offer a mechanism for the annotators or the research team to maintain and share useful domain-dependent regular expressions. The RELabel button in Figure 2 offer the annotators to try their instincts for automatic annotations on the fly.

The Download button allows the annotators to save the results of the current annotation, i.e., **m_file** in Figure 2. A previously annotated file can also be uploaded to the annotation system again so that the annotators can revise the annotations. To protect the previously annotated results from being overwritten, the algorithmic annotator will be suppressed for reloaded files.

4. Progressively Powerful Models

We gather a sufficient number of m_files to obtain the m_dataset in b3, and run whatever types of machine learning procedures we plan to. In **b4**, we show that we may use whatever software tools as the algorithm annotator, including JIEBA, spaCy, Stanford NLP tools (STFNLP in Figure 1).

Whenever possible and desirable, we may use our own model, **MIG**, as one of the possible tools to annotate the documents. This is not only possible but should be more reasonable when the size of the refined data set is large such that it is unreasonable and inefficient to expect human experts to inspect and annotate all of the data files all at once.

Assume that we have N files in the refined data set and that N is large. We can split the refined data set into p parts, each having $\frac{N}{p}$ files. Assume further that, we choose one of the tools in b4, but not MIG, as the algorithmic annotator to annotate an installment of $\frac{N}{p}$ files in the first iteration. The annotator may examine, check, and revise annotations of these $\frac{N}{p}$ files. We can use these files to train an MIG, and use this MIG in the next run.

Assume that the MIG can perform better than the other choices, because it is directly trained with our annotated data, but the other tools do not have this privilege. Hence, starting from the second iteration, we choose MIG as the algorithmic annotator. In the second run, we use the second installment of $\frac{N}{p}$ files. Since MIG is a better model, we expect the quality of the annotations in the a_file is better than before. As a result, it is easier for the human annotator to create, update, and delete labels this time. This type of improvement can continue in the following runs at least on average. Hence, splitting N files in the refined data set into parts can make our annotation process more efficient.

We have assumed that the MIG we obtained in the first iteration will outperform the other tools. If this is not true, we may wait until MIG can outperform others before we switch to our own MIG in b4. If MIG continues to underperform, we probably should improve our machine learning procedure in block b3 first.

5. Concluding Remarks

We have used our own annotation tools for our work on classifying statements by their functions in judgment documents for civil cases. Statements for civil cases typically are more indecisive than those for criminal cases, so the availability of human annotations is more helpful and necessary. We attempt to identify the statements of the requests of

the plaintiffs, of the responses of the defendants, and of the viewpoints and decisions of the judges. More importantly, we attempt to automatically find the conflicting issues between the plaintiffs and the defendants. Based on these results, we hope to find "similar" civil cases. The results of this application-oriented research remain promising.

Acknowledgements

This research was supported in part by the project 110-2221-E-004-008-MY3 of the National Science and Technology Council of Taiwan.

References

[1] Atkinson K, Bench-Capon T, and Bollegala D. Explanation in AI and law: Past, present and future. *Artificial Intelligence*. 2020; 289:103387.

[2] Li P-H, Fu T-J, and Ma W-Y. Why Attention? Analyze BiLSTM Deficiency and Its Remedies in the Case of NER. *Proc. of the 33rd AAAI Conference on Artificial Intelligence*. 2020; p. 8236–8244.

[3] Mumford J, Atkinson K, and Bench-Capon T. Machine learning and legal argument. *Proc. of the 21st Workshop on Computational Models of Natural Argument*. 2021; p. 47–56.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220482

The LegAi Editor: A Tool for the Construction of Legal Knowledge Bases

Tomer Libal
University of Luxembourg
ORCiD ID: Tomer Libal https://orcid.org/0000-0003-3261-0180

Abstract. A key challenge in legal knowledge representation is the construction of formal knowledge bases. Such knowledge bases then allow for various applications such as searching and reasoning. In this paper we describe a web application for the annotation of legal texts. The result of the annotation is a formal representation of the legal texts in a form which might be used for searching and reasoning. The novelty of the tool is threefold: it provides a wizard which guides the user in the annotation process; it uses a high-level annotation language which is close to the language of the original text; and, it allows validation of the formal structure by allowing a simple comparison of the original and the annotated versions of the legal text.

Keywords. Legal Knowledge Representation, Automated Reasoning, Annotation Editor

1. Introduction

Information systems are playing an important role in helping people in a wide range of tasks, ranging from searching to decision-making.

One area in which such tools can contribute is the legal domain: New court cases and legislations are accumulated every day. In addition, international organizations like the European Union are constantly aiming at combining and integrating separate legal systems [4].

Approaches for searching over legal texts have long seen commercial success [17]. At the same time, some tools for reasoning over sets of norms have been developed, such as for business (e.g. [6]) and law (e.g. [12] and [9]).

Unlike the tools and methods for searching, which have seen a wide commercial success and plans are in place for supporting them, those for reasoning have seen a much smaller success, especially by legal practitioners.

One of the reasons for the relatively limited success of applications of automated reasoning in the legal domain is the lack of tools for the construction of legal knowledge bases, which can be used by non-logicians.

Various surveys of the available tools (e.g. [11]), have identified usability by domain experts as one of the main requirements. Nevertheless, most such tools fail in achieving this goal [15].

The most popular approach is by the use of programming languages (a classical example can be found in [14]). This approach has enjoyed success [3] in the last 40 years

and is still popular today. Nevertheless, basic knowledge of logic programming is still required. In order to have a wider use of legal reasoning, other professionals, such as lawyers and jurists, should be able to use the tools.

Various attempts have been done in order to make such tools more accessible to legal practitioners (e.g. [7]). The advantage of these approaches is the ability to compile and test the correctness and quality of the knowledge.

Another approach, demonstrated in [13], is by the use of general formats and languages. Being of general formats, one can more easily utilize a language which is most suited for capturing the semantics of legal texts.

A disadvantage of all the approaches above is the lack of tools and methodologies for asserting the correctness of the logical representations of the legal texts. A discussion about the need can be found in [8][1].

Among existing validation results, one can find a methodology for building legal ontologies [10] and more concretely to our approach, one for validating formal representations of legal texts [1].

In this paper, we demonstrate the LegAi annotation editor. The focus of this editor is on the ability to validate the quality of the formalized legal texts, as well as on usability. Nevertheless, the editor uses a specific formal language, which can be transformed into first-order Deontic logics.

The paper will those be divided into three parts: we first describe the formal language used; in the second part, the annotation editor and its user interface are presented; we end by presenting an approach for ensuring the quality of the legislation.

2. The Legal Linguistic Templates

In the automated reasoning community there is a discussion whether the knowledge representation language should be as expressive as possible, or be efficiently reasoned over [5]. Expressivity refers to the ability of the language to directly capture the nuances of the target language, which in our case is the legal language. Being efficient to reason over refers to the ability to use existing and efficient tools to reason over formulae denoted in this language.

In the present paper, we take the former approach, since expressivity is essential to both an easy to use editor (section 2) and validation (section 3). At the same time, among our goals is to support efficient reasoning over formulae denoting legislations.

Our approach to deal with this apparent paradox is to put requirements on the expressive language. The main requirement is the ability to algorithmically translate formulae into a language in which efficient theorem proving is possible, which will be called the ”lower” language, as opposed to the ”higher”, expressive, language.

The LegAi lower language consists of classical first-order logic (see, e.g. [2]), to which we add a second, non-classical, negation operator and one ”Obligation” operator.

The LegAi higher language consists of vocabulary and Legal Linguistics Templates (LLTs).

Vocabulary and **LLTs** can be considered as logical terms and connectives, respectively. As an example, consider the vocabulary $personalData(DataSubject)$, which is

[1] See discussion in Sec. 3.2 .

a first-order term with one parameter, which is a free variable. An example of an LLT is $General - Prohibition(List)$, which accepts a list of other LLTs and denotes a legal prohibition, which can be overcome later via exceptions. Free variables are universally quantified over the whole legal knowledge base. So far, we have not seen a need to scope variable quantification locally.

Each LLT comes with its own "semantics", which are denoted in the lower language. The semantics are denoted recursively. For example, given the semantics operator $\mathscr{S}$:

$$\mathscr{S}\{General - Prohibition(List))\} = Obligation(\neg\mathscr{S}\{List\}) \quad (1)$$

3. The LegAi Annotation Editor

The goal of the LegAi annotation editor is to support annotating legal texts with the LegAi higher language, while at the same time, supporting validation of this annotations, which will be described in the next section.

The editor can be found online[2] and requires registration. After registration, an email to activate the account must be sent to the code maintainer, who associates the new user to a specific account with specific rights.

Once logged in, the user has the ability to paste legal texts into an annotation editor. The user starts with selecting a sentence they wish to annotate and then a (top-level) LLT wizard opens. The wizard guides the user in the annotation process, shows the possible/required LLTs that can/must be nested.

An example of annotating two simplified articles of the GDPR is shown on the top left of the figure above.

4. Ensuring Correctness via a Reverse Translation

In order to check if an annotation captures correctly the meaning of a legal sentence, the user can view the formal tree structure, as defined in the previous section and shown on the right of the figure above.

Nevertheless, such a formal structure is not easily readable by non-logicians. In order to rectify that, a feature called reverse translation was added to the editor.

The reverse translation, which is accessible on a separate tab, generates a legal text from the formal tree structure. It then places the reversely generated text next to the original text. The user can then compare the two versions and decide if the formal version correctly captures the meaning of the original one, as shown on the bottom left of the figure above.

[2] https://legai.uni.lu

5. Conclusion

In this paper we have presented the LegAi annotation editor. This editor is currently used by legal firms in order to generate legal knowledge bases for specific legal problems.

The editor is accompanied by the LegAi decision support tool (not described in this paper), which allows automated reasoning over the knowledge bases.

The LegAi higher language is currently translated only into the LegAi lower language. In the future, we plan on translating the higher language into TPTP [16] and other formats. The knowledge base, denoted in the higher language, can be exported to be used with other tools.

References

[1] Cesare Bartolini, Gabriele Lenzini, and Cristiana Santos. An interdisciplinary methodology to validate formal representations of legal text applied to the gdpr. In JURISIN, 2018.
[2] Barwise, Jon. "An introduction to first-order logic." Studies in Logic and the Foundations of Mathematics. Vol. 90. Elsevier, 5-46, 1977.
[3] Trevor JM Bench-Capon et al. Logic programming for large scale applications in law: A formalisation of supplementary benefit legislation. In Proceedings of ICAIL, pages 190–198. ACM, 1987
[4] Anne-Marie Burley and Walter Mattli. Europe before the court: a political theory of legal integration. International organization, 47(1):41–76, 1993.
[5] Benzmüller, Christoph. "Universal (meta-) logical reasoning: Recent successes." Science of Computer Programming 172 : 48-62, (2019).
[6] Mustafa Hashmi and Guido Governatori. Norms modeling constructs of business process compliance management frameworks: a conceptual evaluation. Artif. Intell. Law, 26(3):251–305, 2018
[7] Kowalski, Robert. "Logical english." Proceedings of Logic and Practice of Programming (LPOP), 2020.
[8] Libal, Tomer. "Towards automated GDPR compliance checking." International Workshop on the Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning. Springer, Cham, 2020.
[9] Libal, Tomer, and Matteo Pascucci. "Automated reasoning in normative detachment structures with ideal conditions." Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. 2019.
[10] Martynas Mockus and Monica Palmirani. Legal ontology for open government data mashups. In 2017 Conference for E-Democracy and Open Government (CeDEM), pages 113–124. IEEE, 2017.
[11] Novotna, Tereza and Tomer Libal. "An Evaluation of Methodologies for Legal Formalization". Proceedings of the "International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems. Springer, 2022.
[12] Monica Palmirani and Guido Governatori. Modelling legal knowledge for gdpr compliance checking. In Legal Knowledge and Information Systems: JURIX, volume 313, pages 101–110. IOS Press, 2018.
[13] Robaldo, Livio, et al. "Formalizing GDPR provisions in reified I/O logic: the DAPRECO knowledge base." Journal of Logic, Language and Information 29.4 : 401-449. 2020.
[14] Sergot, Marek J., et al. "The British Nationality Act as a logic program." Communications of the ACM 29.5 : 370-386, 1986.
[15] Soavi, Michele, et al. "From Legal Contracts to Formal Specifications: A Systematic Literature Review." SN Computer Science 3.5 : 1-25. 2022.
[16] Sutcliffe, Geoff. "The TPTP problem library and associated infrastructure." Journal of Automated Reasoning 43.4 : 337-362. 2009.
[17] Weaver, David A., and Bruce Bimber. "Finding news stories: a comparison of searches using LexisNexis and Google News." Journalism & Mass Communication Quarterly 85.3 515-530, 2008.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220483

Scribe: A Specialized Collaborative Tool for Legal Judgment Annotation

Sid Ali MAHMOUDI [a,1], Guillaume ZAMBRANO [a], Charles CONDEVAUX [a] and Stéphane MUSSARD [a]
[a] CHROME, *University of Nîmes, France*

Abstract. *Scribe* is a legal judgment annotation platform. Its objective is to improve dataset quality for machine learning models, to make annotation task faster and easier, and to boost interactions between *annotators* and *developers*. The platform manages 3 different classes of annotation: claims, named entities and sections. The platform facilitates the expression of annotation needs by developers. Multiple annotators can quickly respond to these needs by working in parallel. The collaborative process ends when the expected model performance is reached. The platform is organized by modules, maintainable and extensible in addition to its flexibility and unified output result in JSON format. See our demo https://lawbot.unimes.fr/annotateur

Keywords. Case law, Judiciary, Dataset annotation, Legal NLP, Legal Judgment Prediction

1. Introduction

These last years, numerous Legaltech projects arose, for instance, to cite a few of them, Lexmachina, Doctrine and Caselaw analytics. The strategy of Legaltechs is clear, based on legal documents and court decisions, they employ machine learning and natural language processing (NLP) techniques to predict the outcome of a dispute: the acceptance or rejection of a claim, the quantum, the legal fees, etc.

Accordingly, organizing a massive quantity of unstructured data related to court decisions with the aid of legal experts is crucial. Based on their annotations, aiming at bringing out keywords and sentences (legal norms, names of the parties, jurisdiction, type of claim, facts, outcome, etc.), the structured data may be stored, before employing NLP models such as models for named entity recognition (NER) in order to automatically annotate all documents.

To our knowledge, there are a few researches about the design of collaborative platforms for annotating court decisions. Recently, [3] use a machine learning model to classify documents in different categories of claims and present a pseudo-platform allowing to automatically classify court decisions in those categories. However, this platform can-

[1] Corresponding Author: Sid Ali Mahmoudi, email: sid.mahmoudi@unimes.fr
This research was supported by the Occitanie region for Sid Ali Mahmoudi's PhD scholarship. The French National Research Agency is acknowledged for funding the LAWBOT project ANR-20-CE38-0013: Deep Learning for Judicial Outcome Prediction.

not be used to manage court decisions from their storage to the production of categories based on NER. [6] proposes a tool to annotate named entities with the possibility to automatically annotate court decisions. Other general-purpose annotation tools exist like Prodigy (https://prodi.gy/), but they are not adapted to the legal domain. Of course, the development of collaborative platforms to perform annotations is possible because of the emergence of new annotations techniques, see e.g. [4] and [7]. The aim of this paper is to present *Scribe*, a platform for court decisions management.

2. Overview of the platform

Scribe consists of a web application, secured by a user authentication system [1]. The main purpose of the platform is to facilitate French court decisions datasets annotation process, and communicate rapidly between legal experts and NLP developers.

System specifications. The *Scribe* prototype has been designed to get simple interfaces adapted to French legal judgments annotation tasks. It enables links creation between datasets and annotations in such way that developers can check annotation states and query the database easily. On the other hand, annotators can search for particular court decisions, annotate decisions (alone or with other experts independently on the same set of decisions), save and edit annotations.

The first step of the Scribe pipeline is composed of three tasks: claim categories, named entity recognition (NER), and sectioning the decision (header, facts, etc.). Figure 1 highlights *Scribe* use cases. Two actors can interact with *Scribe*: *Annotators* (legal experts) and *Developers*. Both of them need *admin* authorization to access the platform. For each task (claims categories, NER, and sectioning), *annotators* should be able to create, delete or update items (rows) such as type of jurisdiction and chamber for entities tables (see Figure 1); and attributes (columns) like category name or description. *Annotators* can associate a resizable dataset (can add and remove decisions as they want) to a specific task row (such as type of jurisdiction in NER task) by searching specific court decisions from the database with a dynamic number of keywords either required or excluded, wanted dataset size, and the legal judgment deliver (either Cassation, Appeal and/or First instance court). They can display the content of the decisions resulting from the search, one by one, and then select (deselect) the decisions to include (exclude). Once the decisions choice is complete, they can append them to the selected task row as the associated dataset. If the task row is already associated with a dataset, then the selected decisions are added to the existing one. *Annotators* can autonomously annotate created datasets and multiple annotators may annotate the same dataset and update annotations. They can display dataset decisions one by one in the interface annotation, delete unwanted ones, and select text spans corresponding to the correct annotation from the decision text. Once finished, they can save it, and move to another one. *Developers* may check the progress state of the annotation process continuously and can download datasets at any moment in standardized format on the homepage for easy access). The creation of a new row in a given task is a dataset query, for instance creating the row ”claims of non-recoverable charges, the article 700” in the table claims categories. After that, either *developers* or *annotators* can associate a dataset (decisions list) to this row (search by keywords and by standard jurisdiction). Once done, *annotators* can save annotations of this dataset, and the *developer* can simultaneously follow the annotation progress and download the dataset.

Modules. *Scribe* is composed of four main components, as shown in Figure 2: database component, annotation component, download component, and the user authentication component.

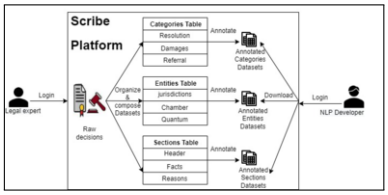


Figure 1. General overview of *Scribe*

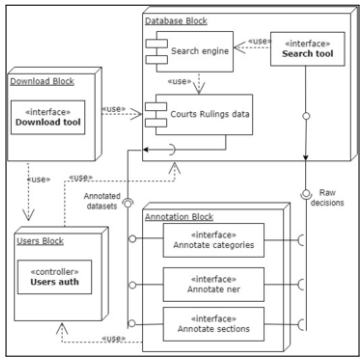


Figure 2. *Scribe* components diagram.

The database component is responsible for saving data (raw decisions, annotations, tasks classes and users) and for the database exploration with a search engine used when composing datasets. *MongoDB* is a Data Base Management System *DBMS* which is a *NoSQL* database that allows the structure of the document to be modified [2], and to handle big data.

The annotation component contains three sub-components, each one dedicated to its specific task (discussed in Section 2) and contains its suitable form. For example, in the NER form, the user selects an entity from the decision text body then all occurrences are automatically selected. In the sectioning form, the user has just to select a decision part that represents the section. As a decision can include multiple claims of the same category, the form of this task allows the *annotator* to create dynamic similar forms. The three sub-components consume dataset raw decisions (obtained from the search step in the database component) and provide their annotations (also stored in the database). *Django* is used to implement the backend logic and *Reactjs* for the frontend. *The download component* enables developers to check the available datasets and the annotation progress state (both raw and annotated datasets may be downloaded).

The authentication component manages user accounts and login, only authorized people can enter the system and read data. For more explanations, see the demo `https://lawbot.unimes.fr/annotateur`.

Data. A record in *MongoDB* is a document, which is a data structure composed of field and value pairs, documents are similar to JSON objects [2]. *MongoDB* simplifies annotation saving process and getting datasets in a standard format. It is tolerant about document keys, we can omit some in the documents and place them in others. Figure 3 represents the diagram of the database. There is a collection of system tasks, each task can have multiple sub-tasks (a dataset may be associated with each sub-task). Finally, from raw decisions, *annotators* save their annotations (dataset creation), accordingly each decision of an annotated dataset is associated with an annotation object that contains the annotations array, all elements of this array follow this format: text (annotated span of text), label (class of the text), start position, and end position.

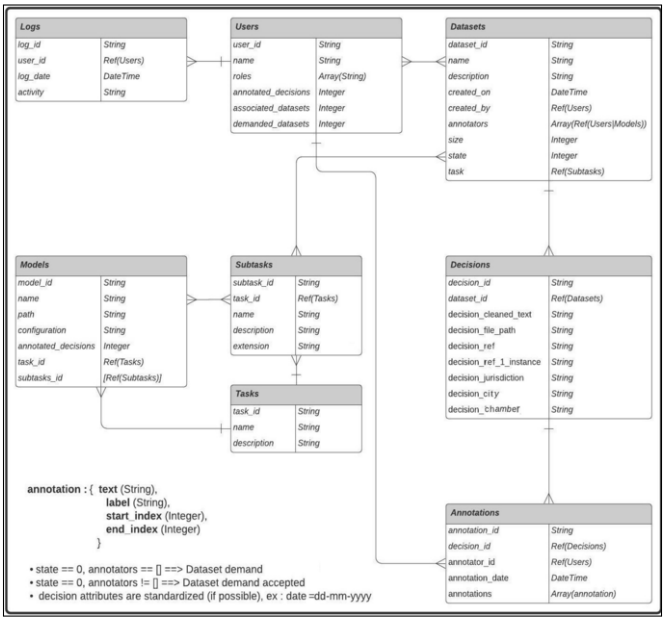


Figure 3. *Scribe* data architecture.

3. Conclusion

In this paper, a collaborative annotation web application has been proposed. This tool is dedicated to the management of court decisions for teams of legal NLP annotators and developers. As a future improvement, it should allow deploying trained models [5,8] and automating annotations.

References

[1] Barkadehi M H, Nilashi M, Ibrahim O, Fardi A Z, Samad S. Authentication systems: A literature review and classification. Telematics and Informatics. 2018;35(5), 1491-1511.

[2] Chauhan A. A Review on Various Aspects of MongoDb Databases. International Journal of Engineering Research & Technology (IJERT). 2019;8(5).

[3] Hammami E, Faiz R, Akermi I, Rosenthal-Sabroux C, Gargouri F, Arduin PE. A Dynamic Convolutional Neural Network Approach for Legal Text Classification, *Information and Knowledge Systems. Digital Technologies, Artificial Intelligence and Decision Making*. 2021, Springer, p. 71-84.

[4] Libal T. A Meta-level Annotation Language for Legal Texts. In Dastani, M., Dong, H. and van der Torre, L., *Logic and Argumentation*, Springer International Publishing. 2020;p. 131-150.

[5] Mahmoudi S, Condevaux C, Mathis B, Zambrano G, Mussard S. NER sur décisions judiciaires françaises: CamemBERT Judiciaire ou méthode ensembliste?. 2022 Jan. In Extraction et Gestion des connaissances EGC'2022.

[6] Tamper M, Oksanen A, Tuominen J, Hietanen A, Hyvönen E, *Automatic Annotation Service APPI : Named Entity Linking in Legal Domain*. 2020; Extended Semantic Web Conference.

[7] Wyner AZ, Katz WP. A case study on legal case annotation, *Frontiers in Artificial Intelligence and Applications*.2013 Jan, 259:165-174.

[8] Zhong H, Xiao C, Tu C, Zhang T, Liu Z, Sun M. How does NLP benefit legal system: A summary of legal artificial intelligence, *arXiv preprint*, arXiv:2004;12158, 2020.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220484

An Interactive Natural Language Interface for PROLEG

Ha-Thanh Nguyen [a] Fumihito Nishino [a] Megumi Fujita [a] Ken Satoh [a]
[a] *National Institute of Informatics, Japan*
nguyenhathanh@nii.ac.jp

Abstract. PROLEG is a famous computer program supporting attorneys in the legal inference process. However, the input of this system is expressed in Prolog, which most lawyers are not familiar with. This technical barrier is a serious problem for using PROLEG in the real legal context. A natural language interface is one of the solutions to this problem. We have developed a prototype of such an interface. This paper describes the prototype and its current performance. The prototype translates input facts into Prolog expressions following PROLEG syntax. The system consists of three main modules, (1) natural language perceiver, (2) PROLEG reasoner, and (3) inference explainer. In addition, we analyze the performance of the prototype and identify existing issues and discuss possible solutions.

Keywords. interactive interface, natural language, PROLEG

1. Introduction

Legal inference is an important task for lawyers. The legal inference process consists of two main phases: collection of the facts and inference from the facts. In order to reach a legal goal, for example exculpation, liability proving, or sentencing, the attorney collects facts and infers the legal conclusions from the facts. However, legal inference is a complex process and attorneys can make errors in the legal inference process. There are several reasons for this. First, the legal inference process has many steps, and it is difficult to check all steps. Second, there are many rules of law, and it is difficult to remember all rules of law. Third, the legal inference process is time-consuming and attorneys often have to make decisions under time pressure.

PROLEG [1,2] is a legal reasoning system based on the Japanese "theory of presupposed ultimate facts" (called "Yoken-jijitsu-ron" in Japanese, the JUF theory [3], in short). Incomplete information is a common issue of many legal cases. In practice, judges usually have to make a decision even if they do not have all information they need. In such a situation, the JUF theory is used for decision-making. The presupposition of ultimate facts is the key idea of the theory. An ultimate fact is a fact that cannot be further explained or justified. These facts are not necessarily true, but they are assumed to be true for the sake of argument. For example, in a criminal case, the defendant is assumed to be innocent until proven guilty.

PROLEG was proven to be useful in actual legal cases. However, the input of this system written in logic programming can be only in the form of logical formulas. This

Figure 1. General architecture of the system.

creates a technical barrier for attorneys and judges who are not familiar with logic. In order to make PROLEG more user-friendly, we propose an interactive inference system allowing users to input natural language. The system contains three modules that receive fact descriptions in natural language, convert them into PROLEG formula, operate the logical reasoning process and return the output along with its visual explanation.

2. System Description

2.1. General Architecture

The general architecture of the system is described in Figure 1. There are three modules which are (1) Natural language perceiver, (2) PROLEG Reasoner, and (3) Inference Explainer.

The natural language perceiver is used for processing the fact description in natural language, extracting the important information, and constructing the corresponding PROLEG formulas. This module is implemented using a combination of a translation deep learning model and supportive heuristic rules. We use BART [4] for the translation model with different strategies:

- **naive end-2-end translation:** we simply train a BART model with our PROLEG pair samples without any further tweaking.
- **translation and correction:** embedded behind the translation model in the framework, the correction model is trained to map the errors back to the originals. These errors are generated by random heuristic rules (such as capitalization, splitting, removing, adding, replacing).
- **argument recognition**: we train a BART model to recognize the arguments for PROLEG formulas instead of directly translating the natural language into PROLEG formula.

Evaluating the outcomes of the three approaches, it was found that the argument recognition approach had the best performance. The reason is that the model does not need to remember the whole syntax of the logical formula but only to detect the correct arguments. With the current limitation in the number of training samples, the translation model sometimes wrongly detects the date-related arguments, we come up with a workaround to prevent this by regular expressions.

The extracted PROLEG formulas are then used as input to the PROLEG reasoner, which is implemented in Prolog. The PROLEG reasoner contains a manually pre-defined set of PROLEG rules and performs the inference by using these rules. The output of PROLEG reasoner is the result of the inference as well as the intermediate reasoning steps.

The final module is the inference explainer which is used for the explanation of the inference result. The explanation is presented in the form of an interactive graph, which

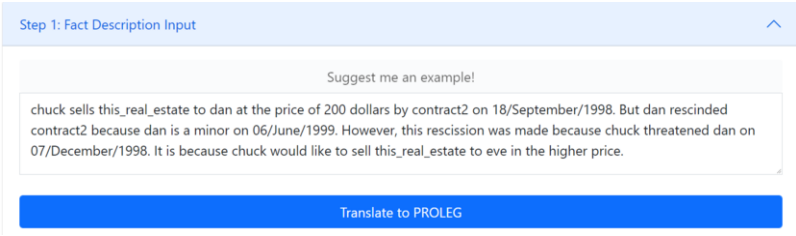


Figure 2. Fact Description Input.

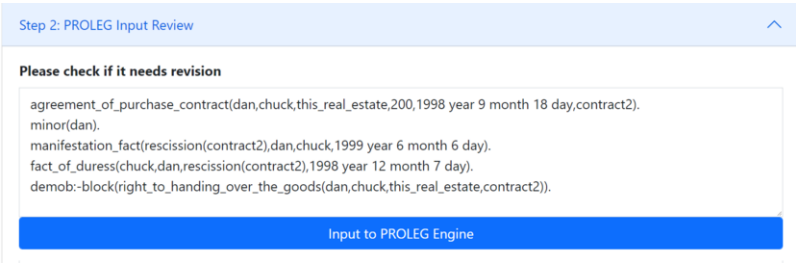


Figure 3. PROLEG Input Review.

the user can use to trace the inference step-by-step and learn the importance of PROLEG rules in each step. This explainer module reduces the complexity of understanding the PROLEG reasoner results and makes it understandable for lay users.

2.2. User Interface

Fact Description Input. The user interface of this step is presented in Figure 2. The user can input a natural language fact in the text area and click on the submit button. The output of the natural language perceiver is PROLEG formulas for the input sentences. For the demonstration purpose, the user can choose to use sentences suggested by the system and modify them if necessary.

PROLEG Input Reivew. The user interface of this step is presented in Figure 3. The translated PROLEG formulas are in the text area, the users can validate the formulas and make modifications if necessary. After the formulas are verified, users can input them into the PROLEG Engine.

PROLEG Inference Graph. The user interface of this step is presented in Figure 4. The PROLEG inference explanation is presented in the form of an interactive graph. Each node in the graph represents a PROLEG rule. If the node condition is satisfied, the node is presented in green, otherwise orange. By clicking on the node, the user can investigate the satisfiability of the rule condition and the reason for such results. This will help users to understand the PROLEG rules and improve their intuition about the underlying reasoning process. In the in-development version, the user can also click on the "step-by-step" button for a detailed explanation of the inference path.

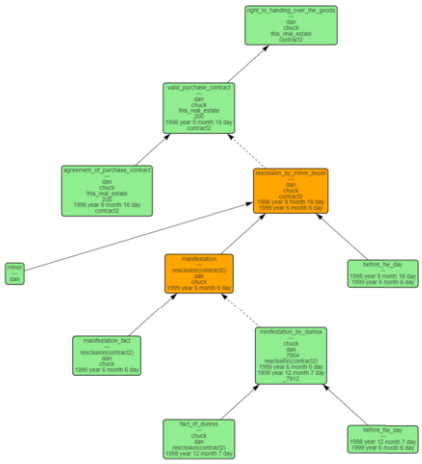


Figure 4. PROLEG Inference Graph.

3. Conclusions

This paper describes the development of an interactive inference system for the PROLEG core language. The system is designed to reduce the burden of writing logical formulas for lay users and provide an interactive explanation of the inference result to help users understand the underlying reasoning process. This system is an important link between the knowledge representation research and the real-life application for lay users, which will help to expand the application of PROLEG in the field of intelligent reasoning. Since the feasibility of the system has been confirmed by design, the next step of this work is to improve the system through data preparation and method enhancement for each module.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103, JP19H05470 and JP22H00543 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4, Japan.

References

[1] Satoh K, Kubota M, Nishigai Y, Takano C. Translating the Japanese Presupposed Ultimate Fact Theory into Logic Programming. In: Proceedings of the 2009 conference on Legal Knowledge and Information Systems: JURIX 2009: The Twenty-Second Annual Conference; 2009. p. 162-71.

[2] Satoh K, Asai K, Kogawa T, Kubota M, Nakamura M, Nishigai Y, et al. PROLEG: an implementation of the presupposed ultimate fact theory of Japanese civil code by PROLOG technology. In: JSAI international symposium on artificial intelligence. Springer; 2010. p. 153-64.

[3] Ito S. Lecture series on ultimate facts. Shojihomu (in Japanese). 2008.

[4] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020. p. 7871-80.

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

doi:10.3233/FAIA220485

Consumer Dispute Resolution System Based on PROLEG

Shidaka NISHIOKA[a,1], Yuto MORI[b] and Ken SATOH[c,1]
[a]*Japan Advanced Institute of Science and Technology, Japan*
[b]*Hitotsubashi University, Japan*
[c]*National Institute of Informatics and Sokendai, Japan*

Abstract. It is challenging for lay consumers to predict a legal conclusion by applying appropriate law in a consumer dispute. We developed a system that assists consumers to predict a possible legal conclusion. The system enables consumers to identify the type of consumer disputes by using a tree structure, and to apply appropriate legal rules implemented as PROLEG programs. We arranged the system to avoid possible inconsistency between the tree structure and the PROLEG program.

Keywords. PROLEG, ISAI PROLEG, PROLEG Menu, Japanese Ultimate Fact theory, Japanese Consumer Law, Consumer Disputes

1. Introduction

In Japan, disputes between consumers and business operators ("Consumer Disputes") are common. Under Japanese consumer law, there are variations in the types of Consumer Disputes defined by statutes, and applicable legal rules differ depending on the type. It can be challenging for consumers with no legal knowledge to determine the types of Consumer Disputes and to correctly select the applicable legal rules.

Japanese ultimate act theory (yoken-jijitsu-ron in Japanese, the "JUF theory") is the legal rules that Japanese courts apply to civil cases. PROLEG is a logic programming language that can implement JUF theory[1]. The interactive system for arranging issues based on PROLEG ("ISAI PROLEG")[2] provides a user interface that allows users to input values representing the facts of cases into variables in PROLEG programs. PROLEG and ISAI-PROLEG can implement the legal rules of the Japanese consumer law in the form of JUF theory. However, for consumers to take advantage of programmed legal rules, they must correctly identify the type of Consumer Disputes and select a PROLEG program implementing the legal rule applicable to the identified type.

We developed Consumer Dispute Resolution System that assists consumers in identifying types of Consumer Disputes and select applicable PROLEG programs.

2. PROLEG Menu

PROLEG Menu is a software that can implement a tree structure with a user interface. Each node of the tree can have a question and multiple answer choices for the question.

[1] Corresponding Authors: Shidaka Nishioka, shidaka.nishioka@jaist.ac.jp; Ken Satoh, ksatoh@nii.ac.jp

At each node, a user can select one of the choices through the user interface. The next node to be reached is determined based on the user's answer. Each leaf node is associated with a PROLEG program in the form of ISAI PROLEG. When the user reaches any of the leaf nodes, PROLEG Menu calls the PROLEG program associated with the leaf node.

3. Description of System

The system comprises two parts: Tree and Logic, as shown in **Figure 1**.

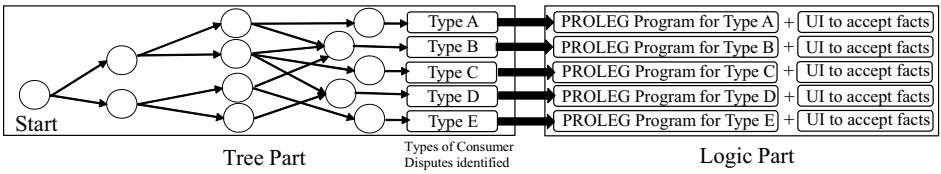


Figure 1. Functions of Tree and Logic Parts.

3.1. Tree Part

The Tree Part enables a user, typically a lay consumer, to identify the type of Consumer Disputes and to select the appropriate PROLEG program. We implemented the Tree Part by using PROLEG Menu such that it has a tree structure. Each node in the tree has a question. The user's answer to the question at a certain node determines which child node the user goes to next. By repeating this process, the user reaches one of the leaf nodes. The questions at each node have been deployed such that the system can identify the type of Consumer Disputes. Therefore, when the user reaches a leaf node, the type of Consumer Dispute is identified. Each leaf node was associated with a PROLEG program which implements the legal rule applicable to the identified type of Consumer Disputes.

3.2. Logic Part

The system transitions to the Logic Part when a PROLEG program is called at the end of the Tree Part. The Logic Part accepts a user's and a business operator's factual assertions and infers the legal conclusion by applying legal rules to those facts. We implemented the Logic Part by using the ISAI PROLEG such that it has a user interface that allows users to input the facts they assert. In the Logic Part, the system requests a consumer and a business operator, the parties to a Consumer Dispute, to input the facts they want to assert in their Consumer Dispute. After the parties complete inputting, the system calculates to reach the legal conclusion inferred by applying the applicable legal rules to the facts input by the parties and displays the results. This enables consumers to predict possible legal conclusions for their Consumer Disputes.

4. Tree Structure for Identifying Types of Consumer Disputes

Generally, each type of Consumer Disputes consists of one or more factors. In the Tree Part, each node has a question that asks to a consumer if there are facts satisfying one of

the factors of Consumer Disputes. To identify the types of Consumer Disputes, we deployed questions about factors common to more types of Consumer Disputes at upper nodes. Questions about factors included only in fewer types of Consumer Disputes are set at lower nodes.

For example, under the Specified Commercial Transactions Act (the "SCTA") of Japan, Consumer Disputes can be classified in three types as shown in the table below:

Table 1. Types of Consumer Disputes and Factors in each type of Consumer Dispute

Types	Consumer	Longer Than 2 Months	More Than 50,000 Yen
1	Y	Y	Y
2	Y	Y	-
3	Y	-	-

In the **Table 1**, "Y" indicates that the type of Consumer Dispute in the same line has that factor. For example, Consumer Dispute Type 2 has Factors *Consumer* and *Longer Than Two Months*. To identify the type of Consumer Dispute, the Tree Part asks the consumer at the initial node if the case has a fact satisfying the Factor *Consumer*, because the Factor *Consumer* is common to all types. If the consumer answers "no," the system can conclude that the SCTA does not apply to the case. If the consumer answers "yes," then the Tree Part asks for the sufficiency of the Factor *Longer Than Two Months*, because it is common to Type 1 and 2. If the consumer's answer is "no," the system can conclude that the case is of Type 3. If the consumer's answer is "yes," the Tree Part further asks at the next node whether the Factor *More Than 50,000 Yen* is satisfied. In this manner, the Tree Part identifies the type of Consumer Dispute.

5. Avoiding Inconsistencies Between Tree and Logic Parts

Previously, PROLEG programs operate independently, not associated with a tree structure. On the other hand, in the system, the PROLEG programs in the Logic Part are associated with the Tree Part where the types of Consumer Disputes are determined by using a tree structure. Since both factors to identify a type of Consumer Disputes and legal requirements of applicable legal rules are defined by the Japanese consumer law, in the system, one of the factors to identify the type of Consumer Disputes appearing as a question at a node in the Tree Part can be identical to one of the legal requirements in the applicable legal rules implemented as a PROLEG program in the Logic Part. Therefore, an inconsistency may occur between the Tree and Logic Parts.

For example, assume that a Consumer Dispute is of Type 1 in the **Table 1**, which has Factors *Consumer*, *Longer Than Two Months* and *More Than 50,000*. Suppose that the applicable legal rule to cancel the contract in Type 1 includes Requirements *Consumer*, *Telling False* and *Belief*. Here, *Consumer* appears as a Factor and a Requirement. Previously, the PROLEG program implementing this legal rule must contain atomic formulae representing all the three Requirements *Consumer*, *Telling False* and *Belief* with the variables representing the presence or absence of facts that satisfy the three Requirements. If the system has this PROLEG program as is in the Logic Part, a user who answered "yes" to the question asking "are you a consumer?" in the Tree Part may input a fact indicating that the user is not a consumer in the Logic Part.

To avoid this inconsistency, we implemented the system such that, if a Factor in the Tree Part and a Requirement in the Logic Part are identical, the PROLEG program in the Logic Part does not contain that Requirement. For example, in the case of Type 1 above, because the Tree Part asks the presence or absence of Factor *Consumer* to identify

the type, the PROLEG program in the Logic Part implementing the applicable legal rule contains only atomic formulae representing Requirements *Telling False* and *Belief* to prevent the user from inputting contradictory facts about Requirement *Consumer*.

Removal of a certain requirement from the PROLEG programs in this manner still maintains the accuracy of the inference for the following reasons. Suppose that, in a certain legal rule, a consumer's Claim X can be established by the presence of Facts *a′*, *b′* and *c′* that satisfy Requirements *a*, *b* and *c,* respectively. This can be expressed in the form of PROLEG as follows:

claim_x(_*a′*, _*b′*, _*c′*) <= requirement_a(_*a′*), requirement_b(_*b′*),
requirement_c(_*c′*).

If the consumer answers in the Tree Part that Factor *a* was satisfied, it means that Fact *a′* exists and the truth value of the atomic formula "requirement_a(_*a′*)" turns out to be true before the PROLEG program in the Logic Part starts. Based on this premise, the following holds:

{claim_x(_*a′*, _*b′*, _*c′*) <= requirement_a(_*a′*), requirement_b(_*b′*),
requirement_c(_*c′*).} ∧ requirement_a(_*a′*) ↔ T
⇔ claim_x(_*b′*, _*c′*) <= requirement_b(_*b′*), requirement_c(_*c′*).

In this manner, the system avoids possible inconsistencies between the Tree and Logic Parts, as shown in **Figure 2**.

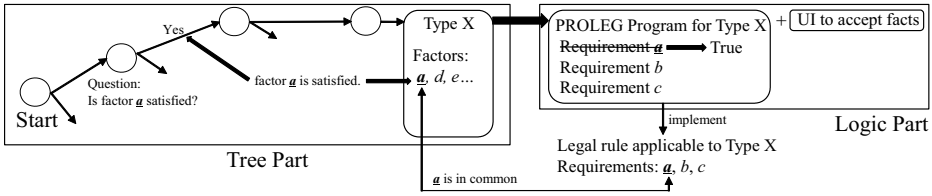


Figure 2. Method to avoid inconsistency between Tree and Logic Parts

6. Conclusion

We demonstrated how the system identifies the types of Consumer Disputes by using a tree structure and avoids possible inconsistency between Tree and Logic Parts. In these manners, the system assists consumers to predict legal conclusions for Consumer Dispute.

This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103 and JP22H00543.

References

[1] Satoh K, et al (2011) Proleg: an implementation of the presupposed ultimate fact theory of Japanese civil code by prolog technology. LNAI6797, pp 153 - 164.
[2] Satoh K, et al (2021) Interactive System for Arranging Issues based on PROLEG in Civil Litigation. Proc. of ICAIL 2021, pp 273 - 274.

This page intentionally left blank

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

Subject Index

Legal Knowledge and Information Systems
E. Francesconi et al. (Eds.)

Author Index