Robust Input Representations for Low-Resource Information Extraction

Dissertation

zur Erlangung des Grades des Doktors der Ingenieurwissenschaften
der Fakultät für Mathematik und Informatik der Universität des Saarlandes

vorgelegt von

Lukas Lange

Saarbrücken
2022

Dean of the Faculty	Univ.-Prof. Dr. Jürgen Steimle Fakultät für Mathematik und Informatik (MI) Universität des Saarlandes
Day of Colloquium	22.07.2022 Universität des Saarlandes Saarbrücken

Examination Committee:

Chairman:	Univ.-Prof. Dr. Josef van Genabith
First Reviewer, Advisor:	Univ.-Prof. Dr. Dietrich Klakow
Second Reviewer:	PD. Dr. Roman Klinger
Academic Assistant: (Akademischer Mitarbeiter)	Dr. Volha Petukhova

Abstract

Artificial intelligence, and in particular, the field of natural language processing, is going through a tremendous transition with the rise of deep learning systems. Motivated by these recent fundamental changes, a wide range of new research questions came up concerning the stability of such large-scale systems and their applicability beyond well-studied tasks and datasets, such as information extraction in non-standard domains and languages. One important aspect is the usage of deep networks in low-resource environments. Neural models are known for requiring large amounts of training data because millions of parameters have to be tuned. This includes the training in non-standard languages, but also text domains and tasks, for which — even in English — only little training data is available. Therefore, a core challenge of deep learning methods targeting low-resource information extraction involves overcoming the resource limitations in the training process. Recent advances in this field are achieved by pre-training representations on large-scale corpora to capture generally applicable knowledge. However, while performing great in standard applications, these general models lack the specific knowledge of specialized domains for which fewer training resources exist. Instead, relevant domain knowledge can be incorporated into general-domain models in order to improve performance in non-standard domains. Moreover, the transfer of pre-trained representations across languages offers great opportunities but also even greater risks, as differences between the languages and their representations inside multilingual models often negatively influence the cross-lingual transfer and outweigh the positive transfer effects.

In this work, we address the previously described challenges for information extraction in non-standard domains and languages and propose novel model architectures and training strategies to overcome the existing limitations. In particular, we propose solutions to close the domain gap between representation models and address domain-specific challenges, e.g., anonymization in pipeline models for the clinical domain. Moreover, we explore cross-task transfer with prediction methods to select suitable transfer sources and perform cross-language transfer with our innovative models for multilingual temporal tagging. Our main contributions are as follows:

(1) We show that word representations trained on texts from the general domain can be greatly improved in non-standard domains by incorporating domain-specific knowledge from the target domain by either fine-tuning languages models on documents from the target domain or by adding domain-specific word representations. For this, we propose a novel meta-embedding architecture to create a joint representation of multiple embeddings from various domains that captures the diverse knowledge contained in the embeddings. We demonstrate the effectiveness of our approach on a variety of sentence classification and sequence-tagging tasks across languages and domains. Our analysis shows that our approaches are particularly successful in low-resource settings.

(2) We explore cross-domain, -task, and -language transfer. We propose a new similarity measure for datasets based on task-specific features and properties of deep learning models and select suitable sets of transfer sources using dynamic prediction methods. Our

new model similarity measure based on feature mappings outperforms currently used similarity measures as it is able to capture both task and domain similarity at the same time. Our dynamic selection method for sets of sources outperforms the single-source transfer — as suggested in prior work — and effectively avoids negative transfer.

(3) We study natural language processing pipelines consisting of multiple steps, such as our 3-step temporal tagging pipeline for the extraction, normalization, and anchoring of temporal expressions. Specifically, we introduce the first neural method for normalizing temporal expressions based on masked language modeling. Our experiments in 17 languages demonstrate the robust performance of our method across languages. We further study cross-language transfer in the context of temporal tagging and explore the prospects of embedding alignments for multilingual models. In particular, we set the new state of the art in low-resource languages. Moreover, we study an NLP pipeline for anonymization and concept extraction for processing clinical documents. We propose a differentiable version of an NLP for anonymization and clinical concept extraction show that anonymization positively influences the concept extraction performance.

Kurzzusammenfassung

Künstliche Intelligenz und insbesondere der speziellere Bereich der Verarbeitung natürlicher Sprache erfährt mit dem Aufkommen von Deep-Learning-Systemen einen enormen Wandel. Motiviert durch diese jüngsten fundamentalen Veränderungen, ist eine Vielzahl neuer Forschungsfragen aufgekommen, die sich mit der Stabilität solcher großen Systeme und ihrer Anwendbarkeit jenseits gut untersuchter Aufgaben und Datensätze befassen, z.B. zur Informationsextraktion in unüblichen Domänen und Sprachen. Ein wichtiger Aspekt ist der Einsatz von tiefen Netzen in ressourcenarmen Umgebungen. Neuronale Modelle sind dafür bekannt, dass sie große Mengen an Trainingsdaten benötigen, da Millionen von Parametern trainiert werden müssen. Dazu gehört das Training in Nicht-Standardsprachen, aber auch Textdomänen und Aufgaben, für die — selbst im Englischen — nur wenige Trainingsdaten zur Verfügung stehen. Eine zentrale Herausforderung von Deep-Learning-Methoden für die Informationsextraktion mit limitierten Ressourcen ist daher die Überwindung der Ressourcenbeschränkungen im Trainingsprozess. Jüngste Fortschritte in diesem Bereich wurden mit dem Vortrainieren von Repräsentationen auf großen Korpora erzielt, um allgemein anwendbares Wissen zu erlernen. Diese allgemeinen Modelle sind zwar für Standardanwendungen gut geeignet, verfügen jedoch nicht über das spezifische Wissen spezieller Bereiche, für die weniger Trainingsressourcen zur Verfügung stehen. Stattdessen kann relevantes spezifischeres Domänenwissen in allgemeine Modelle integriert werden, um die Leistung in unüblichen Domänen zu verbessern. Darüber hinaus bietet der sprachübergreifende Transfer von vortrainierten Repräsentationen große Chancen, aber auch noch größere Risiken, da Unterschiede zwischen den Sprachen und ihren Repräsentationen in mehrsprachigen Modellen den sprachübergreifenden Transfer oft negativ beeinflussen und die positiven Transfereffekte überwiegen.

In dieser Arbeit befassen wir uns mit den zuvor beschriebenen Herausforderungen bei der Informationsextraktion in unüblichen Domänen und Sprachen und schlagen neuartige Modellarchitekturen und Trainingsstrategien vor, um die bestehenden Einschränkungen zu überwinden. Insbesondere schlagen wir Lösungen vor, um die sogenannte Domänenlücke zwischen Repräsentationsmodellen zu schließen und domänenspezifische Herausforderungen anzugehen, z.B. die Anonymisierung in Pipeline-Modellen für die klinische Domäne. Darüber hinaus erforschen wir den aufgabenübergreifenden Transfer mit Vorhersagemethoden zur Auswahl geeigneter Transferquellen und führen den sprachübergreifenden Transfer mit unseren innovativen Modellen für die mehrsprachige Extraktion und Normalisierung von temporalen Ausdrücken durch. Unsere Hauptbeiträge sind die Folgenden:

(1) Wir zeigen, dass Wortrepräsentationen, die auf Texten aus der allgemeinen Domäne trainiert wurden, in Nicht-Standard-Domänen erheblich verbessert werden können, indem domänenspezifisches Wissen aus der Zieldomäne entweder durch das Vortrainieren von Sprachmodellen auf Dokumenten aus der Zieldomäne oder durch Hinzufügen von domänenspezifischen Wortrepräsentationen einbezogen wird. Zu diesem Zweck schlagen wir eine neuartige Meta-Embedding-Architektur vor, um eine gemeinsame Darstellung mehrerer Repräsentationsmodelle aus verschiedenen Domänen zu erstellen, die das in

den Repräsentationen enthaltene vielfältige Wissen erfasst. Wir demonstrieren die Effektivität unseres Ansatzes bei einer Vielzahl von Klassifizierungsaufgaben in verschiedenen Sprachen und Domänen. Unsere Analyse zeigt, dass unser Ansatz besonders in ressourcenarmen Umgebungen erfolgreich ist.

(2) Wir untersuchen den domänen-, aufgaben- und sprachenübergreifenden Transfer. Wir schlagen ein neues Ähnlichkeitsmaß für Datensätze vor, das auf aufgabenspezifischen Merkmalen und Eigenschaften von Deep-Learning-Modellen basiert, und wählen mithilfe dynamischer Vorhersagemethoden geeignete Sammlungen von Transferquellen aus. Unser neues Modellähnlichkeitsmaß, basierend auf Repräsentationsmodellen, übertrifft die derzeit verwendeten Ähnlichkeitsmaße, da es in der Lage ist, sowohl Aufgaben- als auch Domänenähnlichkeit gleichzeitig zu erfassen. Unsere dynamischen Auswahlmethode für die Bestimmung einer Vielzahl von geeigneten Trainingsdatensätzen übertrifft den in früheren Arbeiten vorgeschlagenen Transfer einzelner Datensätze, und kann effektiv negativen Transfer vermeiden.

(3) Wir untersuchen mehrstufige Pipelines für die Verarbeitung natürlicher Sprache, wie z.B. unsere dreistufige Pipeline für die Extraktion, Normalisierung und Verankerung von zeitlichen Ausdrücken. Insbesondere stellen wir die erste neuronale Methode zur Normalisierung temporaler Ausdrücke vor, die auf maskierter Sprachmodellierung basiert. Unsere Experimente in 17 Sprachen zeigen die robuste Leistung unserer Methode in verschiedenen Sprachen. Darüber hinaus untersuchen wir den sprachübergreifenden Transfer im Kontext der Extraktion und Normalisierung zeitlicher Ausdrücke und erforschen Optimierungsmethoden für die Repräsentationen in mehrsprachigen Modellen. Insbesondere sind unsere Modelle der neue Stand der Technik in ressourcenarmen Sprachen. Darüber hinaus untersuchen wir eine NLP-Pipeline zur gemeinsamen Anonymisierung und Konzeptextraktion für die Verarbeitung klinischer Dokumente, die wir mit Multi-Task-Training vergleichen. Wir schlagen eine differenzierbare Version dieser Anonymisierungspipeline vor und untersuchen die Auswirkungen der Anonymisierung auf die Konzeptextraktion und zeigen, dass diese die Leistung der Konzeptextraktion positiv beeinflusst.

Acknowledgment

I had the great pleasure to pursue my PhD studies at Saarland University and the Bosch Center for Artificial Intelligence. Many amazing people from both institutions supported during this time. First of all, I would like to express my deepest gratitude towards my supervising professor Dietrich Klakow. He encouraged me to dive into the field of deep learning and sparked my interest in academic research. I am very thankful for his constructive feedback and invaluable advice. I am equally grateful to Jannik Strötgen and Heike Adel, my fantastic advisors at Bosch. Both managed to create a working environment where I could focus on my research topics. I immensely enjoyed our common brainstorming and discussion sessions. Thank you all for all the time and effort you invested in the last years to make my PhD studies become a success.

Special thanks to Annemarie Friedrich for providing incredibly useful feedback on paper drafts and presentations. I have learned a lot from you. Then, I would like to thank the PhD students at Bosch who I had to pleasure to get to know and work with: Hendrik Schuff, Subhash Chandra Pujari, Stefan Grünewald, Sophie Henning, Youmna Ismaeil, Riccardo Cipolletti, Elena Wege, and Fynn Hellweg. Thank you for many great discussions on research topics and even more nice off-topics chats and meetings. Good luck during your own remaining PhD time. Thanks to all BCAI-R26 team members: Dragan Milchevski, Daria Stepanova, Trung-Kien Tran, Mohamed Gad-Elrab, and Evgeny Kharlamov. You helped me to broaden my research scope and look at my problems from different perspectives.

I would also like to thank Michael A. Hedderich and all other members from the LSV group of Saarland University, including but not limited to Dana Ruiter, David Adelani, Dawei Zhu, Marius Mosbach, Thomas Kleinbauer and Alexander Blatt. It was great to work with Michael on low-resource NLP topics and discuss my research in this round. Many thanks to Xiang Dai for working with him during his sabbatical at BCAI.

In addition, I would like to express my gratitude to more people who helped me during my PhD time and made my life easier. Special thanks to Andrian Hanussek, who provided excellent mentorship. Axel Grzesik, Christian Höppler, and Franz Grzeschniok supported me in various company-related processes. Michelle Carnell and Susanne Vohl from the graduate school office at Saarland University helped me a lot with administrative issues.

I would like to thank my parents, Marita and Jürgen, and siblings, Leon und Lorena, and my friends. In particular, thanks to Friederike Bruns for proofreading a pre-final version of this document on short notice. Finally, I want to thank Corinna Tasch for being encouraging and patient with me during the last years. Your support means everything to me!
— Thank you!

Contents

Chapter 1

Introduction

Artificial intelligence, and in particular, the field of natural language processing (NLP), is going through a tremendous transition with the rise of deep learning systems (Otter et al., 2021). The vast majority of NLP tasks for which rule-based or statistical systems were standard approaches for decades are now being almost exclusively solved using deep neural networks. Motivated by these recent fundamental changes, a wide range of new research questions came up concerning the stability of such large-scale systems and their applicability beyond the well-studied tasks and datasets, such information extraction in non-standard domains and languages.

One important aspect of these questions concerns the usage of deep networks in low-resource environments. Neural models are known for requiring large amounts of training data because millions of parameters have to be tuned (Raffel et al., 2020). On the one hand, high-quality NLP resources are typically only created for a small number of languages, with a special focus on English (Bender, 2019). Thus, thousands of other languages are not covered by most NLP methods and models. On the other hand, tackling low-resource settings is even crucial when dealing with popular NLP languages as low-resource settings do not only concern languages but also non-standard text domains and tasks, for which — even in English — only little training data is available. For example, most of today's research focuses on processing news articles or Wikipedia pages in a small set of high-resource languages, which usually limits the applicability to new languages and domains (Ruder, 2019a; Ramponi and Plank, 2020).

Therefore, a core challenge of deep learning NLP methods targeting non-standard text domains and languages involves overcoming the resource limitations in the training process. Recent advances in this field are achieved by pre-training representation models on large-scale corpora to capture generally applicable knowledge (Devlin et al., 2019; Brown et al., 2020). However, while performing great in standard applications, these general models lack the specific knowledge of specialized domains for which fewer training resources exist (Gururangan et al., 2020). For example, the style and vocabulary of specialized text domains can differ tremendously from standard texts (Ben-David et al., 2006), such as

mathematical equations and symbols in scientific publications or various technical terms in engineering domains (Beltagy et al., 2019). For this, general-domain representation models have been shown to not capture this type of information well enough in practice (Lee et al., 2020). In contrast, smaller domain-specific models can reflect domain properties but miss broader and generalizable knowledge. While possible in theory, training a domain-specific model in a similar large-scale pre-training is often not possible due to lack of resources on the one hand and the expensive computation costs on the other hand.

Instead, relevant domain knowledge can be incorporated into general-domain models in order to improve performance in non-standard domains (Rocktäschel et al., 2015). This can be achieved by combining representations from the general domain and domain-specific variants (Kiela et al., 2018) or by transferring information from related resources (Daumé III, 2007). However, selecting suitable transfer sources is challenging and demands various considerations, for example, regarding the compatibility of the task, domain, and language between transfer source and target (Bingel and Søgaard, 2017; Vu et al., 2020). Moreover, the transfer of representation models across languages offers great opportunities but also even greater risks. While reusing data from a high-resource language can help overcome the lack of data in a low-resource language, differences between the languages and their representations inside multilingual models often negatively influence the cross-lingual transfer and outweigh the positive transfer effects (Cao et al., 2020b).

In this work, we address the previously described challenges for information extraction in non-standard domains and languages and propose novel model architectures and training strategies to overcome the existing limitations by creating domain and language-robust input representations. Our main contributions are the following:

(1) We show that language models and word embeddings trained on texts from the general domain can be greatly improved in non-standard domains by incorporating domain-specific knowledge from the target domain. More precisely, we explore two methods for this integration. First, we show that fine-tuning languages models on documents from the target domain increases performance remarkably in the context of clinical information extraction (Chapter 4). Second, we demonstrate that domain knowledge can also be included by adding domain-specific word representations instead of adapting the general domain models (Chapter 3). For this, we propose a novel attention-based meta-embedding method that is optimized with adversarial training to create a joint representation of multiple embeddings from various domains that captures the diverse knowledge contained in the embeddings (Chapter 5).

(2) To further improve our models, we investigate cross-domain, -task, and -language transfer and explore prediction methods to select suitable transfer sources based on domain- and task-similarity measures, as well as language-specific features (Chapter 6). For this, we propose a new similarity measure based on task-specific features and properties of neural NLP models and demonstrate its efficiency, in particular, for transfer across tasks and domains. Moreover, we propose prediction methods that do not only predict the most similar

source — as done in related work — but also compute an actual set of transfer sources that should be included in the training process (Chapter 7).

(3) We investigate NLP pipelines consisting of multiple steps, such as our 3-step temporal tagging pipeline for the extraction, normalization, and anchoring of temporal expressions, as well as a pipeline for anonymization and concept extraction for processing clinical documents. In particular, we investigate the effects of the latter pipeline model in comparison to multi-task training and propose a differentiable version of the anonymization pipeline for joint de-identification and concept extraction that restricts access to privacy-related information (Chapter 3).

(4) We propose the first neural normalization method for temporal expressions by using masked language modeling and context-independent representations that work robustly across languages. For this, we study cross-language transfer in the context of temporal tagging by training a single multilingual model and applying it to many languages that were not observed during training. Further, we explore the prospects of improving our multilingual models by creating a common multilingual embedding space inside the models via embedding alignment methods (Chapter 8).

To conclude, our main contributions are robust model architectures and novel training processes for NLP in non-standard domains and languages. We propose solutions to close the domain gap between representation models and address domain-specific challenges, e.g., anonymization in pipeline models for the clinical domain. Moreover, we explore cross-task transfer with prediction methods to select transfer sources and perform cross-language transfer with our innovative models for multilingual temporal tagging.

The remainder of this thesis is structured as follows: Chapter 2 introduces various information extraction tasks that are considered in this work, as well as background information on relevant model architectures for sequential data processing, pre-trained input representations, and multi-task training methods. Our contributions are described in the Chapters 3 to 8. We start with anonymization and clinical concept extraction models as described in Chapter 3 and analyze the effects of anonymization in real-world information extraction pipelines. In Chapter 4, we further explore concept extraction models in the clinical domain and study domain-specific input representations in Spanish and English and describe our newly introduced *CLIN-X* language models. Chapter 5 is concerned with the combination of general-domain and domain-specific embeddings and introduces our feature-based meta-embeddings trained with an adversarial discriminator. Based on these methods, Chapter 6 studies the inclusion of auxiliary embeddings from different languages in meta-embedding models to improve performance in monolingual applications via knowledge transfer across languages. Chapter 7 is concerned with transfer learning across different tasks and domains and describes methods for the prediction of transfer sources in non-standard domains and

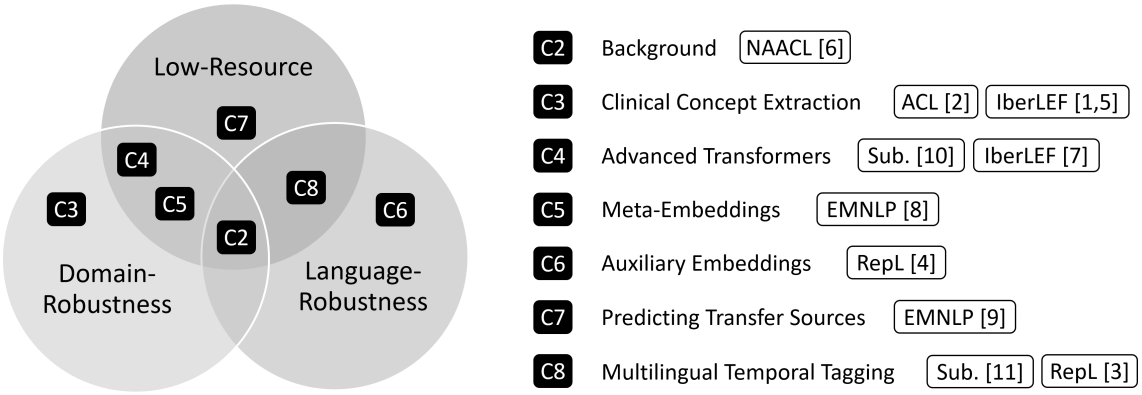


Figure 1.1: Illustration of the topics and chapters discussed in this thesis (left) and the mapping of chapters to our academic publications (right).

low-resource scenarios. For this, we propose a new domain similarity measure and predictors for sets of transfer sources as multiple sources can be beneficial in transfer scenarios, which is often neglected in current research. In Chapter 8 we study the transfer across languages in the context of multilingual temporal tagging, and we describe our multilingual models that process many different languages at once and propose new methods for the normalization of temporal expression and the alignment of languages inside these models. At the end of the thesis, a summary and an outlook are given in Chapter 9. This structure is visualized in Figure 1.1 We now list our academic publications that were created and published during the dissertation process.

Publications and Co-Authoring

The research described in this dissertation is published in multiple academic papers. This research was carried out entirely by myself, and I worked on every aspect of the publications related to this thesis, including designing the methods, conducting the experiments, and writing the papers. Jannik and Heike usually acted as advisors by discussing ideas and improving the drafts. Dietrich joined the discussions and provided feedback. The contributions of other authors are detailed separately below. The work in this dissertation primarily relates to the following peer-reviewed articles (in order of publication):

[1] **L. Lange**, H. Adel and J. Strötgen (2019). NLNDE: The Neither-Language-Nor-Domain-Experts' Way of Spanish Medical Document De-Identification. In *Proceedings of The Iberian Languages Evaluation Forum (IberLEF).*
(Best system of the MEDDOCAN shared task.)

This work will be partly discussed in Chapter 3.

[2] **L. Lange**, H. Adel and J. Strötgen (2020). Closing the Gap: Joint De-Identification and Concept Extraction in the Clinical Domain. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).*

The details of this work will be discussed in Chapter 3.

[3] **L. Lange**, H. Adel and J. Strötgen (2020). On the Choice of Auxiliary Languages for Improved Sequence Tagging. In *Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP@ACL)*.

The details of this work will be discussed in Chapter 6.

[4] **L. Lange**, A. Iurshina, H. Adel and J. Strötgen (2020). Adversarial Alignment of Multilingual Models for Extracting Temporal Expressions from Text. In *Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP@ACL)*.

This work will be partly discussed in Chapter 8. This research was conducted during Anastasiia's internship at BCAI. Anastasiia performed the data pre-processing and carried out preliminary experiments with BERT for the extraction. I trained the final models with a different architecture and implemented the alignment methods.

[5] **L. Lange**, X. Dai, H. Adel and J. Strötgen (2020). NLNDE at CANTEMIST: Neural Sequence Labeling and Parsing Approaches for Clinical Concept Extraction. In *Proceedings of The Iberian Languages Evaluation Forum (IberLEF)*.

This work will be partly discussed in Chapter 3. This research was conducted during Dai's sabbatical at BCAI. We jointly worked on the named entity recognition models. Dai experimented with domain adaptation of language models, and I studied biaffine classifiers. The normalization and ICD coding methods were developed by myself.

[6] M. Hedderich*, **L. Lange***, H. Adel, J. Strötgen and D. Klakow (2021). A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

This survey paper was written in a joint effort with Michael. Michael has mainly written the second part of Section 1 and Sections 3.1, 3.2, 4, and 6. I have mainly written the first part of Section 1 and Sections 2, 3.0, and 5. In this work, only parts of Section 5 of the survey paper are discussed in Section 2.3 of this dissertation.

[7] **L. Lange**, H. Adel and J. Strötgen (2021). Boosting Transformers for Job Expression Extraction and Classification in a Low-Resource Setting. In *Proceedings of The Iberian Languages Evaluation Forum (IberLEF)*.
(Best system of the MEDDOPROF shared task.)

This work will be partly discussed in Chapter 4.

[8] **L. Lange**, H. Adel, J. Strötgen and D. Klakow (2021). FAME: Feature-Based Adversarial Meta-Embeddings for Robust Input Representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

The details of this work will be discussed in Chapter 5.

*Both authors contributed equally.

[9] **L. Lange**, J. Strötgen, H. Adel and D. Klakow (2021). To Share or not to Share: Predicting Sets of Sources for Model Transfer Learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

The details of this work will be discussed in Chapter 7.

[10] **L. Lange**, H. Adel, J. Strötgen and D. Klakow (2022). *CLIN-X*: Pre-trained Language Models and a Study on Cross-Task Transfer for Concept Extraction in the Clinical Domain. In *Oxford Bioinformatics*.

The details of this work will be discussed in Chapter 4.

The following works currently under submission are also discussed:

[11] **L. Lange**, J. Strötgen, H. Adel and D. Klakow (2022). Multilingual Normalization of Temporal Expressions with Masked Language Models. Submitted to the ACL Rolling Review.

The details of this work will be discussed in Chapter 8.

The following articles were published during the dissertation phase. All of them are related to low-resource information extraction but are beyond the scope of this thesis:

[12] **L. Lange**, M. Hedderich and D. Klakow (2019). Feature-Dependent Confusion Matrices for Low-Resource NER Labeling with Noisy Labels. 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)*.

[13] **L. Lange**, H. Adel and J. Strötgen (2019). NLNDE: Enhancing Neural Sequence Taggers with Attention and Noisy Channel for Robust Pharmacological Entity Detection. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks (BioNLP-OST@EMNLP-IJCNLP)*.

[14] A. Friedrich, H. Adel, F. Tomazic, J. Hingerl, R. Benteau, A. Marusczyk and **L. Lange** (2020). The SOFC-Exp Corpus and Neural Approaches to Information Extraction in the Materials Science Domain. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

[15] M. Hedderich, **L. Lange** and D. Klakow (2021). ANEA: Distant Supervision for Low-Resource Named Entity Recognition. In *Practical ML for Developing Countries Workshop (PML4DC@ICLR)*.

Published Code

We published the source code for many of our works publicly available on GitHub under an open-source software license. This includes the code for fine-tuning our concept extraction architecture for clinical concept extraction (see Chapter 4), our adversarial meta-embeddings (see Chapter 5), and our transfer models and prediction methods (see Chapter 7). In addition, the new domain- and language adapted *CLIN-X* language models (LMs) are publicly available on the HuggingFace model hub. In this section, we provide a list of pointers to our published resources:[1]

- Code for fine-tuning models for clinical information extraction (see Chapter 4):
  URL: `https://github.com/boschresearch/clin_x`
  URL: `https://github.com/boschresearch/nlnde_meddoprof`
- *CLIN-X*$_{EN}$ LM pre-trained on English clinical documents (see Chapter 4):
  URL: `https://huggingface.co/llange/xlm-roberta-large-english-clinical`
- *CLIN-X*$_{ES}$ LM pre-trained on Spanish clinical documents (see Chapter 4):
  URL: `https://huggingface.co/llange/xlm-roberta-large-spanish-clinical`
- XLM-R$_{ES}$ LM pre-trained on Spanish general-domain documents (see Chapter 4):
  URL: `https://huggingface.co/llange/xlm-roberta-large-spanish`
- Code for our meta-embeddings and the adversarial training (see Chapter 5):
  URL: `https://github.com/boschresearch/adversarial_meta_embeddings`
- Code for our models and predictors for our transfer experiments (see Chapter 7):
  URL: `https://github.com/boschresearch/predicting_sets_of_sources`

[1] These links were last accessed on March 18, 2022.

Chapter 2

Background

The scope of this thesis is automatic information extraction, a subarea of natural language processing, and contributes new input representation methods and model architectures to solve challenging information extraction problems in non-standard text domains and languages.

Therefore, Section 2.1 of this chapter will provide a general overview of the different information extraction tasks addressed in this thesis and suitable evaluation measures. A detailed overview of the machine-learning architectures required for the other chapters is given in Section 2.2. More information on word embeddings and other input representation methods is provided in Section 2.3. Finally, a description of different multi-task learning procedures is given in Section 2.4.

2.1 Information Extraction from Text

According to Jiang (2012), "[t]he general goal of information extraction is to discover structured information from unstructured or semi-structured texts" (Jiang, 2012, p. 1).

In the following subsections, various information extraction tasks covered in this work are described in more detail. The first part will focus on different sequence-tagging tasks for the extraction of information from texts. The second part concerns the normalization of extracted expression. Finally, the third part will describe the evaluation metrics used to measure the performance of our models for the different tasks.

2.1.1 Sequence Labeling

The most important category of tasks for this thesis are sequence-labeling problems, also called sequence tagging. There exist many sequence-labeling tasks, but all of them are modeled in a similar way. For this, the input is a sequence of words X, also called tokens. The output sequence Y is a sequence of labels, also called annotations, with $y_i \in Y$ being

the label for token $x_i \in X$, i.e., there exists exactly one label per token. Note that there also exists a wide range of work on nested annotations that relax this assumption (Dai, 2018) which is beyond the scope of this work.

In general, these sequence-labeling tasks are performed on the so-called token level, as one label has to be predicted per token. This is in contrast to sentence-level tasks like text classification, where one label has to be assigned to the complete input sequence. For sequence tagging, the model gets the sequence X as input and has to predict the label sequence Y also for unseen input instances. In this work, we use machine-learning models that are well-known for their generalization abilities to unseen instances by learning a mapping $f(X) \rightarrow Y$. More on this will be described in Section 2.2.

Note that one sequence X can have multiple layers of annotations Y_n, i.e., there exist multiple sequence-labeling tasks for the same sentence. We will provide an overview of the most important tasks for this thesis, namely part-of-speech tagging, named entity recognition, and further variants, in the following paragraphs.

Part-of-Speech Tagging. Our first sequence-labeling task is part-of-speech (POS) tagging. Here, the tags are word categories defined in a tagset, such as noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection, numeral, article, or determiner. A more detailed overview of this task is given by Manning and Schütze (1999) in Chapter 10 of their book. In our experiments, we use the multilingual universal dependencies corpora (Nivre et al., 2020), which contain unified part-of-speech tags in many languages. In practice, part-of-speech tagging is often used as a pre-processing step for other applications. For example, named entities are most often nouns, and this knowledge can be integrated as features into a model for named entity recognition.

Named Entity Recognition. As stated by Jiang (2012), "The task of named entity recognition [...] is to identify named entities from free-form text and to classify them into a set of predefined types" (Jiang, 2012, p. 5). Based on this definition, the task of named entity recognition (NER) can be divided into the extraction and classification steps.

Typical classes for named entity recognition are PERSON, ORGANIZATION and LOCATION besides other classes that will be discussed later in this section. A comprehensive introduction to the task of named entity recognition is given by Nadeau and Sekine (2007). More recently, Yadav and Bethard (2018) and Li et al. (2022) surveyed named entity recognition with respect to deep learning systems.

In contrast to part-of-speech tagging, which requires exactly one label per token, named entities are often multi-word expressions, and only a fraction of all tokens is annotated with named entity labels. As we model NER as a sequence labeling problem that assigns exactly one label to each element of the sequence, non-entity tokens are labeled with the neutral label: O. Moreover, named entity labels are extended with a special markup like the BIO prefixes to distinguish between the tokens of a multi-word expression and two separated but consecutive expressions. Table 2.1 gives an example sentence and its annotation

Token	Segment	IOE	IOB	BIO	BIOSE
The	Outside	O	O	O	O
Nasdaq	Begin	I-ORG	I-ORG	B-ORG	B-ORG
Stock	Inside	I-ORG	I-ORG	I-ORG	I-ORG
Market	End	E-ORG	I-ORG	I-ORG	E-ORG
(NASDAQ)	Single	E-ORG	B-ORG	B-ORG	S-ORG
opens	Outside	O	O	O	O
Friday	Single	E-DATE	I-DATE	B-DATE	S-DATE
at	Outside	O	O	O	O
9:30	Begin	I-TIME	I-TIME	B-TIME	B-TIME
a.m.	Inside	I-TIME	I-TIME	I-TIME	I-TIME
EST	End	E-TIME	I-TIME	I-TIME	E-TIME

Table 2.1: Overview of standard label encodings for sequence-labeling problems.

with labels in different encoding schemes. Annotating only the label types without further markup would results in information loss for the first phrases *Nasdaq*$_{ORG}$ *Stock*$_{ORG}$ *Market*$_{ORG}$ *(NASDAQ)*$_{ORG}$, as two following entities with the same type cannot be captured without one of the sequence-labeling encodings, i.e., we cannot reconstruct that *Nasdaq Stock Market* and *(NASDAQ)* are two distinct entities.

Today, the BIO encoding (B=Beginning, I=Inside, O=Outside, Ramshaw and Marcus, 1995) is most commonly used. In contrast to the IOB encoding, it consistently marks the beginning tokens of entities, not only when two entities of the same type follow each other. This makes the encoding easier to learn for machine-learning models (Sang and Veenstra, 1999). Moreover, the BIOSE encoding (B=Beginning, I=Inside, O=Outside, E=Ending, S=Single, Borthwick, 1999) marks single-unit phrases and the ending tokens of multi-word expression. As it captures this additional semantic information, Ratinov and Roth (2009) found that it was beneficial for named entity recognition. This finding was later confirmed by Collobert et al. (2011) and Yang et al. (2018) who both concluded that the additional information from BIOSE labels can be beneficial for traditional machine-learning algorithms, as well as deep learning models. There exist some further encodings like BMEWO+ (Carpenter, 2009) which are very task- and model-specific but have less relevance in practice. Therefore, we will use either the BIO and BIOSE encodings in this work.

Concept Extraction. The actual set of labels for NER can differ according to the task. In practice, many named entity recognition corpora have unique label sets that differ for each task and, more important, by its textual domain. For example, standard entities like person names or locations are often not relevant in scientific texts. Thus, the definition of NER is relaxed in these datasets to identify any concept of interest. For example, these can be materials science concepts, such as solid oxide fuel cells (Friedrich et al., 2020) or pharmacological compounds in the clinical domain (Gonzalez-Agirre et al., 2019) or product-related concepts in social media data (Strauss et al., 2016). More details on the concept extraction (CE) tasks used in this thesis will be provided in the Chapters 3 and 7.

Anonymization. A special variant of named entity recognition and concept extraction is the detection of privacy-sensitive information, such as person names and contact information, which can be used for the de-identification or anonymization (ANON). This task is a prerequisite for the secure processing of medical documents, e.g., patient notes and clinical trials, as personal health information (PHI) has to be removed. The different PHI types are typically defined by governments, for instance, in the Health Insurance Portability and Accountability Act (HIPAA) of the United States. We will explicitly focus on anonymization methods and their impact on other information extraction tasks in Chapter 3.

Temporal Expression Recognition. Further types of special named entities are temporal expressions. These expressions are an important part of natural language and represent a certain point in time like *Yesterday* or a specific duration period, e.g., *5 minutes*. After detecting the temporal expression in a text, it has to be classified according to one of the temporal classes: DATE, TIME, DURATION and SET. The extraction and classification of temporal expressions are the first steps of temporal tagging, which combines the detection with the subsequent normalization of temporal expressions. Details on the normalization will follow in Section 2.1.2. A comprehensive overview of the task of temporal tagging is given by Strötgen and Gertz (2016). We will focus on temporal tagging methods in Chapter 8 of this work.

Further Tasks. In this work, we assume that a task is defined by its specific set of labels. As our models are generally applicable, we do not make task-specific changes in the architecture except the last classification layer, whose size must be set according to the number of target labels. With this, we can potentially address all of the previously mentioned tasks and further sequence-labeling tasks with our model architectures. We will later show that our models can deliver robust performances across many tasks.

Even though the main focus of this work lies on sequence-tagging tasks, we use text classification tasks in Chapter 5 to show the general applicability of certain methods beyond the token level. These tasks will be introduced in Chapter 5 accordingly.

2.1.2 Concept Normalization

After a named entity or concept is found and classified, it may also be normalized to a specific format. The normalization of named entities (NORM), also called disambiguation or entity linking, is the task of determining the true meaning of a named entity by linking it to a unique identifier or entry in a knowledge base or by normalizing it according to a specific format (Hoffart et al., 2011). In this work, we address two normalization tasks, ICD coding of clinical concepts and temporal expression normalization. Both will be described in more detail in the following.

ICD Coding. ICD codes are unique identifiers for clinical concepts. The abbreviation ICD refers to the "International Statistical Classification of Diseases and Related Health

Problems"(WHO et al., 2004), which basically is a hierarchical medical classification list created and maintained by the World Health Organization (WHO). In this work, we focus on ICD-10 which contains identifiers for roughly 14,000 unique identifiers. For example, the ICD-10 code *G30.0* refers to *Alzheimer's disease with early onset*.[1] In the ICD hierarchy, it is part of *G30: Alzheimer's diseases* and the more general categorization *G: Diseases of the nervous system*. The current standard is ICD-11 which replaced ICD-10 on January 1, 2022. Besides this hierarchy, there exist country-specific variants like the German modification ICD-10-GM or specialized hierarchies, e.g., ICD-O for diseases related to oncology (WHO et al., 1976), as used in this work.

Most previous methods simplified this task as a text classification problem and built classifiers using CNNs (Karimi et al., 2017) or tree-of-sequences LSTMs (Xie et al., 2018). Since ICD codes are organized under a hierarchical structure, Mullenbach et al. (2018) and Cao et al. (2020a) proposed models to exploit code co-occurrence using label attention mechanism and graph convolutional networks, respectively. We will perform ICD coding experiments in Chapter 4.

Time Expression Normalization. After detecting temporal expressions, they usually have to be normalized following some pre-defined standard format, e.g., the usually used TimeML specifications (Pustejovsky et al., 2005). TimeML's most important attributes are `type`, the class of an expression, e.g., DATE, TIME, DURATION or SET, and `value`, the normalized meaning of an expression which could look like `YYYY-MM-DD` for specific days. For example, any value for a specific day has to be given in the `YYYY-MM-DD` format,e.g., `2022-05-01` for *May 1, 2022*. The `type` is usually resolved during the extraction and the `value` during normalization. Certain temporal expressions, so-called explicit expressions, contain all necessary information for the normalization in the expression itself. For example, the term *May 1, 2022* represents the same day in every context and should be normalized to `2022-05-01`. However, many expressions are incomplete, i.e., they are not self-contained with respect to all necessary temporal information. A relative expression like *yesterday* needs an anchor point for the correct normalization. Assuming *yesterday* refers to the anchor *May, 1 2022*, for example the document creation time (DCT), it should be annotated with `type="DATE"` and `value="2022-04-30"` in TimeML.

Determining the anchor point can be challenging as it requires context information that could be given anywhere in the document. Therefore, systems for temporal expression normalization, such as HeidelTime (Strötgen and Gertz, 2013), create an intermediate context-independent representation (CIR) of the `value`. In the syntax of HeidelTime, the expression *yesterday* would result in a CIR of `UNDEF-last-day`. Similarly, an *underspecified expression*, such as *May* would be represented with a CIR of `UNDEF-year-05`. To determine the final `value`, the CIR needs to be anchored given, e.g., a reference date and further cues (such as tense information). Note that such a syntax for CIRs is language-independent. More details can be found in Section 8.3.2.

[1] `https://icd-codes.com/icd10cm/G00-G99/G30-G32/G30/G30.0` [last accessed March 5, 2022.]

Strötgen and Gertz (2016) divided temporal expressions into four subgroups based on their realization. In addition to explicit and relative expressions, there are implicit and underspecified expressions. The first one, implicit expressions, cannot be normalized without further knowledge of their implicit temporal semantics. Several different examples exist for this class, but most famous are the names of holidays and events. For example, normalizing the expression *Christmas Eve 2021* requires the knowledge that Christmas Eve is on December 24 and, thus, should be normalized to `2021-12-24`.

Underspecified expressions are more difficult to normalize, as not only the reference time has to be known, but further assumptions are required. Often, these expressions depend on context information, such as the verb's tense or external information from a calendar. For example, knowing that the temporal expression *Monday* refers to the last Monday before the anchoring date `2022-03-01` results in a normalized value of `2022-02-28` based on knowledge from a calendar.

Furthermore, temporal expressions can have different granularities. All previous examples have the day granularity. Other important units are year, month or hour. However, many other granularities exist, and, e.g., some may be divided into several subgranularities, such as a year can be divided into halves, quarters, or weeks (Strötgen and Gertz, 2016).

In this thesis, we will address the topic of temporal expression extraction and normalization in Chapter 8.

HeidelTime. As we will most often compare to HeidelTime (Strötgen and Gertz, 2013) for temporal tagging, we will provide a short description in the following. HeidelTime is a rule-based system for temporal tagging, and so far, the only publicly available system for multilingual temporal tagging with rules for roughly 200 languages (Strötgen and Gertz, 2015). Each rule consists of two parts. First, the rule specifies a regular expression that extracts temporal expressions from texts and specifies a type, e.g., DATE. In addition, the regular expressions detect several groups, e.g., day and month for *1st of May*, that are used in the subsequent normalization. The second part of the rule defines how the extracted expression should be normalized to a context-independent representation. For example, *1st of May* is normalized to the CIR `UNDEF-year-05-01` by HeidelTime. These instructions often include mappings from terms to values, e.g., *May*→`05` or *yesterday*→`last-day`. Underspecified parts of the value are marked in the CIR accordingly, for example, missing year information is denoted by `UNDEF-year`.

To determine the final `value`, HeidelTime resolves the undefined parts given, e.g., a reference date and further cues. Typical references are the document creation time or a previous temporal expression in the document. In addition, information on the sentence tense is incorporated by HeidelTime to determine the temporal relation to the reference point. For example, verbs like *will* indicate the temporal expression might be in the future with respect to the reference. Explicit expressions like *May, 01 2022* are fully normalized in their CIR (`2022-05-01`) and do not rely on the anchoring step.

2.1.3 Evaluation Metrics

As we deal with several different tasks, we also need to evaluate them with suited evaluation metrics. Therefore, we will introduce the most important metrics for this work in the following.

For this, we refer to the gold-standard annotations as the actual annotations and compare them to the predictions of a classifier or another system. For each predicted label, we can then determine whether it is a true positive (TP) or false positive (FP) depending on whether the gold standard contains the same label. Analogously, we can check for each gold-standard label whether the system missed the instance, which would be called a false negative (FN). Instances that are neither annotated in the gold standard nor predicted by the systems are correctly identified as true negatives (TN). An overview of these concepts is given in Table 2.2. Based on this categorization, we apply the following metrics.

Accuracy. Our first evaluation metric is accuracy (Acc.). It describes the number of correctly classified instances compared to the overall population (Metz, 1978).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} \tag{2.1}$$

This score is applicable when classes are more or less balanced, and exactly one label has to be assigned for each sentence or token. Following previous work, accuracy is our standard metric for sentence-level tasks like text classification and natural language inference (Bowman et al., 2015) as well as part-of-speech tagging (Plank et al., 2016).

For tasks like named entity recognition that have a non-labeled class, we want to use a different metric due to a large number of tokens without labels and the resulting dominance of true negatives. Therefore, we use the F_1-score, which combines precision and recall for these kinds of tasks.

Recall. In this context, recall (R) refers to the fraction of relevant instances that were actually detected by the system. It is a measure of the detection rate and gives information on false negatives, i.e., the number of missed relevant instances where a low recall indicates many misses (Powers, 2011).

	Predicted Positive (PP)	Predicted Negative (PN)
Actual Positive (P)	True Positive (TP) correct	False Negative (FN) type II error
Actual Negative (N)	False Positive (FP) type I error	True Negative (TN) correct

Table 2.2: Classification of system predictions compared to gold-standard annotations in 4 classes following Powers (2011).

$$\text{Recall} = \frac{\text{TP}}{\text{P}} \tag{2.2}$$

Precision. In contrast, precision (P) refers to the fraction of relevant instances among the retrieved instances. It measures how correctly the retrieved instances are classified and gives information on false positives. A low precision indicates a high confusion between classes (Powers, 2011).

$$\text{Precision} = \frac{\text{TP}}{\text{PP}} \tag{2.3}$$

$\mathbf{F_1}$-score. Looking at either recall or precision in isolation gives information on a certain error type. However, it does not reveal accurate information about the performance of a system. Maximizing recall for a certain class without taking care of precision can easily be achieved by tagging every word with this class. Of course, such a model has poor precision and is not usable in practice. The same holds for a model that maximizes precision by detecting only a fraction of easy examples and neglecting most instances. Thus, precision and recall should be considered jointly. This is done by the F_β-score (F_β) that combines precision and recall (Dice, 1945; Sorensen, 1948). It is computed as follows:

$$F_\beta\text{-score} = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{2.4}$$

Here, β is a parameter that can be used to increase the influence of either precision ($0 < \beta < 1$) or recall ($\beta > 1$). For $\beta = 1$, the F_1-score is the harmonic mean between precision and recall. The maximum F_1 of 1.0 indicates perfect precision and recall, while the minimum F_1 of 0.0 shows that either precision, recall, or both are zero.

$$F_1\text{-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.5}$$

Macro $\mathbf{F_1}$-score. The previously introduced F_1-score is also called the micro F_1-score because it is computed as the micro average of F_1 scores for all instances regardless of their specific class. In certain settings, and in particular, in the presence of class imbalances, this score might not be appropriate, as it favors majority classes and gives smaller weights to minority classes. Therefore, this score is less suited to evaluate the performance of models when all classes should be equally weighted. Therefore, the macro F_1-score was introduced to solve this problem. However, it is not well-defined how to compute the macro average. Opitz and Burst (2019) studied this problem and found two commonly used but different formulations of the Macro F_1-score.

First, the averaged macro F_1^A is the mean of all per-class F_1-scores. Here, F_{1_x} refers to the F_1-score of class x.

$$\text{Macro } F_1^A = \frac{1}{n} \sum_{x \in \text{classes}} F_{1_x} \tag{2.6}$$

Second, the Macro F_1 of averages (F_1^H) is the harmonic mean over macro precision and recall. Analogously to F_x, we define Precision$_x$ and Recall$_x$, as well as their Macro variants for this second version of the Macro F_1-score:

$$\text{Macro } F_1^H = 2 \cdot \frac{\text{Macro Precision} \cdot \text{Macro Recall}}{\text{Macro Precision} + \text{Macro Recall}} \tag{2.7}$$

Opitz and Burst (2019) recommends the usage of the first version Macro F_1^H, as this is more robust towards the error type distribution. This is confirmed by the findings of Shmueli (2019), who points out that Macro F_1^H is implemented by the popular sklearn python library and, thus, more widely adopted.

In this work, we always follow related work and use the corresponding version of the F_1-score. Usually, this is the micro F_1. Whenever a different version is used, we explicitly mention the usage of macro F_1 and whether it is the F_1^A or F_1^H-score.

Other Metrics. A more detailed overview of the previously described evaluation metrics and further measures is given by Powers (2011). Here, we only introduced the most important metrics covered by multiple chapters. Certain tasks require special metrics that we will introduce in the corresponding chapters. For example, a leak score can be computed for anonymization tasks to measure the output of personal information, and we will use several evaluation methods for rankings in Chapter 6 and Chapter 7.

Strict versus Relaxed Matching. In addition to the evaluation metrics, there are different ways to handle annotations covering multiple tokens, which is often needed for named entity recognition, temporal tagging, and concept extraction. Most prominent is strict matching, which requires a complete overlap between gold annotations and system outputs. An alternative is relaxed matching, which is more lenient and requires only a partial overlap to count a match as true positive (Batista, 2018).

Taking a look at the first examples in this section, a gold-standard annotation was *9:30 a.m. EST*. For this, a system might find that *9:30 a.m.* is a date but misses the additional timezone information. A strict evaluation on entity level that requires a complete overlap will count this as a false negative and false positive, even though the system recognized the most important part correctly. In contrast, a relaxed evaluation method will count this output as a true positive, as parts of the system's output overlap with the gold standard. This helps to estimate the real-world performance of a system more closely, as detecting an overlap is usually more useful than falsely outputting no annotation at all. However,

following related work, we will use strict matching if not stated otherwise. It is only common in the temporal tagging community to output both variants which we will also do in Chapter 8.

Statistical Significance Testing. In addition to comparing the numbers of evaluation metrics, we perform statistical significance testing whenever appropriate. For this, we follow the best practices proposed by Dror et al. (2018). As we mostly focus on sequence-labeling evaluations, we will use paired permutation testing to compare the outputs of two systems. More details on this method are given by Dror et al. (2018) and Reimers and Gurevych (2017). Whenever we use these tests, we mark statistically significant differences between the two systems and mention the details in the corresponding section.

2.2 Model Architectures for Sequential Data Processing

Regardless of the specific target task and its evaluation method, the model has to be able to process sequential textual data. However, not every machine-learning model is suited for this input format. Therefore, this section will describe three model architectures that were introduced to handle sequential data. This section will start with a short introduction of conditional random fields. Then, the focus will be shifted towards deep neural networks, where we will describe recurrent neural networks and transformers, the de-facto standard models for natural language processing with neural models.

2.2.1 Conditional Random Fields

Conditional random fields, denoted by CRFs, have been used for different sequence-tagging tasks, as they are well-known for modeling sequential data. The general CRFs, as proposed by Lafferty et al. (2001) are a type of discriminative undirected probabilistic graphical model. More specific, Lafferty et al. (2001) define a CRF on observations X and random variables Y as follows: "Let $G = (V, E)$ be a graph [with vertices V and edges E] such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X, the random variables Y_v obey the Markov property with respect to the graph" (Lafferty et al., 2001, p. 3):

$$P(Y_v|X, \{Y_w : w \neq v\}) = P(Y_v|X, \{Y_w : w \sim v\}) \tag{2.8}$$

where $w \sim v$ means that w and v are neighbors in G.

With this, a CRF is an undirected graphical model whose nodes can be divided into exactly two disjoint sets X and Y, the input and output variables, for which the conditional distribution $p(Y|X)$ is modeled (Sutton and McCallum, 2006).

General CRFs are powerful as they allow dependencies between arbitrary elements from the sequence, e.g., the first and the last element. However, this makes inference more

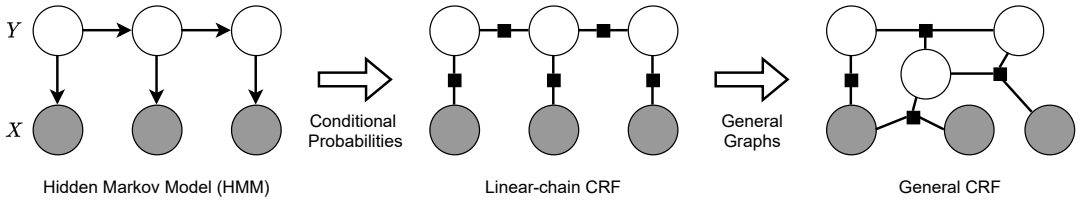


Figure 2.1: Overview of CRF models in comparison to HMMs. The plot is adapted from Sutton and McCallum (2006).

complex and often intractable when searching for an optimal solution. Thus, linear-chain CRFs are usually favored over general CRFs when used in combination with deep learning models as done in this thesis, as these only allow dependencies to previous elements similar to the conceptually simpler hidden markow models (HMMs, Baum and Petrie, 1966).

Training and Decoding. Given an input sequence X with n elements $(x_1, \ldots, x_n)$, we want to predict a sequence of labels Y, which are called states (Sutton and McCallum, 2006). This can be done by computing the conditional probability $P(Y|X)$ with a scoring function over all possible tag sequences $\mathbb{Y}$:

$$P(Y|X) = \frac{\exp(\text{Score}(X, Y))}{\sum_{\hat{Y} \in \mathbb{Y}} \exp(\text{Score}(X, \hat{Y}))} \tag{2.9}$$

In this work, we use CRFs on top of neural models, which we call an NN-CRF in the case of arbitrary neural networks. For this, we use a scoring function that incorporates neural features and trainable transition weights following Huang et al. (2015). The neural features are taken from the hidden states h, e.g., the last layer of a BiLSTM model, for each element in the input sequence. These features are called emission scores and are stored in a matrix P of size $n \times |T|$ where $|T|$ is the size of the tagset T, i.e., the set of unique labels. The entry P_{ij} refers to the score of the j-th tag of the i-th word in the sequence. The transition weights are stored in a matrix A of size $|T| + 2 \times |T| + 2$, such that entry A_{ij} contains the probability of transitioning from tag i to tag j. Note that we add two additional tags (START and STOP) to mark the beginning and end of a label sequence. Based on the emission scores P and transition scores A, Lample et al. (2016) define the scoring function and the training and decoding objectives as follows:

$$\text{Score}(X, Y) = \sum_{i=0}^{n} A_{y_i, y_i+1} + \sum_{i=1}^{n} P_{i, y_i} \tag{2.10}$$

During training, the log-probability of the correct tag sequence is maximized:

$$\log P(Y|X) = \text{Score}(X, Y) - \log(\sum_{\hat{Y} \in \mathbb{Y}} \mathrm{e}^{\text{Score}(X, \hat{Y})}) \tag{2.11}$$

During decoding, the output sequence Y^* with optimal scores is selected:

$$Y^* = \arg\max_{\hat{Y} \in \mathbb{Y}} \text{Score}(X, \hat{Y}) \tag{2.12}$$

For this, a dynamic programming approach like the Viterbi algorithm can be used (Viterbi, 1967; Rabiner, 1989) to speed up the computations by storing relevant intermediate results that are useful for multiple sequences. This is the de-facto standard decoding algorithm for NN-CRF models, as it is faster in practice to use dynamic programming compared to computing scores for all possible sequences independently (Lample et al., 2016).

More detailed descriptions of CRF models and their decoding, as well as theoretical considerations are given by Lafferty et al. (2001); Sutton and McCallum (2006); Klinger and Tomanek (2007); Collins (2015).

Examples CRF Features for Named Entity Recognition. In order to process an input sequence X, each element of the sequence has to be represented by a set of features. Popular features for named entity recognition, among others, are summarized by Liu et al. (2017b) and include:

- **Bag-of-words features** encode the unique id of a specific word, also called unigram, or its corresponding lemma. Additional information on neighboring words in a certain range is typically included as well. For example, indicators for given bigram or trigram combinations in the sequence covered by the word or the unigram ids of neighboring tokens are well-known features.

- **Orthographical features**: This includes information on whether the word is upper case, contains uppercase characters, contains punctuation marks, contains digits, or non-ASCII characters. Each of these features is typically binary and represents one property, e.g., *is_uppercase*.

- **Word shapes** are an extension of the binary orthographic features that give a fine-grained overview on the orthography of the word. For this, every character is mapped to one of the following symbols. *C* for uppercase characters, *c* for lowercase characters, *d* for digits, *p* for punctuation symbols and *x* and other characters. Using this mapping, the token *GPT-3* has the shape *CCCpd*.

- **Frequency** and word count are typically used as auxiliary features to estimate the quality of other input features and may be easily derived from larger text corpora. Higher-frequency words are typically well known and, thus, easier for the model to process, while low-frequency words tend to reduce model confidence. In addition, certain features, such as part-of-speech tags or morphological affixes, which have to be computed with a different model in advance, may have lower quality for infrequent words due to failures in the first step.

- **Word representation features**: In contrast to the previous, manually designed features, a CRF can also incorporate continuous vector representations as in the NN-CRF model described above. This allows the CRF model to include embedding vectors and hidden states derived from neural models.

In this work, we focus on word representations derived from neural networks that will be described in the following. For this, the CRF can be seen as the final layer of such a network. This allows us to incorporate the benefits of CRF models for sequential decoding and the representation learning power of deep learning systems, which is particularly helpful for the previously described sequence-labeling tasks.

2.2.2 Neural Networks

In recent years, neural networks outperformed traditional machine-learning methods like the CRF in many fields due to increased access to computational power and data. One of these fields is natural language processing, including the tasks covered in this work.

In general, a neural network for classification $f_\theta(x)$ can be considered a function with parameters θ that maps an input x to a probability distribution over labels y. Deep feed-forward networks with L layers and without skip connections (Srivastava et al., 2015; He et al., 2016) or recursion (Goller and Küchler, 1996; Sperduti and Starita, 1997) can be seen as a composition of functions $f_\theta^l(\cdot)$, corresponding to each layer l (Collobert et al., 2011):

$$f_\theta(x) = f_\theta^L(f_\theta^{L-1}(\cdots f_\theta^1(x)\cdots)) \tag{2.13}$$

While these networks are suited for a wide range of tasks, there are several issues when applied in the context of natural language processing. Most important, feed-forward networks cannot handle sequential data properly, as only the current input is considered, and the network cannot memorize previous inputs. This is in contrast to the sequential nature of textual information, which requires an understanding in larger contexts due to complex relationships between neighboring and more distant words.

Thus, for this work, particularly two types of neural networks are interesting. The first models are recurrent neural networks (RNN), which can capture long-distant relationships between words in a sentence. This is achieved by keeping a memory vector from previous timesteps when unrolling the model over time, i.e., by applying it consecutively to each element of the sequence. Vanilla recurrent neural networks and two of their variants are discussed in section 2.2.3. In contrast, transformers use an attention mechanism to capture word relationships in longer contexts. More details on transformers will follow in Section 2.2.4.

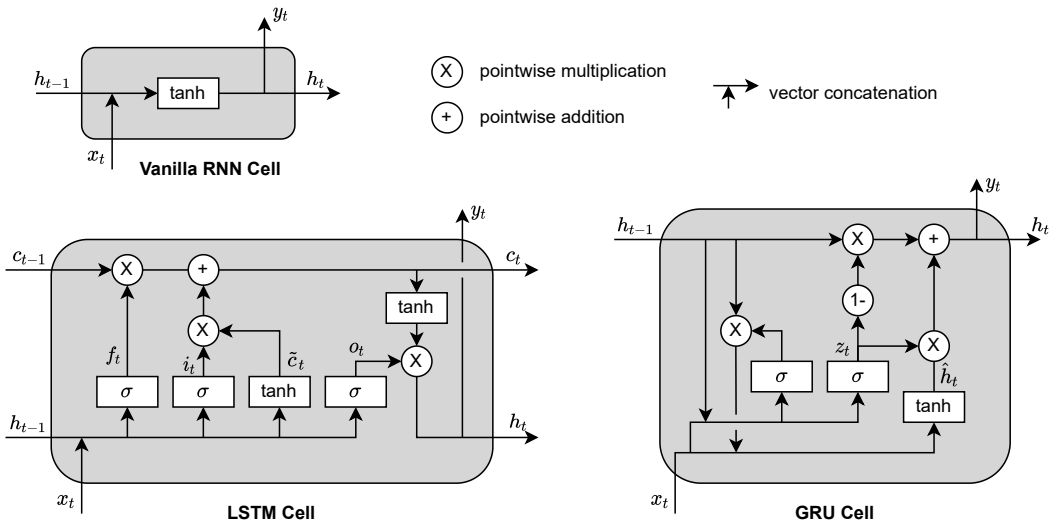


Figure 2.2: Illustration of three recurrent neural architectures: Vanilla RNN, LSTM and GRU. The diagrams are inspired by Phi (2018).

2.2.3 Recurrent Neural Networks

This section will describe three recurrent neural network architectures: (1) the Vanilla RNN (2) long short-term memory and (3) gate recurrent units. An visual overview of these architectures is given in Figure 2.2.

Vanilla Recurrent Neural Networks. While feed-forward networks are not able to maintain memory and, thus, memorize previous information, the conceptually more complex recurrent networks were designed to overcome this gap and lead to major improvements in the processing of time series, textual, and other sequential data. The Elman network architecture, to which we will refer to as *vanilla recurrent neural network*, is defined as follows as proposed by Elman (1990):

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{2.14}$$

For each element at timestep t, the recurrent layer computes the function from Equation 2.14 with x_t being the input vector, h_t the hidden layer vector and W, U, b trainable parameter matrices and vectors. σ_h is an activation function. One possible activation is $\tanh$, as this ensures that all values stay in the range (-1, 1), which helps in regulating the network's output (Elman, 1990).

By memorizing the state h_t over time, the recurrent model has access to information from previous timesteps and thus, can perform sequence predictions that feed-forward networks are not able to model. In addition, this distributed hidden state does not only allow to store past information, but it also enables weight sharing over all timesteps (Jordan, 1997).

In order to process the complete sequence, each element has to be input into the RNN consecutively. Starting at $t = 1$, the first element is processed with the initial memory

$h_0 = 0$ set to the zero vector. All further elements are input at corresponding timesteps t while the hidden state h_{t-1} is recursively maintained and forwarded. This makes the recurrent neural network more powerful, but it also increases training time and complexity, as the network has to be unfolded over all timesteps. During training, the gradient updates of the network are collected for all timesteps. This process is called back-propagation through time (BPTT Werbos, 1988).

Long Short-Term Memory. A problem when using gradient descent for vanilla RNNs is that error gradients vanish exponentially quickly with respect to the sequence size. One solution to this problem that allows for longer sequences, and thus long-range dependencies, is the long short-term memory (LSTM) as proposed by (Hochreiter and Schmidhuber, 1997). The LSTM introduces three so-called gates which allow a more controlled information flow and is defined as follows:

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
\tag{2.15}
$$

New components are the cell c, the input gate i, the output gate o, and the forget gate f. The cell c remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell. The input gate i controls the influence of memorized information in the cell and decides which piece of information is relevant for the current time step using the sigmoid function (σ) and the element-wise multiplication with the input gate. The output gate o controls which information is given to the hidden state by, first, normalizing the cell state using the hyperbolic tangent function ($\tanh$) followed by the element-wise multiplication with the output gate. The forget gate f controls which information from the cell state shall be forgotten. It uses the sigmoid function (σ) to compute relevancy scores between 0 and 1, where elements with 0 scores are removed, and elements closer to one are remembered.

Gated Recurrent Units. Gated recurrent units (GRUs) are an alternative to the previously described LSTM and introduced by Cho et al. (2014). The GRU does not have an output gate, and thus, it has fewer parameters. This has advantages in certain situations, for example, in low-resource settings, as GRUs have been shown to exhibit better performance on smaller datasets (Chung et al., 2014). The definition for a GRU is as follows:

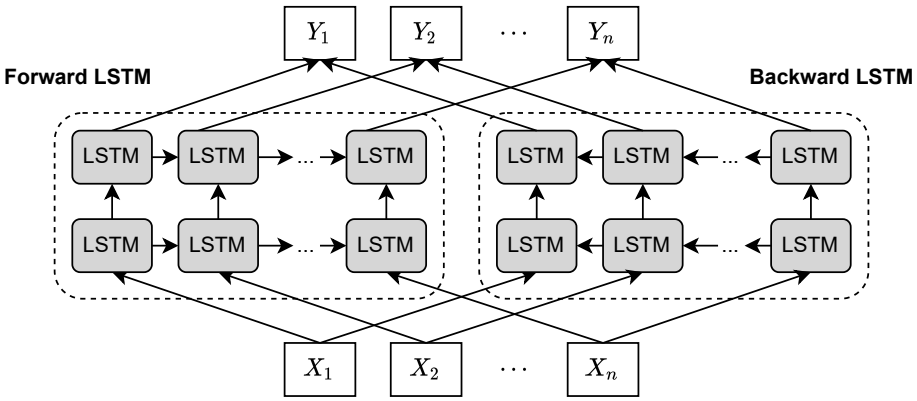


Figure 2.3: Illustration of a bidirectional LSTM architecture. The diagram is based on figures by Devlin et al. (2019).

$$
\begin{aligned}
z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\
r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\
\hat{h}_t &= \tanh(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \\
h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \hat{h}_t
\end{aligned}
\tag{2.16}
$$

Here, z and r are the update and reset gates, which have similar functions as the input and forget gates of the LSTM.

Bidirectional Recurrent Neural Networks. One limiting factor of these recurrent architectures described previously is the strict forward flow of information into one direction, i.e., the memorized timesteps represent past information, and the model does not have access to future steps. While this is usually wanted when processing time-series data, textual data has a different structure, and words often depend on other words that occur later in the sentence.

Bidirectional recurrent neural networks (BRNN) connect two recurrent layers of opposite directions to the same output as visualized in Figure 2.3. For example, there can be two independent LSTM layers, with one being used to process a sentence starting from the first word, while the second layer starts at the last words and processes the sentence in the reverse order. Then, the outputs of both LSTM layers are combined by, e.g., concatenating both output vectors. With this, the output layer can get information from past (backward), and future (forward) states simultaneously (Schuster and Paliwal, 1997) which makes bidirectional networks more powerful than unidirectional networks when processing texts. Many of our experiments are based on such bidirectional LSTM networks, which we denote by BiLSTM.

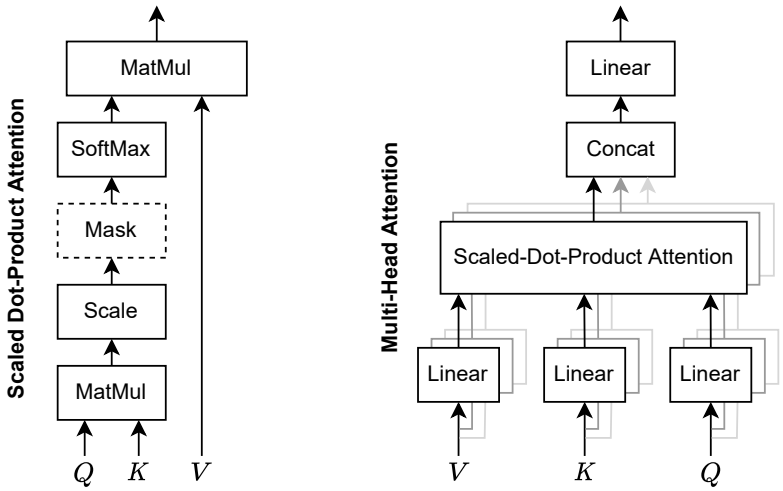


Figure 2.4: Illustration of the scaled dot-product attention and its use in the multi-head attention as used by transformer models. This diagram is based on figures by Vaswani et al. (2017).

2.2.4 Transformers

While RNN models are well-suited for natural language processing tasks, they come with a major drawback in scalable environments. Due to the sequential unrolling of hidden states over the input sequence, the RNN has to be applied to every single input instance iteratively. Thus, they cannot be parallelized efficiently for long sequences. The transformer is another deep learning architecture for sequential data that was created to overcome this limitation by processing the whole input sequence at once and assigning dynamic attention weights to focus on relevant parts for each input (Vaswani et al., 2017). In particular, the transformer evolves around the core concept of attention. This is a differential weighting mechanism that assigns significance scores to inputs. A visual overview of the attention functions and different layers are given in Figure 2.4 and 2.5, respectively. Transformer attention and the encoder-decoder structure will be described in more detail in the following.

Scaled Dot-Product Attention. The main idea of transformer models is scaled dot-product attention. Assuming we pass a textual input into the transformer, this attention method calculates attention weights between the target token and every other token in the input. These attention weights can be seen as relevance scores, whereas high scores signal important dependencies or relationships between the tokens. The computed attention weights are then further used to create weighted combinations of the token and its context, which results in a contextualized representation of the token. In contrast to the unfolding memory of RNNs, this method can be computed independently for all tokens in the input sequence to capture their dependencies in parallel.

In practice, a transformer consists of multiple attention units. Each of these units consists of three trainable matrices W^Q, W^K, and W^V. The input embedding x_i for each token

i is multiplied with each of these matrices to produce three internal vectors $q_i = x_i W^Q$, $k_i = x_i W^K$ and $v_i = x_i W^V$ called query, key and value vectors, respectively.

The actual attention weights a_{ij} between two tokens i and j are computed as the dot product between the query vector q_i and key vector k_j. The weights are further normalized with respect to the dimensions of the key vectors to stabilize the gradients and then passed to a softmax function. Following Vaswani et al. (2017), the attention calculation for all tokens can be expressed as a matrix multiplication calculation using the softmax function. For this, the matrices Q, K and V are defined as the matrices with their i-th row vectors being the query, key and value vectors q_i, k_i, and v_i, respectively. With this, the attention function is defined as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{2.17}$$

Multi-Head Attention. As said earlier, a transformer model consists of multiple attention units. Each of these units consists of a set of (W_Q, W_K, W_V) matrices and is called an attention head. Furthermore, each layer typically has more than one attention head which is called multi-head attention. This allows the model to learn different relationships between token pairs. For example, one attention head might learn tense information between verbs, while a different head focuses on the relationships between verbs and their adverbs. In a different example, one attention head might focus on neighboring words, while another head learns long-distance relationships. The multi-head attention function is defined as follows, where $[\cdots]$ is the concatenation operation for vectors.

$$\begin{aligned} \text{MultiHeadAttention}(Q, K, V) &= \text{Concat}[\text{head}_1; \cdots ; \text{head}_n]W^O \\ \text{with: } \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_I^V) \end{aligned} \tag{2.18}$$

Tenney et al. (2019) found that attention heads in earlier transformer layers related to low-level NLP tasks, while attention heads of higher layers can encode semantic knowledge. This indicates that a transformer model might be able to mimic a classical NLP pipeline. In practice, attention heads are less interpretable, and the learned relationships are less meaningful to humans. This leads to many discussions whether the attention scores can be used as explanations for model decisions (Serrano and Smith, 2019; Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). Moreover, not all attention heads are equally useful (He and Choi, 2021) and some may be pruned to reduce model size (Behnke and Heafield, 2020; Prasanna et al., 2020).

Encoder-Decoder Architecture. The transformer was introduced as a sequence-to-sequence (seq2seq) model in the context of machine translation (Vaswani et al., 2017) and like earlier seq2seq models, this transformer model uses a multi-layer encoder-decoder architecture as visualized in Figure 2.5.

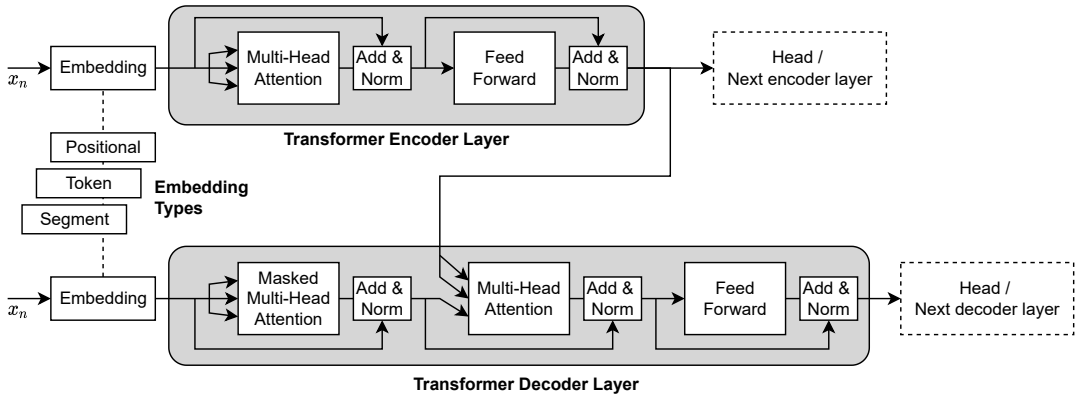


Figure 2.5: Illustration of the standard transformer encoder-decoder structure and its internal attention mechanisms. This diagram is inspired by Vaswani et al. (2017).

Each encoder layer consists of a multi-head attention unit followed by a feed-forward network for further processing and a layer normalization function. In addition, residual connections, so-called skip connections (He et al., 2016), allow the gradients to surpass the attention and feed-forward modules and avoid vanishing gradients in deep networks by this. Then, the encoder output is further processed by the next encoder layer.

The decoder layers have a similar structure as encoder layers but incorporate the encoder output with a second multi-head attention module in addition. In contrast to the encoder, the decoder has only access to the past tokens. For this, left-to-right decoding has to be used that masks the current and future tokens to prevent reverse information flow (Vaswani et al., 2017).

The last layer of a transformer model is typically one or more linear layers depending on the target tasks. For natural language processing models, this layer may map the internal representation output of the decoder to a fixed vocabulary of words, e.g., such that individual words can be predicted for machine translation or text generation.

In such a transformer, the decoder structure is symmetrical to the encoder, and both have the same number of layers. Later works found that using only a single encoder layer for auto-regressive models (Radford et al., 2019) or decoder layer for bidirectional models (Devlin et al., 2019) can be sufficient. These models have been proven to be particularly useful for language modeling and the resulting language models as universal word representation. Thus, the next section will focus on this and other representation methods, which are nowadays typically used in modern neural NLP systems.

2.3 Pre-Trained Word Representations

Word embeddings and other pre-trained language representations are the core input component of many neural network-based models for NLP tasks. These are numerical representations of words or sentences, as neural architectures do not allow the processing of strings and characters as such. In this section, we will introduce different word embedding

methods and language models in more detail and discuss how they can be used to represent input sequences for NLP tasks in non-standard domains and languages.

2.3.1 Word Embeddings

Collobert et al. (2011) showed that training word embeddings for the task of language modeling in a self-supervised fashion on a large-scale corpus result in high-quality word representations, which can be reused for other downstream tasks as well. For this, Collobert et al. (2011) proposed to learn a word lookup table as a linear layer where the representation for each word in a fixed vocabulary is given by a row in the lookup table layer. Given a sequence of words, the lookup table layer outputs an embedding matrix, which can then be fed to further neural network layers. For training these word embeddings, Collobert et al. (2011) used a window approach for language modeling using pairwise ranking.

The first embedding method that was widely adopted is the word2vec model by Mikolov et al. (2013c). It is based on the assumption by Firth (1957) that neighboring words define the meaning of the target word, and thus, it tries to learn embeddings based on the relationships between neighboring words. More precisely, Mikolov et al. (2013c) proposed two variants. The continuous bag of words (CBOW) model learns to predict a center word c given its context of neighboring words o as a bag of words. The skipgram model (SG) does the reverse and learns to predict the context words given a center word. In practice, the CBOW method tends to be faster, but the skipgram model became more popular and is said to create better representations for infrequent words (Mikolov et al., 2013c). Therefore, all word2vec embeddings used in this work are trained with skipgram method if not stated otherwise.

More precisely, the skipgram model is trained to minimize the following loss objective for center words w_t in a sequence of length T within a fixed-size window of m:

$$L_{SG} = 1 - \frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m; j \neq 0} \log P(w_{t+j}|w_t) \tag{2.19}$$

wheres the conditional probability for an outside context word o given a center word c is defined as:

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)} \tag{2.20}$$

In theory, this allows training high-quality word representation based on neighboring words. However, this approach is not practical due to the high computational costs of the softmax function. In particular, for large vocabulary sizes, the computation becomes intractable as the denominator increases with each element in the vocabulary. This was recognized by Collobert et al. (2011) as well.

Therefore, Mikolov et al. (2013a) use the negative sampling procedure. For this, the few positive samples that appear in the sentence context, as well as a small sample of negative

words from the vocabulary, are considered in the softmax. The samples are generated based on word probabilities and allow to reduce the softmax costs significantly. The loss objective for skipgram with negative sampling is formulated as follows:

$$L_{\text{SG + neg. sample}}(u, c) = 1 - \log\ \sigma(u_o^T v_c) + \sum_{i=1}^{k} \mathbb{E}[X]_{j \sim P(w)}[\log\ \sigma(-u_j^T vc)] \tag{2.21}$$

Here, the first part maximizes the probability of co-occurring words. In the second part, k samples are randomly drawn from $\mathbb{E}[X]_{j \sim P(w)}$ based on the word probabilities as the negative samples which are not found in the context. This operation is very sparse compared to the size of the complete vocabulary and the embedding matrix, as these updates should only concern the rows corresponding to positive or negative samples. These sparse matrix updates help in particular for distributed training and avoid the transfer of large updates across systems.

Pennington et al. (2014) proposed GloVe embeddings as an alternative to the word2vec model. GloVe uses a count-based model and is usually easier and faster to train compared to the predictive word2vec model (Baroni et al., 2014). However, both are tied to a fixed vocabulary and cannot represent so-called out-of-vocabulary tokens.

Subword-based embeddings such as fastText n-gram embeddings (Bojanowski et al., 2017) and byte-pair-encoding embeddings (Heinzerling and Strube, 2018) addressed these out-of-vocabulary issues by splitting words into multiple subwords, which in combination represent the original word. These fastText embeddings are trained on character n-grams, and each word vector is computed by the sum of its components. This has the advantage that the n-gram vectors are able to capture information below word level, which allows the creation of word representations for unknown and unseen words (Mikolov et al., 2018). For example, Zhu et al. (2019) demonstrated that subword-based embeddings are beneficial for low-resource sequence-labeling tasks, such as named entity recognition and typing and outperform word-level embeddings. In addition, pre-trained embeddings were published for more than 270 languages for both embedding methods. This enabled the processing of texts in many languages, including multiple low-resource languages found in Wikipedia.

2.3.2 Pre-Trained Language Models

More recently, a trend emerged of pre-training large embedding models using a language model objective to create context-aware word representations by predicting the next word or sentence (Howard and Ruder, 2018). This includes pre-trained transformer models, such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019b). These methods are particularly helpful for low-resource languages for which large amounts of unlabeled data are available, but task-specific labeled data is scarce (Cruz and Cheng, 2019).

In particular, BERT (Devlin et al., 2019) became well-known in the broader NLP community and is used across many tasks and languages. BERT itself is a bidirectional transformer encoder. In contrast to the transformer architecture presented in Section 2.2.4, the

decoder stack is omitted for BERT to allow bidirectional connections. BERT is trained using a masked language modeling objective. For this, certain parts of the input sequence are masked, i.e., some words are replaced by a special *Mask* token. Then, the model has to predict the most probable replacement for the mask. In addition, BERT is also trained on next-sentence predicting, i.e., given two sentences, the model has to decide if the sentences follow each other in a document. This helps to learn sentence-level representations, while the masked language modeling objective optimizes BERT on the token level. As transformer models can be parallelized highly efficiently, BERT and other transformer language models were trained on large-scale datasets with this self-supervised training setup. The resulting language models can be used to generate contextualized input representations for words and sentences or can be fine-tuned directly for downstream applications (Devlin et al., 2019). Both approaches lead to great improvements compared to standard word embeddings and will be an important part of this work.

While pre-trained language models achieve remarkable performances in many tasks, it is still questionable if these methods are suited for real-world low-resource scenarios. For example, all of these models have large hardware requirements, in particular, considering that the performance of transformers keeps scaling with their size and training time (Raffel et al., 2020). Therefore, these large-scale methods might not be suited for low-resource scenarios where hardware is also a resource aspect.

Biljon et al. (2020) showed that low- to medium-depth transformer sizes perform better than larger models for low-resource languages, and Schick et al. (2020) managed to train models with three orders of magnitude fewer parameters that perform on-par with large-scale models like GPT-3 on few-shot tasks by reformulating the training task and using ensembling. Melamud et al. (2019) showed that simple bag-of-words approaches are better when there are only a few dozen training instances or less for text classification, while more complex transformer models require more training data. Bhattacharjee et al. (2020) found that cross-view training (Clark et al., 2018) leverages large amounts of unlabeled data better for task-specific applications in contrast to the general representations learned by BERT. Moreover, data quality for low-resource languages, even for unlabeled data, might not be comparable to data from high-resource languages. For example, Alabi et al. (2020) found that word embeddings trained on larger amounts of unlabeled data of low-resource languages are not competitive to embeddings trained on smaller but curated data sources of higher quality.

2.3.3 Domain-Specific Pre-Training

The language of a specialized domain can differ tremendously from the language of the general domain, such as the one found in news articles (Ben-David et al., 2006). For example, scientific articles often contain formulas and technical terms, which are not observed in news articles. However, the majority of recent language models are pre-trained on general-domain data, such as texts from the news or web domain, which can lead to a so-called *domain gap* when applied to a different domain (Glorot et al., 2011).

One solution to overcome this gap is the adaptation to the target domain by fine-tuning the language model. Gururangan et al. (2020) showed that training a model with additional domain-adaptive and task-adaptive pre-training with unlabeled data leads to performance gains for both high- and low-resource settings for numerous English domains and tasks. This is also displayed in the number of domain-adapted language models (Alsentzer et al., 2019; Huang et al., 2019; Adhikari et al., 2019; Jain and Ganesamoorty, 2020, i.a.,), most notably BioBERT (Lee et al., 2020) that was pre-trained on biomedical PubMED articles and SciBERT (Beltagy et al., 2019) for scientific texts. For example, Friedrich et al. (2020) showed that a BERT model from the general domain performs well in the materials science domain, but the domain-adapted SciBERT performs better.

2.3.4 Multilingual Language Models

Analogously to low-resource domains, low-resource languages can also benefit from resources available in other high-resource languages. For this, one option is a single model trained on many languages at once, such as multilingual BERT (mBERT, Devlin et al., 2019) or XLM-RoBERTa (XLM-R, Conneau et al., 2020). These models are trained using unlabeled, monolingual corpora from different languages and can be used in cross- and multilingual applications due to many languages seen during pre-training.

In cross-lingual zero-shot learning, no task-specific labeled data is available in the target language. Instead, labeled data from a different language is leveraged. A multilingual model can be trained on the target task in a high-resource language and afterward, applied to the unseen target languages, such as for named entity recognition (Lin et al., 2019; Hvingelby et al., 2020), reading comprehension (Hsu et al., 2019), or POS tagging and dependency parsing (Müller et al., 2020). Hu et al. (2020) showed, however, that there is still a large gap between low and high-resource settings. Lauscher et al. (2020) and Hedderich et al. (2020) proposed adding a minimal amount of target-task and -language data (in the range of 10 to 100 labeled sentences) which resulted in a significant boost in performance for classification in low-resource languages.

The transfer between two languages can be improved by creating a common multilingual embedding space of multiple languages. This is useful for standard word embeddings (Ruder, 2019b) as well as pre-trained language models. For example, by aligning the languages inside a single multilingual model, i.a., in cross-lingual (Schuster et al., 2019; Liu et al., 2019a) or multilingual settings (Cao et al., 2020b). This alignment is typically done by computing mappings between two different embedding spaces, such that the words in both embeddings share similar feature vectors after the mapping (Mikolov et al., 2013b; Joulin et al., 2018).

One method is the alignment via an orthogonal transformation that is based on a dictionary $\{y_i, x_i\}_{i=1}^{n}$ of paired words from a target (x) and source language (y). For these dictionaries, one can compute an orthogonal transformation matrix O by maximising the

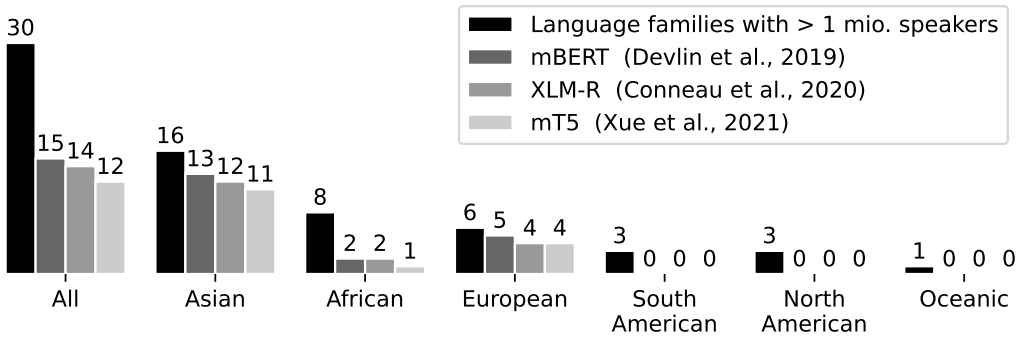


Figure 2.6: Language families with more than 1 million speakers covered by three popular multilingual language models.

cosine similarity of pairs in the dictionary (Smith et al., 2017), where I denotes the identity matrix:

$$\max_{O} \sum_{i=1}^{n} y_i^T O x_i \quad \text{with} \quad O^T O = I \tag{2.22}$$

Artetxe et al. (2016) proposed a numerically exact solution for the transformation by using singular value decomposition (SVD). Based on two matrices X_D and Y_D that contain the pairwise entries of our dictionary, such that the i-th row of X_D corresponds to the i-th row of Y_d, one can compute the SVD as:

$$Y_D^T X_D = U \Sigma V^T \tag{2.23}$$

with U and V being of columns of orthonormal vectors and Σ being a diagonal matrix containing the singular values, such that $O = UV^T$. Finally, both embedding spaces can be transformed to a common space by applying the transformations V^T and U^T to the source and target spaces, respectively.

Using a similar method, fastText embeddings were published for 44 languages that are all aligned to a common space (Joulin et al., 2018). More precisely, all embeddings were aligned with pairwise transformations to the English space. This allows to use different embeddings inside the same model and helps when two languages do not share the same space inside a single model (Cao et al., 2020b). For example, Zhang et al. (2019) used bilingual representations by creating cross-lingual word embeddings using a small set of parallel sentences between the high-resource language English and three low-resource African languages, Swahili, Tagalog, and Somali, to improve document retrieval performance for the African languages.

While these multilingual models are a tremendous step towards enabling NLP in many languages, possible claims that these are universal language models do not hold (Pires et al., 2019). For example, mBERT covers 104, XLM-R 100 and mT5 (Xue et al., 2021) 101 languages, which is a third of all languages in Wikipedia, as outlined earlier. Further, Wu and Dredze (2020) showed that, in particular, low-resource languages are not well-represented in mBERT. Figure 2.6 shows which language families with at least 1 million

speakers are covered by mBERT, XLM-R, and mT5.[2] In particular, African and American languages are not well-represented within the transformer models, even though millions of people speak these languages. This can be problematic, as languages from more distant language families are less suited for transfer learning, as Lauscher et al. (2020) showed.

Nonetheless, pre-trained multilingual transformer models often show good results in practice. Therefore, our models in Chapter 8 of this work are based on these methods.

2.4 Multi-Task Training

The main reason pre-trained word representations are a major part of modern deep learning systems for NLP is the wide knowledge contained inside these representations. By training on large-scale collections on unlabeled texts, these models learn general world knowledge that is useful for many other tasks, i.e., parts of the model are trained for one pre-training task and then transferred to the actual target task (Howard and Ruder, 2018). Multi-task training leverages this idea of feature sharing across tasks in a more systematic way and enables the transfer or simultaneous training of complete models. For this work, multi-task training is an important concept as it is well-known to improve NLP systems (Collobert and Weston, 2008), in particular, for low-resource scenarios (Lin et al., 2018). Thus, three variants of multi-task training are used in parts of this thesis, namely joint (Chapter 3) and sequential multi-task training (Chapter 4 and Chapter 7), as well as adversarial training (Chapter 5 and Chapter 8). All three are explained in more detail in the following.

2.4.1 Joint Multi-Task Training

The first variant is joint multi-task training, often simply referred to as multi-task training (Ruder, 2017). Generally speaking, this type of training leverages the training resources of multiple tasks by jointly training a single model. With this, the model is able to incorporate the training signals of related tasks and, thus, learns more general features. On the one hand, this can help to prevent overfitting on the target task as the shared features are trained to be useful on many tasks. On the other hand, there often is a lack of labeled data for the target task. Using the training data of a related auxiliary task can help to learn generally applicable features, which reduces the need for target supervision (Hedderich et al., 2021a).

Sharing the feature representation R directly between different tasks is called hard parameter sharing. An alternative is the so-called soft sharing for which the features are not completely shared, as each task has its own feature representation, but its weights are tied closely together with the help of regularization methods (Ruder, 2017). This allows each model to learn more task-specific representations, but it makes the training more complex. We will focus on methods with the more commonly used hard parameter sharing, as this allows to train a single model for many tasks. For this, the model is split into two parts.

[2] A language family is covered if at least one associated language is covered. Language families can belong to multiple regions, e.g., Indo-European belongs to Europe and Asia.

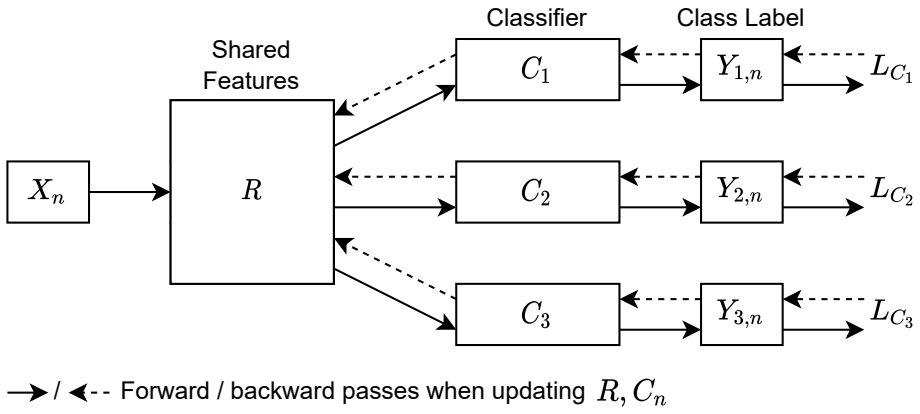


Figure 2.7: Overview on the standard multi-task learning architecture with hard parameter sharing and the gradient flows.

The first part is the feature representation R, which is shared between all tasks. This could be anything between a single embedding layer or a complete transformer model. Based on this, there exists one sub-network C_n for each task n, which can have arbitrary complexity. As we focus on classification tasks in this work, each of these sub-networks typically is a classification layer and, therefore, also called a classification head. In some settings, the head is preceded by an individual recurrent layer per task. This general architecture is visualized in Figure 2.7. Depending on the training data, the model can be trained on all tasks simultaneously if all examples in the training batches are labeled for all tasks. Otherwise, the model might be trained by alternating batches or epochs per task. During training, each task n will update its corresponding classification head C_n, as well as the shared feature representation R. It will not influence the weights of other classification heads. The resulting loss function is typically a weighted combination of the individual losses L_{C_n}, whereas tasks can be weighted according to their importance using the weights α, β, γ, ...:

$$L = \alpha \cdot L_{C_1} + \beta \cdot L_{C_2} + \gamma \cdot L_{C_3} + \dots \tag{2.24}$$

In practice, certain tasks are more useful for multi-task training than others. For example, lower-level NLP tasks, like part-of-speech tagging and chunking, seem to be well-suited when the target is a higher-level task, such as question answering or named entity recognition (Vu et al., 2020). However, choosing a good set of training tasks is a challenging problem on its own, which is also addressed in this work in Chapter 7. One might want to select related tasks based on similarity measures (Bingel and Søgaard, 2017; Schröder and Biemann, 2020). Another example is Meta-Learning (Finn et al., 2017). Given a set of auxiliary tasks and a low-resource target task, meta-learning trains a model to decide how to use the auxiliary tasks in the most beneficial way for the target task. For NLP, this approach has been evaluated on tasks , such as sentiment analysis (Yu et al., 2018), user intent classification (Yu et al., 2018; Chen et al., 2020), natural language understanding

(Dou et al., 2019), text classification (Bansal et al., 2020) and dialogue generation (Huang and Du, 2019).

2.4.2 Sequential Multi-Task Training

Multi-task training does not necessarily involve the simultaneous training of multiple tasks. In analogy to the transfer of pre-trained word representations, the different tasks might be trained sequentially. For this, the model is trained on one task, and its weights are saved. Then, the shared weights are transferred to the second model, which is then trained on the second task. This can result in a training chain of many tasks (Ruder, 2017). In this work, we mostly focus on a two-step approach by training a model on a source task, which is then transferred to the target task. We call this model transfer.

Sequential training has the advantage of reusability. Once a model is trained and stored, it can be reused in many other training procedures. On the negative side, the sequential transfer has a higher chance of overfitting the model to each individual task, and knowledge from previous tasks might be forgotten, which is called catastrophic forgetting (Goodfellow et al., 2014a).

2.4.3 Adversarial Training

Differences in the feature representations between the source and the target domain can be an issue in transfer learning, especially in neural approaches where it can be difficult to control which information the model takes into account. Adversarial discriminators (Goodfellow et al., 2014b) can prevent the model from learning a feature representation that is specific to a data source. Adversarial training is an unsupervised method, i.e., it does not require any labeled data. Most often, information on the data source is sufficient for this training method. NLP applications include the removal of biases and unwanted properties. For computer vision, adversarial training is typically used to train generative models, so-called generative adversarial networks (GANs) in which the generative network competes with the adversarial network, e.g., to create realistic images. (Goodfellow et al., 2014b).

The process of adversarial training for NLP is visualized in Figure 2.8 and consists of three major components. The classifier C and the shared features R are similar to the standard multi-task training setup. The input representation is shared between the discriminator and downstream classifier. However, the features R are trained with gradient reversal for the adversarial path to fool the discriminator (Raff and Sylvester, 2018). In particular, the gradients arriving at the features are multiplied with a hyperparameter $\lambda < 0$ in order to reverse the training effects.

The adversarial discriminator D, also called adversary, tries to classify the shared features according to certain properties. These properties should be chosen according to the target task, as adversarial training tries to remove their inherent effects. For example, language- or domain-specific information can be removed when the discriminator has to

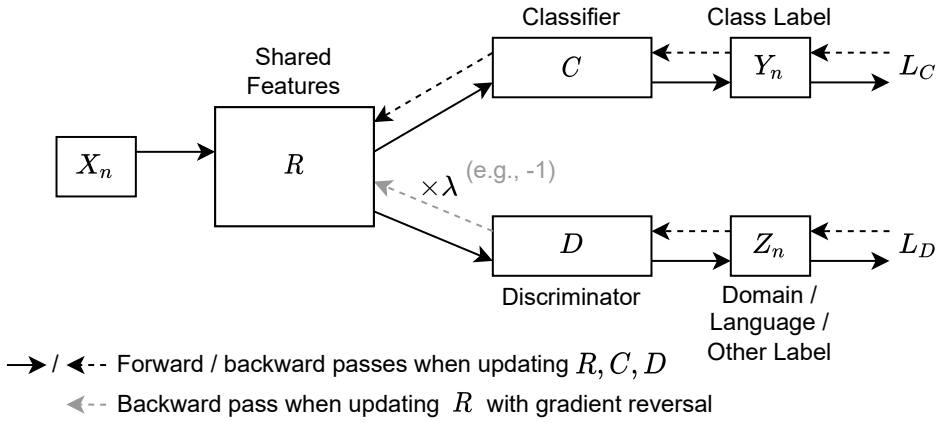


Figure 2.8: Overview of adversarial training and the gradient flows.

distinguish between the sentence origins from particular languages or domains given the input features.

In our work, the discriminator D is a multinomial non-linear classification model with a standard cross-entropy loss function L_D in all settings covered by this work. In our sequence-tagging experiments, the downstream classifier C has a CRF output layer and is trained with a CRF loss L_C to maximize the log probability of the correct tag sequence (Lample et al., 2016). In our sentence classification experiments, C is a multinomial classifier with cross-entropy loss L_C. Let θ_R, θ_D, θ_C be the parameters of the representation module, discriminator, and downstream classifier, respectively. Gradient reversal training will update the parameters as follows:

$$\begin{aligned}
\theta_D &= \theta_D - \eta\lambda\frac{\partial L_D}{\partial \theta_D} \\
\theta_C &= \theta_C - \eta\frac{\partial L_C}{\partial \theta_C} \\
\theta_R &= \theta_R - \eta(\frac{\partial L_C}{\partial \theta_R} - \lambda\frac{\partial L_D}{\partial \theta_R})
\end{aligned} \tag{2.25}$$

with η being the learning rate and λ being a hyperparameter to control the discriminator influence.

We will focus on the creation of language- and domain-independent representations using adversarial training similar. This is related to the work of Gui et al. (2017), Liu et al. (2017a), Kasai et al. (2019), Grießhaber et al. (2020) and Zhou et al. (2019) who learned domain-independent representations with adversarial training. Kim et al. (2017) and Chen and Cardie (2018) worked with language-independent representations for cross-lingual transfer. Other applications include the removal of biases from a dataset or model (Barrett et al., 2019).

2.5 Summary

This chapter provided an overview of the most important concepts required for the understanding of the following chapters in this thesis. We have given a broad overview of the different information extraction tasks that will be addressed in the next chapters.

Our models for these tasks will be based on recurrent neural networks or transformers, or both, in combination with conditional random field output layers. For each of these model types, we have seen their advantages for processing sequential data by incorporating and modeling dependencies between individual tokens.

Further important concepts introduced in this chapter are pre-trained word embeddings and transformer-based input representations. We showed how these models are able to learn general knowledge from unstructured texts, and we dived into related work on self-supervised language modeling for the domain- and language-specific pre-training. In particular, domain-specific language models will be important for this thesis in Chapter 3 and Chapter 4. Moreover, multilingual LMs will be used in Chapter 5 and Chapter 8 for multilingual and cross-lingual applications.

Finally, this chapter gave a description of three different multi-task training variants that can be used to train robust models by enforcing generalizable feature representations based on joint or sequential learning of multiple tasks. Alternatively, adversarial training was introduced that enables the unsupervised removal of language- or domain-specific information from feature representations.

The following chapters will show how we use these architectures and training routines in our newly proposed methods in the context of information extraction in low-resource languages and non-standard text domains.

Chapter 3

Anonymization and Clinical Concept Extraction

This chapter will introduce various methods and models for information extraction in the clinical domain to extract the structured information contained in documents, such as patient reports of trial protocols. However, the processing of clinical documents requires proper de-identification, i.e., the anonymization of personal information in texts. Therefore, we propose a sequence-labeling model for the anonymization of texts that incorporates domain-specific clinical knowledge from specialized embeddings. Anonymization is most often only a pre-processing step and not the actual target task. However, current research considers anonymization and downstream tasks, such as concept extraction, only in isolation and does not study the effects of anonymization on other tasks. In this chapter, we also close this gap and show that anonymization even has a slightly positive impact on the performance of a concept extraction. Furthermore, we propose two joint models and achieve state-of-the-art performance on benchmark datasets in English (both tasks) and Spanish (concept extraction). Finally, we introduce a pipeline for the normalization of extracted clinical concepts to standardized clinical codes by linking the concepts to a knowledge base. The models and experiments described in this chapter are based on publications for anonymization (Lange et al., 2019a), for joint concept extraction and anonymization (Lange et al., 2020b) and for ICD coding (Lange et al., 2020c).

3.1 Introduction

In the clinical or biomedical domain, natural language processing has large potential to significantly improve the efficiency and effectiveness of processes, e.g., the extraction of structured information from clinical narratives can help in clinical decision making or drug repurposing (Marimon et al., 2019). A better understanding of this information can also facilitate novel clinical studies on the one hand, and help practitioners to optimize clini-

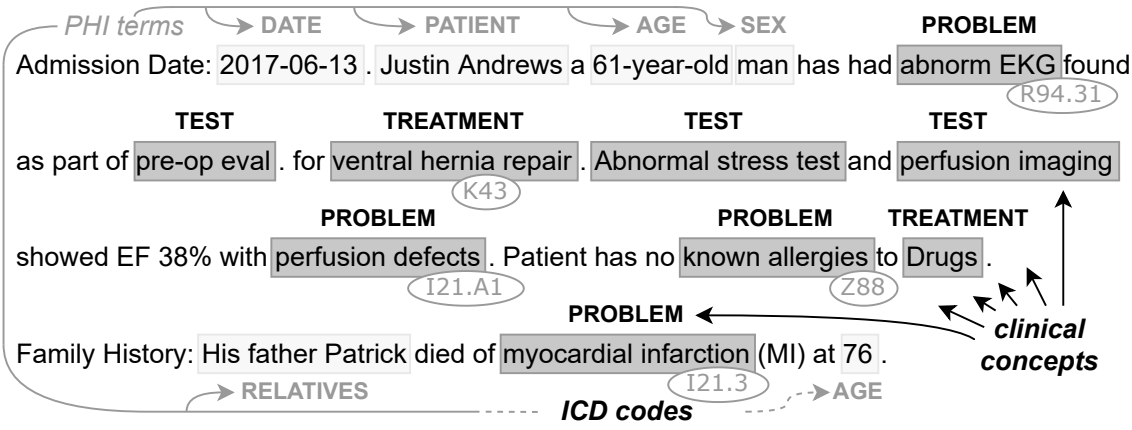


Figure 3.1: Example document from the clinical domain with annotations for anonymization and clinical concepts and their ICD codes.

cal workflows on the other hand. For example, to improve clinical decision support and personalized care of cancer patients, Jensen et al. (2017) developed a methodology to estimate disease trajectories from electronic health records (EHRs), which can predict 80% of patient events in advance. However, free text is ubiquitous in EHRs. This leads to great difficulties in harvesting knowledge from them. Therefore, natural language processing systems, especially information extraction components, play a critical role in extracting and encoding information of interest from clinical narratives, as this information can then be fed into downstream applications. Nonetheless, the automatic processing of documents with privacy-sensitive content like patient reports is restricted due to the necessity of applying anonymization techniques.

Text anonymization, also called de-identification, aims at detecting and replacing protected health information (PHI),[1] such as patient names or personal information, as shown in the upper part of Figure 3.1. Recent studies show that automatic anonymization leads to promising results (Uzuner et al., 2007; Stubbs et al., 2015). Therefore, we participated in the MEDDOCAN shared task on medical document anonymization in Spanish (Marimon et al., 2019) to show the prospects of our domain-robust model architecture for a non-standard domain and language. In this chapter, we will describe our models that have won the competition in more detail.

A severe limitation of current approaches, however, is that anonymization is typically addressed in isolation but not together with a downstream task, such as concept extraction (CE) from medical texts (Uzuner et al., 2011; Gonzalez-Agirre et al., 2019). Instead, the downstream task models are trained and evaluated on the non-anonymized data, and it remains unclear how anonymization affects their performance. In this chapter, we argue that to evaluate how promising NLP is in the medical domain, the tasks of anonymization and information extraction should be analyzed together. Therefore, we close this gap and analyze the effect of anonymization on clinical concept extraction. Moreover, we consider

[1] PHI types are typically defined by governments, for instance in the Health Insurance Portability and Accountability Act (HIPAA) of the United States.

the two tasks jointly and propose two end-to-end models: A multi-task model that shares the input representation across tasks and a stacked model that trains a differentiable pipeline of anonymization and concept extraction in an end-to-end manner. For the stacked model, we propose to use a masked embedding layer to restrict the access of the concept detector to privacy-sensitive information and train it on an anonymized version of the data. To make the model differentiable, we use the Gumbel softmax trick (Maddison et al., 2017; Jang et al., 2017).

We conduct experiments on clinical benchmark datasets in English and Spanish. Our results indicate that anonymization does not affect concept extraction models negatively but has even a slight positive effect on the results, probably because anonymization homogenizes the input for concept extraction. Modeling both tasks jointly leads to better results than treating anonymization as a pure pre-processing step.

After successfully extracting clinical concepts from documents, they may be normalized to specific elements in a pre-defined taxonomy. For clinical concepts, these are most often ICD codes, as introduced in Section 2.1.2. This chapter will also describe our approach for the normalization of oncological concepts to the ICD-O taxonomy, also referred to as eCIE-O-3.1 in Spanish. We address this task in the context of the Spanish CANTEMIST shared task (Miranda-Escalada et al., 2020). Examples for ICD codes from an English document are given in Figure 3.1.

3.2 Related Work

In this section, we report on related work in the fields of anonymization, medical concept extraction, and multi-task learning.

Anonymization. The increasing importance of anonymization is reflected in the number of shared tasks (Uzuner et al., 2007; Stubbs et al., 2015; Marimon et al., 2019). State-of-the-art methods for anonymization typically rely on recurrent neural networks (Dernoncourt et al., 2017; Kajiyama et al., 2018).

Feutry et al. (2018) and Friedrich et al. (2019) create pseudo-de-identified text representations with adversarial training. In particular, they replace personal information, such as names, with other names. Zhao et al. (2018) augment the training data by creating more general text skeletons, e.g., by replacing rare words, such as names, with a special unknown token. Compared to these works, we use a trade-off and replace personal information by their class names as placeholders in the joint anonymization and concept extraction. This approach is not only common for anonymization (Johnson et al., 2016), but also for relation extraction where entities are often either replaced by their type or enriched with type information (i.a., Miwa and Sasaki, 2014; Gui et al., 2017). We further motivate our choice in Section 3.3.3. Another difference to the above-mentioned works is that we do not augment the training data for our anonymization model.

Medical Information Extraction. Analogously, there have been a series of shared tasks for information extraction in the clinical and biomedical domain (Uzuner et al., 2011; Xu et al., 2012; Pérez-Pérez et al., 2017; Gonzalez-Agirre et al., 2019). Models for these tasks either rely on hand-crafted features (Leaman et al., 2015b; Xu et al., 2012) or RNNs (Hemati and Mehler, 2019; Korvigo et al., 2018; Tourille et al., 2018). Newman-Griffis and Zirikly (2018) study the performance of RNNs for medical named entity recognition in the context of patient mobility and find that they benefit from domain adaption.

In contrast to previous work, we investigate the usage of anonymized texts as input for clinical concept extraction models and propose to model anonymization and concept extraction jointly.

Concept Normalization and ICD Coding. Several machine-learning-based named entity recognition and normalization systems were implemented To identify and normalize medical concepts within the clinical narratives in EHRs. DNorm, introduced by Leaman et al. (2013), applied a pairwise learning-to-rank approach to automatically learn a mapping from disease mentions to disease concepts from the training data. Evaluation results show that the machine-learning method can effectively model term variations and achieves much better results than traditional techniques based on lexical normalization and matching, such as MetaMap (Aronson, 2001). Leaman et al. (2015a) introduced an extension of DNorm, called DNorm-C, which approaches both discontinuous NER and normalization using a pipeline approach. A joint model for NER and normalization was introduced by Leaman and Lu (2016), aiming to overcome the cascading errors caused by the pipeline approach and enable the NER component to exploit the lexical information provided by the normalization component. Zhao et al. (2019) proposed a deep neural multi-task learning method to jointly model NER and normalization from biomedical publications, where stacked recurrent layers are shared among different tasks, enabling mutual support between tasks. Similarly, Lou et al. (2017) proposed a transition-based model to jointly perform disease NER and normalization, combined with beam search and online structured learning.

In contrast to pure concept normalization, which identifies a one-to-one mapping between text snippets and medical concepts, ICD coding assigns the most relevant ICD codes to a document as a whole (Pestian et al., 2007; Névéol et al., 2018). Most previous methods simplified this task as a text classification problem and built classifiers using CNNs (Karimi et al., 2017) or tree-of-sequences LSTMs (Xie et al., 2018). Since ICD codes are organized under a hierarchical structure, Mullenbach et al. (2018) and Cao et al. (2020a) proposed models to exploit code co-occurrence using label attention mechanism and graph convolutional networks, respectively.

Multi-task Learning. While many works propose joint training for other NLP tasks (i.a., Finkel and Manning, 2009; Miwa and Sasaki, 2014), including multi-task learning (i.a., Collobert and Weston, 2008; Klerke et al., 2016; Søgaard and Goldberg, 2016) and stacking of pipeline components (i.a., Miwa and Sasaki, 2014), we are to the best of our knowl-

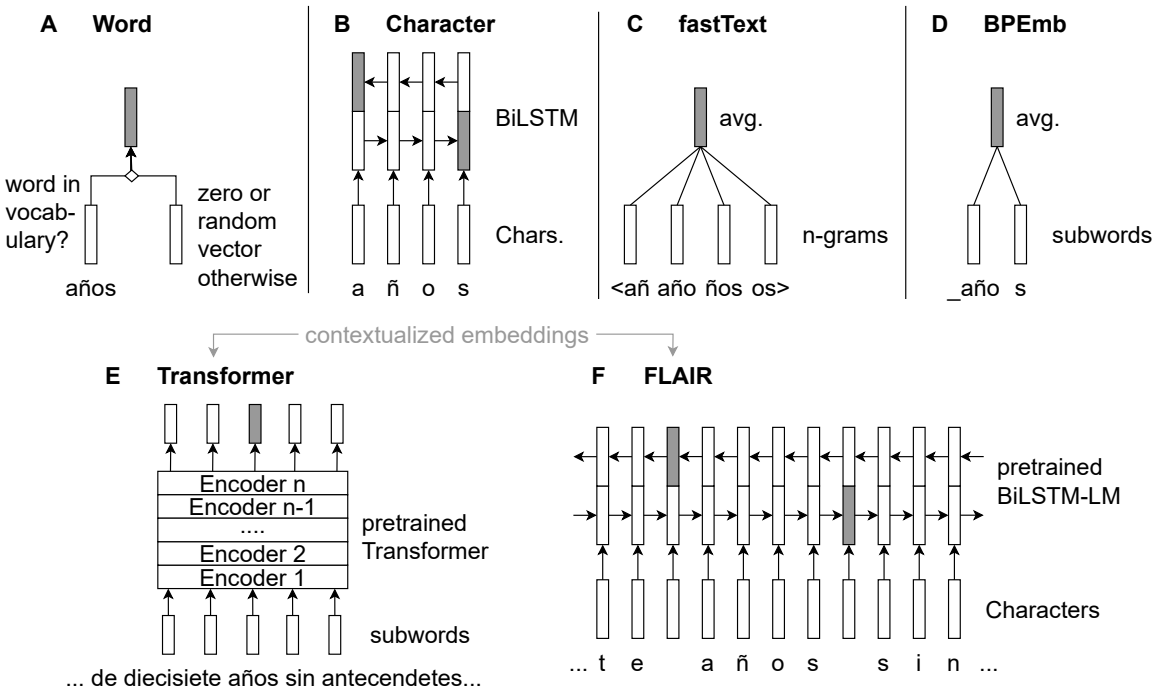


Figure 3.2: Overview of embedding methods, including five subword embeddings. The shaded vectors are used to represent the input token. Example Sentence: "Pedro de diecisiete años sin antecedentes ..." (*en*: Seventeen-year-old Pedro with no criminal record...).

edge the first to combine anonymization with information extraction tasks. More general information on multi-task learning are given in Section 2.4 of this thesis.

3.3 Model Architectures

In this section, we describe our model architectures for the different anonymization and concept extraction experiments in more detail. First, we introduce the different input representations used in our models in this and the other chapters of this thesis (Section 3.3.1). Then, we describe the model architectures for single-task learning (Section 3.3.2), the joint learning of multiple tasks (Section 3.3.4), as well as our pipeline for ICD coding (Section 3.3.5).

3.3.1 Input Representations

For all neural models in this chapter, each token is represented with a combination of different pre-trained language-specific embeddings. These can be either standard word2vec embeddings as introduced in Section 2.3.1 or more advanced subword-based embeddings as depicted in Figure 3.2. In the following section, each of these subword embeddings will be introduced in more detail.

Character Embeddings. The characters of a word are represented by randomly initialized embeddings. Those are passed to a bi-directional long short-term memory network (BiLSTM). The last hidden states of the forward and backward pass are concatenated to represent the word (Lample et al., 2016). These embeddings can generate a representation for any input word as long as its characters are covered in the character embedding vocabulary. Unknown characters, e.g., infrequent Unicode symbols, are replaced by a special symbol. Note that in contrast to all following embeddings, the character embeddings are not pre-trained, and they have to be learned during training. Therefore, they can learn task-specific representations but do not contain general knowledge from a larger-scale pre-training.

FastText Embeddings. The fastText embeddings represent a word by the normalized sum of the embeddings for the n-grams of the word (Bojanowski et al., 2017). More precisely, fastText embeddings are CBOW embeddings trained over n-grams instead of words. For more information on CBOW we refer to Section 2.3. We experiment with domain-independent fastText embeddings (300 dimensions, pre-trained on language-specific texts (Grave et al., 2018)), as well as domain-specific fastText embeddings for English (100 dimensions, pre-trained on English PubMed articles (Pyysalo et al., 2013)) and Spanish (100 dimensions, pre-trained on Spanish SciELO and Wikipedia articles (Soares et al., 2019)). In contrast to standard word2vec embeddings with a fixed vocabulary, the fastText embeddings can generate representations for out-of-vocabulary (OOV) words by splitting the word into known n-grams. Note that Grave et al. (2018) also published fastText embeddings with a fixed vocabulary and without out-of-vocabulary functionality, which we will use in Chapter 5 to analyze different effects of subword embeddings.

Byte-pair encoding Embeddings. Similar to fastText embeddings, byte-pair encoding embeddings (BPEmb, Heinzerling and Strube, 2018) are generated by averaging all subword embedding vectors of a word. For this, the subword segmentation is performed by a trained byte-pair encoding (BPE) model that splits words into subwords. In contrast to fastText that uses fixed-sized and overlapping n-grams, the BPE model incorporates subword frequencies from a large-scale training corpus and merges longer characters sequences for frequent subwords while having a fallback to single characters for infrequent or unseen subwords.

Flair Embeddings. Flair computes character-based embeddings for each word depending on all words in the context (Akbik et al., 2018). For this, the complete sentence is used as the input to a BiLSTM character model instead of only a single word as done for standard character embeddings. The BiLSTM of Flair is pre-trained using a character-level language model objective, i.e., given a sequence of characters, compute the probabilities for the following possible characters. These language models are unidirectional and capture only the previous context. Therefore, these embeddings are also trained in the reverse

direction, which captures the future context of words. Both directions are then combined into a single embedding for each word.

Transformer Embeddings. Pre-trained transformer models can also be used to retrieve word embeddings. For this, the output of the last layer or a combination of multiple layer outputs can be used as a contextualized embedding vector that can be further processed by, e.g., a recurrent neural network as done in our experiments. More information on transformer models and their pre-training is provided in Chapter 2. In our setup, we combine the outputs of the last four layers of transformer models by concatenating the vectors. Moreover, the transformer can optionally be fine-tuned. Typically, we do not fine-tune the transformer models when it is used to generate embedding vectors in order to reduce computational costs. However, we explicitly mention whenever we perform this fine-tuning otherwise. In this chapter, we are going to use the general-domain pre-trained BERT model, as well as the domain-specific ClinicalBERT (Alsentzer et al., 2019) for English tasks and multilingual BERT (mBERT, Devlin et al., 2019) for Spanish. We are going to use different transformer models in later chapters.

Combinations of Multiple Embeddings. Previous work has seen performance gains by, for example, combining various types of word embeddings (Tsuboi, 2014) or embeddings trained on different corpora (Luo et al., 2014) as these embeddings can have unique knowledge and properties. We follow these approaches and combine multiple embeddings to leverage all of their benefits. For this, the n different embedding vectors are concatenated into a single embedding vector $e(i)$ for word i where $[\cdots]$ is the concatenation operation of the individual embeddings e_n:

$$e_{CONCAT}(i) = [e_1(i); \cdots ; e_n(i)] \tag{3.1}$$

Whenever we use multiple embeddings, we explicitly mention which embeddings are used in our models. These are always combined with the concatenation if not noted otherwise, for example, in Chapter 5 where we explore the concept of attention-based meta-embeddings.

3.3.2 Sequence-Labeling Models for Single Tasks

We model both document anonymization (ANON) and clinical concept extraction (CE) as sequence-labeling problems and apply a bidirectional long short-term memory network with a conditional random field output layer similar to Lample et al. (2016). We refer to this architecture as a BiLSTM-CRF. Detailed descriptions of BiLSTM and CRF models are provided in Chapter 2. In recent works on clinical anonymization and concept extraction, this architecture was shown to be very promising (Marimon et al., 2019; Gonzalez-Agirre et al., 2019).

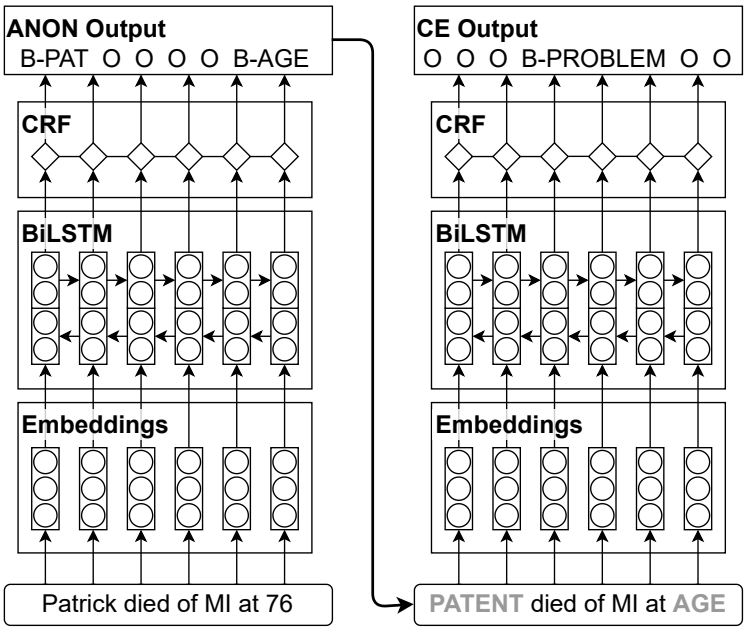


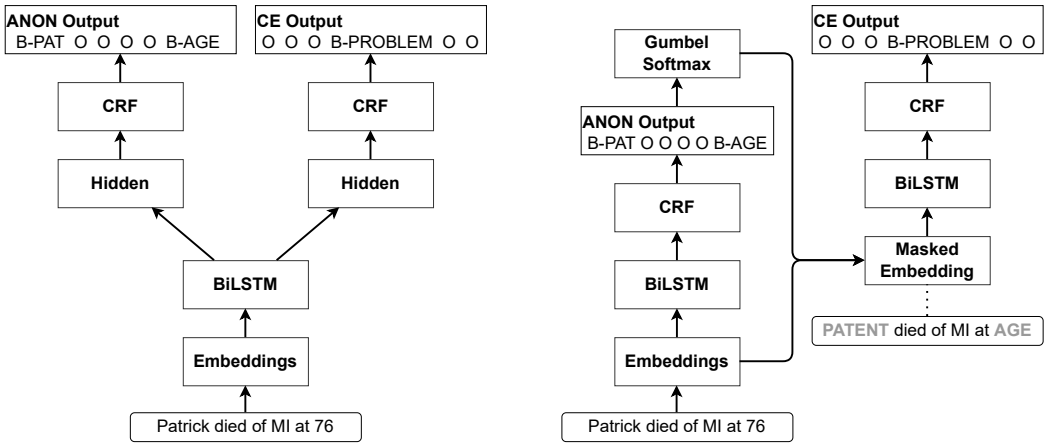
Figure 3.3: Overview of the pipeline model for anonymization and concept extraction.

Depending on the experimental setup, we focus on different embeddings. For this, we concatenate all embeddings as described in Section 3.3.1 and fed the resulting vector into the BiLSTM network to generate a contextualized feature representation f given the embeddings e for each word in the sentence. The features f are then mapped to the size of the label space through a linear layer and fed into a CRF classifier that computes the most probable sequence of labels. We found that a single LSTM layer with a hidden size of 256 units worked best in our experiments.

Biaffine Classifier. More recently, a trend emerged of modeling different natural language processing tasks as parsing tasks and thus, solving them by using a dependency parser. Examples are named entity recognition (Yu et al., 2020) and negation resolution (Kurtz et al., 2020).

We experiment with such a system and model the extraction task as a parsing task. For this, we replace the CRF classifier with a biaffine classifier (Dozat and Manning, 2017). Following Yu et al. (2020), we apply two separate feed-forward networks (FFNN) to the features f generated from the stacked BiLSTM to create start and end representations of all possible spans (h_s/h_e). Then, we use biaffine attention (Dozat and Manning, 2017) over the sentence to compute the scores r_m for each span i in the sentence that could refer to a named entitiy.

$$\begin{aligned} h_s(i) &= FFNNs(f_{s_i}) \\ h_e(i) &= FFNNe(f_{e_i}) \\ r_m(i) &= h_s^\top(i) U_m h_e(i) + W_m(h_s(i) \circ h_e(i)) + b_m \end{aligned} \tag{3.2}$$



(a) Multi-task Model. **(b)** Stacked Model.

Figure 3.4: Overview of our multi-task model architectures for anonymization and concept extraction. Note that the concept extraction part in the stacked model has no access to privacy-sensitive information. "PAT" stands for Patient. The labels are encoded in the BIO format.

3.3.3 Pipeline Models for Multiple Tasks

Most often, the automatic processing of clinical documents requires the anonymization of texts to remove privacy-sensitive information in a first step, and two independent models are trained. To assess the effects of anonymization on concept extraction, we first apply the anonymization model to anonymize the concept extraction dataset and then evaluate the concept extraction model on the anonymized data. We refer to this approach as PIPELINE model (see Figure 3.3). For anonymization, we replace each detected privacy-sensitive term with a placeholder of its PHI type, i.e., there is one placeholder per type.

This replacement choice has advantages over the alternatives described in Section 3.2. Compared to replacing personal information with alternative names, it leads to a more general text and thus, homogenizes the input for the downstream-task classifier. Compared to replacing all personal information with the same token, the resulting text is more specific, allowing the downstream-task classifier to take into account which kind of personal information was mentioned. Thus, the approach is a trade-off between more homogeneous input and more fine-grained information for the downstream-task classifier.

3.3.4 Joint Models for Anonymization and Concept Extraction

Instead of using a sequential pipeline, we propose to jointly train both tasks. For this, we test two approaches: a multi-task model and a stacked model.

Multi-Task Model. In the MULTI-TASK model (Figure 3.4a), the weights up to the BiLSTM layer are shared across both tasks. For each task, we add a task-specific hidden layer

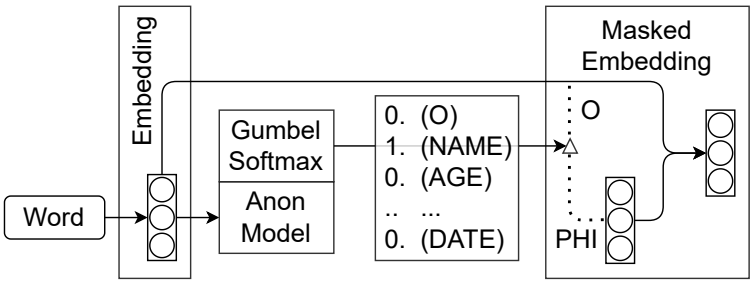


Figure 3.5: Structure of the masked embedding layer based on the gumbel softmax.

with ReLU activation and a CRF output layer. Note that in this architecture, the concept extraction model has access to the original, privacy-sensitive data.

Stacked Model. We also propose a STACKED model (Figure 3.4b), where only the anonymization part has access to the privacy-sensitive information. This can be seen as a differentiable version of the pipeline model, where the input and access to privacy-sensitive information to the concept extraction part is restricted by a **masked embedding layer**. This layer ensures that the concept extraction model does not have access to privacy-sensitive information by replacing the input embeddings of privacy-sensitive tokens with PHI-class embeddings, which are randomly initialized and fine-tuned during training. This is depicted in Figure 3.5. The masked embedding layer requires a discrete output from the anonymization part. In order to ensure that the model stays fully differentiable, we use the **Gumbel softmax trick** (Maddison et al., 2017; Jang et al., 2017). It approximates categorical samples with a continuous distribution on the simplex and computes gradients for backpropagation with the reparameterization trick. The Gumbel softmax function has the following form:

$$y_k^\tau = \frac{\exp((\log \alpha_k + G_k)/\tau)}{\sum_{i=1}^{K} \exp((\log \alpha_i + G_i)/\tau)} \tag{3.3}$$

with $\alpha_1, ...\alpha_K$ being the unnormalized output scores from the anonymization layer and $G_1, ..., G_K$ being i.i.d samples drawn from Gumbel(0, 1) and τ being a temperature. For $\tau \to 0$, the distribution becomes identical to the categorical distribution.

3.3.5 ICD Coding Pipeline

Regardless of the extraction model, the clinical concepts might be normalized to standardized clinical codes and linked to a knowledge base. Our normalization pipeline is visualized in Figure 3.6. We will describe the subtasks of normalization and raking in more detail as used in the context of the CANTEMIST shared task (Miranda-Escalada et al., 2020).

Concept Normalization. As a large number of possible ICD codes appear only once or never in the training data, we decided against deep learning methods for the normalization,

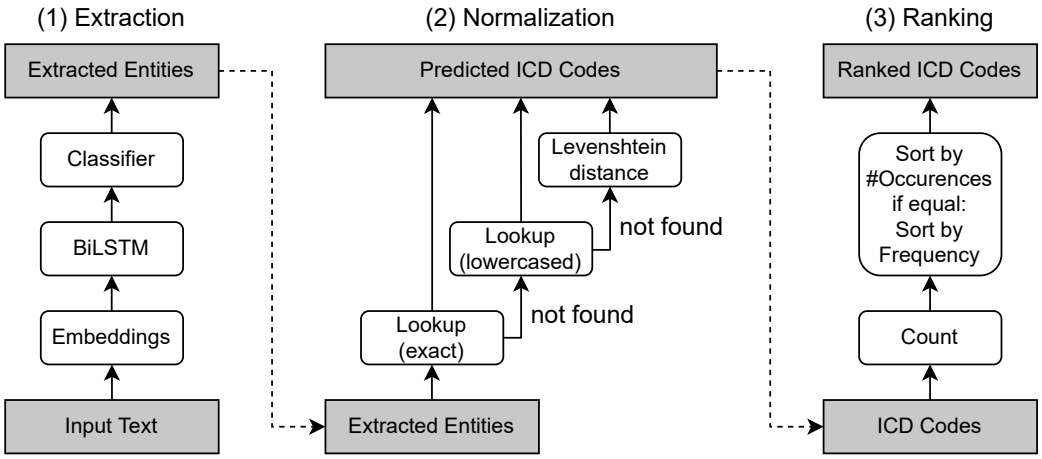


Figure 3.6: Overview of our architecture for ICD coding.

as simply not enough training instances are available for this large label set. Instead, we use an approach based on string matching and Levenshtein distance (Levenshtein, 1966).

For this, we collect all entities from the training set and their ICD code. As there is only little ambiguity among these entities, we use a context-independent method for the normalization. Using the entities from the training set, we are able to correctly assign 70% of the ICD codes to entities from the development set using exact string matching with a low false-positive rate ($< 1\%$). Using lower-cased matching, the number of correctly assigned codes slightly increases. Given that these methods assign ICD codes almost perfectly to known entities, we first apply exact string matching and then lower-cased matching. For the remaining unmatched entities, we compute the Levenshtein distance between the given string and strings from the training data to find the closest neighbor among the known training instances and assign the corresponding code. This method achieves 87% F_1 on the gold extractions of the unseen development set.

ICD Coding. In contrast to pure concept normalization, which assigns one ICD code per extraction, ICD coding refers to the creation of a ranked list of ICD codes for a given document. For example, the ICD code `I21.3` referring to extracted concept *myocardial infarction* as shown in Figure 3.1 is more important for the document than `Z88` (known allergies). In practice, we create a ranking with a sorting function based on code frequencies. We sort by the number of times each code occurs in the given document under the assumption that codes that appear more often inside a document are more important. Whenever two codes appear an equal amount of times, they are ranked by their general frequency as found on the training set. This method achieves a mean average precision (as introduced in Section 3.7.1) of 73.82 using the gold extractions of the unseen development set.

	English (I2B2)		Spanish		
	ANON	CE	ANON	CE	Coding
# classes	25	4	23	4	2
Train (# sentences)	45,793	16,315	15,903	8,068	19,416
Dev (# sentences)	5,088	-	8,277	3,748	18,156
Test (# sentences)	32,587	27,625	7,966	3,930	11,185

Table 3.1: Overview of the dataset statistics. # classes denotes the number of classes including the neutral class *O*.

3.4 Experimental Setup

In this section, we describe the datasets and model configurations we use in our experiments, as well as the training process. More specifically, we perform three distinct sets of experiments. First, we conduct anonymization experiments with domain-specific embeddings. Second, we explore the joint modeling of anonymization and concept extraction. Finally, we perform concept normalization and ICD coding in Spanish.

3.4.1 Datasets and Pre-Processing

We evaluate our models on corpora from the clinical domain in English and Spanish. For English, we use the data from the I2B2-2010 concept extraction task (Uzuner et al., 2011) and the I2B2-2014 anonymization challenge (Stubbs and Uzuner, 2015). For Spanish, we use the MEDDOCAN (Marimon et al., 2019) corpus for anonymization and the PHARMACONER corpus (Gonzalez-Agirre et al., 2019) for concept extraction. As PHARMACONER is a subset of MEDDOCAN, we have both gold-standard concept and anonymization annotation for this data, which we will use for further analyses. The ICD coding experiments are performed on the CANTEMIST corpus. Table 3.1 shows statistics on the dataset sizes and the number of labels. We report the F_1-score for exact matching in all experiments and add additional metrics depending on the task.

We use the pre-processing scripts from Alsentzer et al. (2019) for the English I2B2 corpora and the Spanish Clinical Case Corpus tokenizer (Intxaurrondo, 2019) for all Spanish corpora. We noticed that the Spanish tokenizer sometimes merges multi-word expressions into a single token joined with underscores for contiguous words. As a result, some tokens cannot be aligned with the corresponding entity annotations. To address this, we split those tokens into their components in a post-processing step.

3.4.2 Setup for Anonymization

We employ BiLSTM-CRF models for anonymization and investigate the effect of domain knowledge contained in the embeddings. For this, we use character embeddings or the Flair character language models, as well as domain-specific or general domain fastText

embeddings. Preliminary experiments revealed that a single LSTM layer with a hidden size of 256 units performs well in our experiments.

3.4.3 Setup for Joint Anonymization and Concept Extraction

In addition to the embeddings used in the previous model (domain-specific fastText, Flair, and character embeddings), we include byte-pair encoding embeddings and BERT embeddings. The BERT embeddings have 768 dimensions and are constructed by averaging the last four layers. As described in Section 3.3.1, we concatenate all embeddings into one input vector, resulting in a total input dimensionality of 5,416 for English and 4,748 for Spanish. For the LSTM, we use 256 hidden units per direction. The task-specific hidden layer of the multi-task model has 128 units. Note that we use the same hyperparameters for all our models and all tasks.

3.4.4 Setup for ICD Coding

For the extraction and normalization, we employ either BiLSTM-CRF models as described before or alternatively experiment with the biaffine classifier. The CRF-based models are similar to the models for anonymization and additionally contain byte-pair encoding embeddings. For the biaffine model, we use multilingual BERT, character, and fastText embeddings following Yu et al. (2020). We experimented with the same set of embeddings that we used for the BiLSTM-CRF model, but the performance decreased for the biaffine model. We found that five stacked LSTM layers with a size of 200 hidden units each worked best for the biaffine model. Using this combination of hyperparameters improved the model by roughly one F_1 point compared to a model consisting of 3 layers of size 200. For the remaining hyperparameters, we follow the previous settings.

3.4.5 Training

For training, we use stochastic gradient descent with a learning rate of 0.1 and a batch size of 32 sentences for all models. The learning rate is halved after three consecutive epochs without improvement on the development set. For the joint models, we pre-train the anonymization part for three epochs and, then, use a higher initial learning rate of 0.2 for the concept extraction part. We perform early stopping on the hold-out development set or create a new development set with 10% of the training set size whenever no development set was provided (I2B2-2010 corpus) or the original training and development set were combined.

System	Task 1: NER				Task 2: ST		
	Leak	Precision	Recall	F_1	Precision	Recall	F_1
S_1 (*Char+fastText +Domain*)	0.02432	96.956	96.767	96.861	97.522	97.333	97.427
S_2 (*Flair +fastText*)	0.02378	**97.078**	96.838	96.958	**97.574**	97.333	97.453
S_3 (*Flair +fastText +Domain*)	**0.02299**	96.978	**96.944**	**96.961**	97.508	**97.474**	**97.491**
Hassan et al. (2019)	0.03255	96.991	95.672	96.327	97.529	96.202	96.861
Pérez et al. (2019)	0.03282	96.403	95.637	96.018	97.187	96.414	96.799

Table 3.2: Results for Task 1 (Offset and Type Classification) and Task 2 (Sensitive Token Detection). The main metric (F_1) is highlighted.

3.5 Results for Anonymization

This section describes our results and analysis for our anonymization models for the MEDDOCAN shared task. We report the results on the test set using the official shared task evaluation measures (Marimon et al., 2019).

Evaluation Metrics. The main evaluation measure within this chapter is the F_1-score for all tasks, which is the most common metric for anonymization and other named entity recognition tasks. We also report precision and recall.

In addition, a leak score can be computed for anonymization to measure the number of remaining PHI terms after anonymization in proportion to the document length as follows:

$$\text{Leak} = \frac{\text{False Negatives}}{\text{\#Sentences}} \tag{3.4}$$

3.5.1 Results

In the first sub-task, the systems need to find spans for anonymization and categorize them into 29 classes. Table 3.2 presents our results on this sub-task.

While the domain-independent system (S_2 with Flair and domain-independent fastText embeddings) leads to the highest recall values, the third run (S_3) that also uses domain-specific fastText embeddings achieves the highest F_1-scores. This shows that integrating domain knowledge into the token representation is beneficial. However, the differences among the settings are relatively small, indicating that the architecture itself is already strong enough for the given dataset, and the impact of different input representations is minor.

Table 3.2 also provides the results of our models on the second sub-task (sensitive token detection). In contrast to task 1, this is a binary classification task. The ranking of our models is the same for sub-task 1: the addition of domain-specific input representations performs best. In both sub-tasks, Flair embeddings outperform standard character embeddings.

	Predicted Label																						
	O	CALLE	CS	MAIL	EDAD	FAM	FECHA	HOS	ID_AS	ID_CON	ID_EPS	ID_SUJ	ID_TPS	INST	NOM_PS	NOM_SA	#FAX	#TEL	OTRO	PAIS	PROF	SEXO	TER
O	123293	21			2	23	9	3		1		4	6	28	3		1			2		3	
CALLE	15	2997						2						1	8	1							6
CS			8					3															
MAIL				256											3								
EDAD	14				1014	3																	
FAM	18				2	104						2							2				
FECHA	16						1089																
HOS	10	4					4	551						11									1
ID_AS									573			2	6										
ID_CON										32													
ID_EPS	4																						
ID_SA	9											293									2		
ID_TPS													663									1	
INST	35	8	3					11						190									
NOM_PS	3					1									1585	2							
NOM_SA	3															779							
#FAX																	16						
#TEL	1																	70					1
OTRO	11					2						1							2				
PAIS																				349			
PROF	8																				1		
SEXO																						456	
TER	9	20					1	5						5						3			1141

Table 3.3: Confusion matrix of the best anonymization model (S_3) on the development set.

The performance of the second-best and third-best systems out of 18 participants in the shared task is also given in Table 3.2. We see that our system outperforms these submissions as well and won both subtracks of the competition. More information on the other participant systems and further results are given by Marimon et al. (2019).

3.5.2 Analysis of Anonymization Model

In the following, we provide a more detailed error analysis of our anonymization model and a case study on synthetic data augmentation, as parts of the data were automatically generated.

Confusion Matrix Analysis. Table 3.3 shows the confusion matrix of our best performing system (run S_3).[2] It is similar to the identity matrix, i.e., confusions between classes happen very rarely. The most confusions happen with O, the label assigned to all non-PHI terms, which might be caused by the high number of occurrences of this class in the

[2] Abbreviations for entity types:

CALLE (CALLE), CENTRO_SALUD (CS), CORREO_ELECTRONICO (MAIL), EDAD_SUJETO_ASISTENCIA (EDAD), FAMILIARES_SUJETO_ASISTENCIA (FAM), FECHAS (FECHA), HOSPITAL (HOS), ID_ASEGURAMIENTO (ID_AS), ID_CONTACTO_ASISTENCIAL (ID_CON), ID_EMPLEO_PERSONAL_SANITARIO (ID_EPS), ID_SUJETO_ASISTENCIA (ID_SA), ID_TITULACION_PERSONAL_SANITARIO (ID_TPS), INSTITUCION (INST), NOMBRE_PERSONAL_SANITARIO (NOM_PS), NOMBRE_SUJETO_ASISTENCIA (NOM_SA), NUMERO_FAX (#FAX), NUMERO_TELEFONO (#TEL), OTROS_SUJETO_ASISTENCIA (OTRO), PAIS (PAIS), PROFESION (PROF), SEXO_SUJETO_ASISTENCIA (SEXO), TERRITORIO (TER)

training dataset. Confusions among PHI classes happen mostly between related classes. For example, Hospital (HOS) and Institution (INST) are confused quite often, as Hospital is a subclass of Institution, and other medical institutions are tagged with Hospital, and vice versa, e.g., *Clinica Gnation* is an institution tagged as a hospital. Analogously, Streets (CALLE) and Territories (TER) are getting confused often, as both classes are related and typically constitute multiple tokens. In contrast to this, Countries (PAIS) are tagged correctly almost every time, as there is only a very limited number of countries, and they are usually single token expressions.

Synthetic Augmentation Case Study. As mentioned above, the performance difference between our systems is relatively small. This may be caused by the synthetic augmentation of the MEDDOCAN data, which was used to extend the texts with header and footer information containing many PHI terms. In fact, 85% of PHI terms appear in the augmented text parts. While this extension is necessary to cover more classes and PHI terms, the synthetic nature of these extensions may have an impact on the performance of automatic classifiers. Therefore, we perform a case study in which we remove these parts from the test set and compare only the predictions found in the real texts. With this, only 838 out of 5661 (14.8%) annotations and only 13 out of 29 classes remain in this experiment. The performances of our systems are decreased to F1 scores around 90, which is still rather high. This shows that our systems have learned more than just to reproduce the synthetic data augmentation. However, the performance differences among our systems are still small, indicating that the data augmentation was not the reason for this behavior. Note, however, that we did not retrain our models without the synthetic augmentation.

3.6 Results for Joint Anonymization and Concept Extraction

In this section, we will analyze the effect of anonymization on clinical concept extraction based on two end-to-end models for joint anonymization and concept extraction.

3.6.1 Results

The results for our anonymization methods and concept extraction models are given in Tables 3.4a and 3.4b, respectively. For anonymization, Table 3.4a shows that our anonymization model performs comparable to the current state-of-the-art for anonymization in English and Spanish. The anonymization performance of STACKED is comparable to the original anonymization model (En: 95.9, Es: 96.8), the anonymization performance of MULTI-TASK is slightly lower (En: 95.2, Es: 96.7).

The results for our CE models in comparison to state of the art are shown in Table 3.4b. We outperform the current state of the art in both languages. While PIPELINE, i.e., the application of the concept extraction model on anonymized text, is slightly worse (as it has

Models	English	Spanish
Yang and Garibaldi (2015)	96.0	
Alsentzer et al. (2019)	93.0	
Hassan et al. (2019)		96.3
OUR MEDDOCAN model (Sec. 3.3)		**97.0**
OUR (anonymization only)	**96.1**	96.8
OUR STACKED	95.9	96.8
OUR MULTI-TASK	95.2	96.7

(a) Anonymization.

Models	English	Spanish
de Bruijn et al. (2010)	85.2	
Alsentzer et al. (2019)	87.7	
Sun and Yang (2019)		89.2
OUR (CE only)	88.1	89.7
OUR PIPELINE	88.0	89.6
OUR STACKED	88.7	90.0
OUR MULTI-TASK	**88.9**	**90.3**
Xiong et al. (2019)*		91.1
Our MULTI-TASK*		**91.4**

(b) Concept extraction.

Table 3.4: Results for joint anonymization (left) and concept extraction (right). * indicates models which are trained on a combination of training and development set.

Embedding	English	Spanish
fastText	81.5	78.7
byte-pair encoding	83.4	83.9
Flair	83.0	82.4
Multilingual BERT	84.4	85.9
Clinical BERT (English)	87.2	-
Clinical fastText (Spanish)	-	79.7
Concatenation of all	**88.1**	**89.7**

Table 3.5: Effects of different embeddings on the concept extraction tasks (without anonymization).

been trained on the original text), training anonymization and concept extraction jointly leads to considerable improvements for both stacked and multi-task. Especially the results of STACKED in comparison to PIPELINE show that end-to-end training of the two steps is promising in both languages.

The performance of each embedding used in our experiments is shown in Table 3.5. As mentioned before, we did not include multilingual BERT embeddings for English but show their results for completeness. We see that the contextualized embeddings (Flair and BERT) achieve the single-best performance. However, even these powerful embeddings benefit from the inclusion of other embeddings, as the concatenation of multiple embeddings delivers the best performance. In addition, the domain-specific variants for fastText (Spanish) and BERT (English) outperform their general domain counterparts with +1 and + 2.8 F_1, respectively.

3.6.2 Analysis of Pipeline Setting

Finally, we analyze the impact of anonymization on concept extraction.

	Train on	Test on	Dev F_1	Test F_1
(i)	original	original	89.2	89.7
(ii)	original	predicted	89.1	89.6
(iii)	predicted	predicted	89.6	90.0
(iv)	gold	predicted	89.5	90.0

Table 3.6: Pipeline analysis results on Spanish concept extraction. "original": non-anonymized data, "gold"/"predicted": gold/predicted anonymization labels.

The results for training and testing our concept extraction model on different inputs (original vs. anonymized) are shown in Table 3.6. We restrict our analysis to Spanish since the data is labeled with both anonymization and concept information (see Section 3.4.1). Thus, we can also investigate the difference between gold and predicted labels. The CE model benefits from being trained on anonymized data (lines iii, iv). However, it decreases performance to train on non-anonymized data and evaluate on predicted anonymization labels (line ii). This supports our motivation that it is necessary to regard anonymization and downstream applications together. We assume that anonymization creates more homogeneous inputs for downstream tasks, such that sentences are more prototypical and abstract from irrelevant details for the biomedical concept extraction, such as personal information, which are always non-entities. The difference of training on gold vs. predicted anonymization labels (lines iii, iv) is only marginal, showing that state-of-the-art anonymization systems are good enough to be used in such settings.

3.7 Results for ICD Coding Pipeline

The official results for the three tracks of the CANTEMIST shared task are shown in Table 3.7. The official evaluation metric of the test set is highlighted in gray, and the best model is highlighted in bold.

3.7.1 Evaluation Metrics

The main evaluation metric for the extraction and normalization is again the F_1-score. As ICD coding can be seen as an information retrieval task, we follow the best practices and report the mean average precision metric as well. For this use case, precision uses a slightly different definition as introduced in Section 2.1.3 and is defined as the fraction of the retrieved ICD codes that are actually relevant to the document, i.e., it measures the precision given a list of retrieved ICD codes for a single document.

$$\text{Precision} = \frac{|\{\text{relevant ICD codes}\} \cap \{\text{retrieved ICD codes}\}|}{|\{\text{retrieved ICD codes}\}|} \tag{3.5}$$

Based on this definition, precision does not incorporate the ranking of codes and ignores that some documents are more important than others. Nonetheless, we want to use

	Task 1: NER			Task 2: NORM			Task 3: CODING			
	Pre.	Rec.	F_1	Pre.	Rec.	F_1	Pre.	Rec.	F_1	MAP
S_1 (CRF)	82.4	83.0	82.7	74.3	74.9	74.6	75.5	76.2	75.9	73.7
S_2 (Biaffine)	85.0	83.5	84.2	76.7	75.3	76.0	75.9	76.3	76.1	73.9
S_3 (Biaffine-Dev)	85.4	85.2	85.3	76.7	76.6	76.7	77.0	77.1	77.0	74.9
Xiong et al. (2020)	**87.1**	86.8	**87.0**	**82.4**	**82.6**	**82.5**	-	-	-	-
Pablos et al. (2020)	86.8	**87.1**	86.9	82.2	82.1	82.1	**87.5**	**83.6**	**85.5**	**84.7**

Table 3.7: Results of the three tasks: (1) The extraction of tumor morphology mentions, (2) their normalization to corresponding ICD-O-3 codes and (3) the final ranking for the given document. Our best configurations are underlined.

this ranking information in the evaluation. For this, the mean average precision (MAP) computes multiple precision values at all possible cutoffs of the ranking, and thus, gives higher weights to the top elements:

$$\begin{aligned} \text{MAP} &= \frac{\sum_{d \in D} \text{AveP(d)}}{|D|} \\ \text{with: AveP} &= \frac{\sum_{k=1}^{n} P(k) \times \text{rel}(k)}{\text{number of relevant ICD codes}} \end{aligned} \tag{3.6}$$

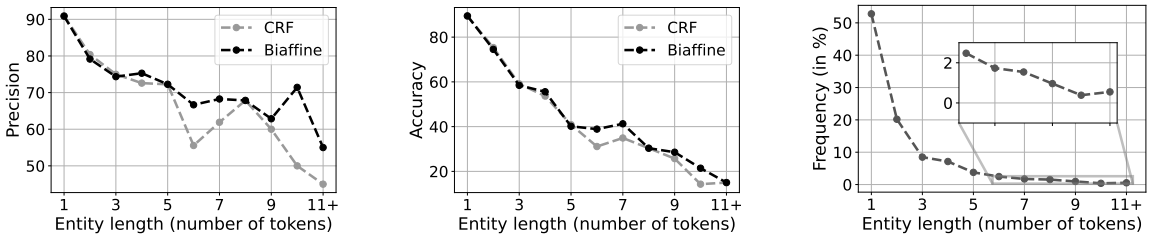
where D is a list of documents, e.g., the test set, and $\text{rel}(k) = 1$ indicates if the code at rank k is a relevant ICD code and is zero otherwise.

3.7.2 Results for NER and Normalization

The BiLSTM-CRF (S_1) with domain-specific embeddings delivers a good performance for our experiments with 82.7 F_1 for the extraction and 72.9 F_1 for the normalization. The biaffine model (S_2) achieves higher precision than the CRF with +2 F_1 points for the extraction and +1 F_1 point for the normalization on the development set. This gap further increases on the unseen test data. Overall, the biaffine model dominates because of the better precision, which might be explained by the fact that many of the tumor mentions cover multiple tokens, and the parsing model is better in capturing those long-distant dependencies. In addition, the biaffine model can be further improved by training on a combination of training and development sets, resulting in our best submission (S_3).

3.7.3 Results for ICD Coding

The results for the third subtask, the ranked coding, are close to the results of our method on the gold extractions. This indicates that the systems are able to extract the most important entities correctly. Overall, the differences between the systems are relatively small. For example, the MAP score for the biaffine model (S_2) is only 0.2 points higher than the CRF (S_1). Only the biaffine model trained on the combination of training and development data (S_3) achieves a slightly higher performance of up to a MAP score of 77.0. To conclude, all



(a) Task 1: Extraction. **(b)** Task 2: Normalization. **(c)** Relative frequencies.

Figure 3.7: Results for entities of different lengths. (a) displays the impact of entity length on the extraction and (b) for the normalization. In (c) the relative frequencies of these entities are shown. The last data point (11+) in all plots is the aggregation of all entities longer than 10 tokens.

three of our methods for the extraction, normalization, and ranking of tumor morphology mentions deliver good performance for their respective tasks, and their sequential execution as a pipeline model works well in practice.

3.7.4 Analysis: CRF vs. Biaffine Classifier

In the following section, the performance differences between the CRF and biaffine attention classifiers are analyzed with a focus on the lengths of the entities. As shown in Table 3.7, the main difference lies in the higher precision of the biaffine model. Figure 3.7a shows the precision for entities with respect to their length. In particular, for shorter entities, there are no differences in performance between the two model architectures. Starting with entities consisting of 6 and more tokens, the biaffine model begins to outperform the CRF model for the extraction and also the subsequent normalization (Fig. 3.7b). The performance difference reaches up to 20 points in precision for the extraction of multi-token entities consisting of 10 tokens and 10 points for entities longer than at least 11 tokens.

For both model types, we observe that the performance drop correlates with the length of the entities. In general, there are fewer training instances for longer entities, as shorter entities are more frequent than longer ones with a long tail of infrequent but long entities (Fig. 3.7c). This performance gap between short and long entities is even larger for the normalization, which ranges from 85 F_1 for single-token entities to 15 F_1 for entities with more than ten tokens. However, as more than half of the entities consist of a single token, the impact of longer entities on the overall F_1-score is limited and, thus, the difference of the CRF and biaffine models regarding the overall precision is 2 points, even though the biaffine model is better suited for the extraction of longer multi-token entities.

3.8 Conclusions

In this chapter, we explored the important issue of anonymization as a pre-processing step in the clinical domain. We modeled anonymization as a sequence-labeling tasks and created a state-of-the-art system based on the combination of general-domain and domain-specific word embeddings that won the MEDDOCAN shared task. Moreover, we closed the gap between using anonymization as an isolated pre-processing step and its usage in a real-world NLP pipeline. For this, we consider the anonymization of clinical text together with concept extraction, a possible downstream application. We investigate the effects of anonymization on concept extraction and show that it positively influences the concept extraction performance. We propose two models to learn both tasks jointly, a multi-task model and a stacked model, which both improve over the single-task model on medical concept extraction benchmark datasets for English and Spanish. Finally, we introduced a simple but effective ICD coding pipeline to normalize extracted clinical expressions and explored the potential of biaffine classifiers as alternatives to conditional random field output layers. The next chapter will take a closer look at more recent transformer models in the context of clinical concept extraction.

Chapter 4

Advanced Transformers for Clinical Concept Extraction

The field of natural language processing has recently seen a large change towards using pre-trained transformer models for solving almost any task. Despite showing great improvements in benchmark datasets for various tasks, these models often perform sub-optimal in non-standard domains like the clinical domain, where a large gap between the pre-training documents and target documents is observed. In this chapter, we aim at closing this gap with domain-specific training of language models, and we investigate its effect on a diverse set of downstream tasks and settings.

In particular, we introduce the pre-trained *CLIN-X* (Clinical XLM-R) language models and show how *CLIN-X* outperforms other pre-trained transformer models by a large margin for ten clinical concept extraction tasks from two languages. In addition, we propose a task-agnostic architecture for sequence labeling based on ensembles over random splits and cross-sentence context. For this, we demonstrate that domain-specific transformers can be further improved with our proposed model architecture for all tasks. Our results highlight the importance of specialized language models, such as *CLIN-X*, for concept extraction in non-standard domains, but also show that our task-agnostic model architecture is robust across the tested tasks and languages so that domain- or task-specific adaptations are not required.

This chapter is based on our submission about clinical concept extraction (Lange et al., 2022a) and partially discusses our participation in the MEDDOPROF shared task (Lange et al., 2021a) using *CLIN-X*.

4.1 Introduction

Collecting and understanding clinical information, such as disorders, symptoms, drugs, etc., from electronic health records has wide-ranging applications within clinical practice

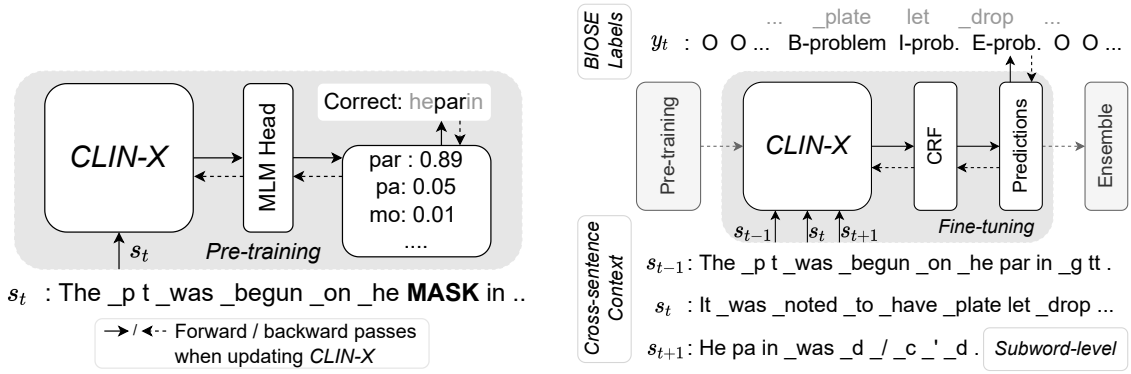
and research (Leaman et al., 2015a). A better understanding of this information can, on the one hand, facilitate novel clinical studies and, on the other hand, help practitioners to optimize clinical workflows as discussed in the previous chapter.

However, information extraction in non-standard domains like the clinical domain is a challenging problem due to a large number of complex terms and unusual document structures (Lee et al., 2020). In addition, pre-trained language models (PLM), such as BERT (Devlin et al., 2019) that demonstrated superior performance for many NLP tasks are typically trained on standard domains, such as web texts, news articles, or Wikipedia. Despite showing some robustness across languages and domains (Conneau et al., 2020), these models still achieve their best performance when applied to targets similar to their pre-training corpora, which can limit their applicability (Gururangan et al., 2020). One way to overcome this domain-gap is the training a new domain-specific model from scratch (Beltagy et al., 2019; Lee et al., 2020) or the adaptation of existing language models to the new target domain by, e.g., pre-training with masked language modeling (MLM) objectives on documents from the target domain (Weber et al., 2020; Naseem et al., 2021).

Over the last years, we have participated in a series of shared tasks on information extraction in the Spanish clinical domain (Marimon et al., 2019; Miranda-Escalada et al., 2020; Lima-López et al., 2021). With our systems (cf., Chapter 3 and Section 4.4.6, we were able to outperform the other participants and won the competitions twice. The winning systems were task agnostic and utilized domain-adapted language models and word embeddings as described in Section 3.5, as well as improved training routines for transformer models. Based on our findings and lessons learned during the competitions, we propose in this chapter a robust model architecture and training procedure for concept extraction in the clinical domain that is task and language agnostic. We introduce a new Spanish clinical language model *CLIN-X*$_{ES}$ (Clinical XLM-R) that outperforms existing transformer models on Spanish corpora and exemplifies the benefits of cross-language domain adaptation for English tasks as well and compare it to an English model: *CLIN-X*$_{EN}$. For this, we perform a comprehensive evaluation of ten clinical information extraction tasks from two languages (English and Spanish). Our results demonstrate significant and consistent improvements compared to standard transformer models across all tasks in both languages.

4.2 Model Architectures

In this section, we start with a brief description of the input representations. Then, we discuss our proposed architectural improvements as well as the advanced training methods. The overall model architecture for pre-training is shown in Figure 4.1a and fine-tuning in Figure 4.1b. The fine-tuned model, as used for downstream applications, is based on the pre-trained *CLIN-X* language model and optimized with the following four advanced techniques. First, the input is computed on the subword level instead of the usual word level, which eliminates the need for external tokenization. In addition, the input is enriched



(a) Pre-training. **(b)** Fine-tuning.

Figure 4.1: Overview of pre-training for our *CLIN-X* language models on clinical documents using the masked language modeling objective (left) and the concept extraction pipeline based on *CLIN-X* and our model components for subword-based concept extraction with cross-sentence context, BIOSE labels and CRFs (right). We highlight the pre-training and fine-tuning processes of *CLIN-X*.

with its cross-sentence context to capture a wider document context. Then, the input is processed by our transformer model that is adapted to the clinical target domain: *CLIN-X*. Finally, the model output is computed using a conditional random field (CRF) output layer. For inference, an ensemble over models trained on different training splits is computed. This reduces variance and captures the complementary knowledge from all models.

4.2.1 Input Representations for the Clinical Domain

State-of-the-art methods for concept extraction typically rely on word embeddings or language models as input representations. The standard approach is the pre-training of these models on large-scale unlabeled datasets once and their reuse as powerful representations for many downstream applications (Collobert et al., 2011). Phan et al. (2019) have shown that contextual information helps in particular in the medical domain, e.g., due to the high number of synonyms. Thus, we focus on the usage of contextualized embeddings in this work, which are most often retrieved from transformer language models nowadays. This is either done with auto-regressive language modeling (Radford et al., 2019) or masked language modeling (Devlin et al., 2019), which we use in this chapter.

Domain-specific Embeddings. A popular way to approach the challenges of NLP in non-standard domains is the inclusion of domain knowledge via domain-specific embeddings (Friedrich et al., 2020). For this, word embeddings or language models are pre-trained or further specialized on documents of the target domain. These embeddings can be used in downstream applications. This kind of domain adaptation has shown great benefits in practice (Gururangan et al., 2020). Thus, we explore domain- and language-adaptive pre-training of transformer models in this chapter.

The *CLIN-X* Pre-trained Language Model. At the time of writing,[1] there is no Spanish clinical transformer model publicly available. Thus, we train and publish the *CLIN-X*$_{ES}$ language model. The model is based on the multilingual XLM-R transformer, which was trained on 100 languages and showed superior performance in many different tasks across languages and even outperformed monolingual models in certain settings (Conneau et al., 2020). Even though XLM-R was pre-trained on 53GB of Spanish documents, this was only 2% of the overall training data. To steer this model towards the Spanish clinical domain, we sample documents from the Scielo archive and the MeSpEn resources (Villegas et al., 2018). The resulting corpus has a size of 790MB and is highly specific for our target setting. For pre-processing, we apply sentence splitting using the standoff2conll toolkit.[2] Then, we strip leading and trailing whitespace characters from the sentences and tokenize them using the XLM-R tokenizer. We cut off sentences after a maximum of 512 subtokens, the maximum sequence length of *CLIN-X* that we chose following XML-R.

For downstream tasks, we cope with longer sentences by applying a sliding-window approach with a stride of 100. Concretely, we split sequences with more than 512 subtokens into subsequences of up to 300 subtokens and enrich them with a context of 100 tokens from the previous and the following subsequence. Analogously, we also incorporate cross-sentence context as described in the next section. In general, this approach allows the model to process long sequences beyond the subtoken limit and keep dependencies between the individual subsequences based on the overlapping tokens.

We initialize *CLIN-X* using the pre-trained XLM-R weights and train masked language modeling (MLM) on the clinical corpus for three epochs which roughly corresponds to 32k steps. This process is visualized in Figure 4.1a. Note that this model is still multilingual, and we demonstrate the positive impact of cross-language domain adaptation by applying this model to English tasks.

In addition to the Spanish *CLIN-X*$_{ES}$ model, we release an English version *CLIN-X*$_{EN}$ trained on clinical Pubmed abstracts (850MB) filtered following Haynes et al. (2005) for a direct comparison of our methods in a monolingual setting. This allows researchers and practitioners to address the English clinical domain with an out-of-the-box tailored model. Pubmed is used with the courtesy of the U.S. National Library of Medicine.

4.2.2 Models for Concept Extraction

In the following section, we describe the architectural improvements we made compared to the standard transformer model for sequence-labeling as proposed by (Devlin et al., 2019).

Subword-level Inputs. Information extraction tasks are typically performed on the token level, while most transformers work on subwords instead. Thus, the input representations from transformers for tokens are either retrieved from the first subword or their average (Devlin et al., 2019). In contrast, we perform concept extraction directly on the subword

[1] state: March 2022

[2] `https://github.com/spyysalo/standoff2conll` [last accessed March 5, 2022.]

level. By doing this, there is no need for external tokenization besides the subword segmentation of the transformer. Note that the usage of domain-specific subwords is still considered beneficial in domain-specific applications (Beltagy et al., 2019; Lee et al., 2020).

Cross-sentence Context. Transformer models are well-suited to incorporate information from a larger context. Luoma and Pyysalo (2020) showed that context information from neighboring sentences has positive effects for named entity recognition on the general domain. (Finkel et al., 2004) also showed the positive impact of context for clinical concept extraction. We follow these approaches and add context information to the input similar to (Schweter and Akbik, 2020). We incorporate the context of 100 subwords to the left and right and use the document boundaries to set the context limits as all corpora are clearly structured in documents.

Conditional Random Field Output. As Kim and Kang (2021) have shown, entity recognition models in the biomedical domain tend to memorize training instances and their labels. This can result in incorrect label encodings as the model fails to generalize. A conditional random field (Lafferty et al., 2001) can constrain these incorrect sequences as the Viterbi algorithm is used for decoding (see Section 2.2.1). In addition, the CRF has advantages over a simple linear layer when it comes to long entities covering multiple tokens that frequently appear in the clinical domain (Lima-López et al., 2021).

4.2.3 Training on Data Splits

Having a robust model architecture is a good starting point for NLP in the clinical domain. However, the actual training procedure of the model might be even more important. Thus, we discuss standard and random splits, as well as ensembles over these splits in the following.

Standard Splits. Typically, each dataset is divided into training, development, and test splits. The training split is used in each epoch to train the model parameters, and the best training epoch is selected based on the evaluation score on the development set. Finally, the held-out test set is used to compute the final score for the selected model. These data splits are helpful to compare the performances of different models on standardized data. However, using the standard training split without modifications may not result in optimal performance (Gorman and Bedrick, 2019).

Random Splits. The training and development parts can be further randomly divided into n separate parts. Then, $n-1$ parts can be used for training and one part as the validation set for early stopping, similar to cross-fold validation. An ensemble based on models trained on the different data splits should be more powerful than the single models as each of them encodes complementary knowledge, which helps to reduce variance and biases (Clark et al., 2019). In our experiments, we use $n = 5$ so that we get five different settings with unique

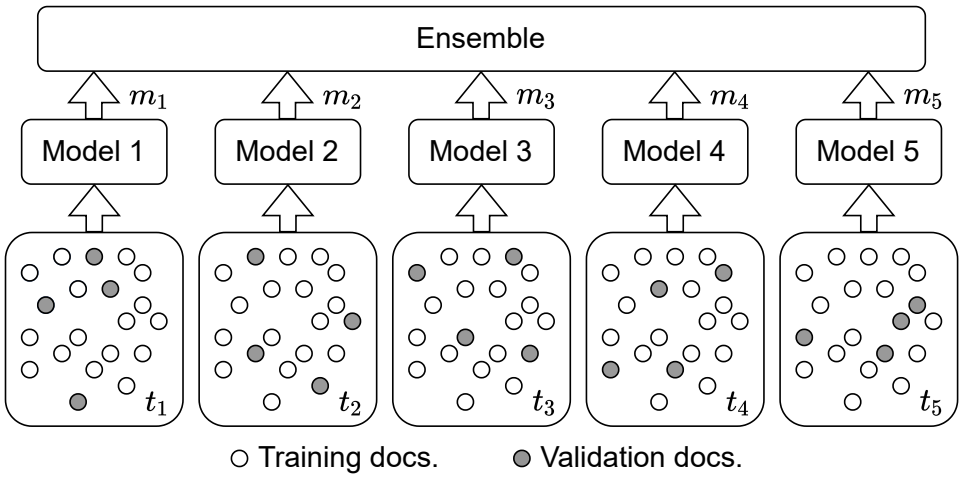


Figure 4.2: Illustration of ensembles over different training splits.

training sets, and we train one model for each setting. Note that we do not change or use the test set at all to ensure comparability to previous results.

Training on All Available Instances. Some recent works find that there is no need for a held-out development set and that these labeled instances might be better used for training. For example, Luoma and Pyysalo (2020) have shown that training on the combined training and development sets boosts performance for named entity recognition remarkably. By this, the model has access to the most data during training, and model selection is based on the training loss. However, the training loss is not as meaningful as a stopping criterion, and it is hard to pick the best model checkpoint. We will compare to this method as an alternative to our split-based experiments.

Ensembles over Models. In addition to the other methods, ensembling can be used to combine multiple model predictions into one. An ensemble is usually better than a single model — in particular, if the models or their training data differ to some degree. We create ensembles by majority voting (Clark et al., 2019) of training runs that either vary by their random seed (standard splits) or their training data (random splits).

4.3 Experimental Setup

This section describes the experiments starting with tasks, datasets, evaluation metrics, and implementation details.

4.3.1 Tasks and Datasets

Many datasets for natural language processing in specialized domains are published in the context of shared tasks — competitions to evaluate different systems and approaches. Be-

Corpus	Size (#Sentences)		
	Train	Dev	Test
I2B2-2006 (Uzuner et al., 2007)	51,429	-	18,770
I2B2-2010 (Uzuner et al., 2011)	16,487	-	27,882
I2B2-2012 (Sun et al., 2013)	7,636	-	5,785
I2B2-2014 (Stubbs et al., 2015)	52,026	-	33,317

Table 4.1: Overview of the English clinical concept extraction dataset used in this chapter. We report the number of sentences for the training, development and test splits.

Corpus	Size (#Sentences)		
	Train	Dev	Test
MEDDOCAN (Marimon et al., 2019)	15,858	8,283	8,009
PHARMACONER (Gonzalez-Agirre et al., 2019)	8,582	4,016	4,184
CANTEMIST (Miranda-Escalada et al., 2020)	19,426	18,172	11,196
MEDDOPROF (Lima-López et al., 2021)	51,350	-	10,008

Table 4.2: Overview of the Spanish clinical concept extraction datasets used in this chapter. We report the number of sentences for the training, development and test splits.

sides English, the clinical domain is well addressed for Spanish, and there exists an active community of NLP researchers that study the processing of Spanish clinical texts. Thus, in the context of the IberLEF workshop series (Iberian Language Evaluation Forum), several shared tasks have been proposed by the Barcelona Supercomputing Center concerning concept extraction in the clinical domain (Marimon et al., 2019; Gonzalez-Agirre et al., 2019; Miranda-Escalada et al., 2020; Lima-López et al., 2021). In addition to datasets of these shared tasks for Spanish, we consider four English datasets published during a series of shared tasks of the I2B2 project (Uzuner et al., 2007, 2011; Sun et al., 2013; Stubbs et al., 2015). Information on the dataset sizes are given in Table 4.1 and Table 4.2 for English and Spanish, respectively. Note that the MEDDOPROF and I2B2-2012 corpora consist of two different extraction tasks each. Thus, we consider both annotation layers as separated tasks in this work resulting in a total of ten tasks.

All of these tasks require information extraction on the token level. Therefore, we model them as sequence-labeling problems similar to our experiments in Chapter 3.

4.3.2 Evaluation Metrics

Following the evaluations in the shared tasks, we use the micro F_1-score for all datasets as the evaluation metric. The F_1-score is the harmonic mean of precision, the fraction of correct concepts among the predicted concepts, and recall, the fraction of correct concepts that were predicted as described in Section 2.1.3. To evaluate multi-token expressions, we apply strict matching, i.e., we require an exact match of all tokens to count the prediction as correct.

4.3.3 Implementation Details

Masked Language Modeling. We use eight NVIDIA V100 (32GB) GPUs for pre–training the *CLIN-X* models. The training takes less than one day with a batch size of four sentences per device and a sequence length of up to 512 subwords. The models were trained with the HuggingFace trainer for MLM.[3]

Sequence Labeling. The sequence-labeling models were trained on single NVIDIA V100 GPUs for up to 20 hours, depending on the dataset size. The models were trained using the Flair framework with the AdamW optimizer with an initial learning rate of 2.0×10^{-5} and a batch size of 16 for 20 epochs. The loss function is the CRF loss when using a CRF layer and cross-entropy loss otherwise. The model selection was performed using the development score if trained on standard or random splits and the training loss otherwise.

4.4 Results and Analysis

This section will discuss the results of our experiments. First, we evaluate the different embeddings methods and study the effects of domain-specific training. Then, we evaluate the different training methods and their ensembles and perform an ablation study. Then, we compare our models to the current state of the art for clinical concept extraction. Finally, we perform a qualitative analysis of our models.

4.4.1 Results for Different Embeddings

The choice of input embeddings has a large impact on downstream performance and may even be the most important factor. Table 4.3 shows the average performance of several different embeddings and transformer models for the two languages. As expected, the monolingual transformers (BERT, BETO) excel at their target language but cannot compete with multilingual models (mBERT, XLM-R) when applied to another language. The lower part of Table 4.3 lists domain-specific variants of the embeddings, which are generally more powerful in our domain-specific setting. We see that our *CLIN-X* models perform best for their respective languages. Furthermore, the *CLIN-X*$_{ES}$ performs almost as well as the *CLIN-X*$_{EN}$ model on the English datasets, for which it was not explicitly trained. This shows that the domain adaptation of multilingual models can also help for texts from other languages of the same domain, i.e., cross-language domain adaptation.

4.4.2 Results for Different Training Methods

The foundation for all following concept extraction models is the *CLIN-X*$_{ES}$ transformer, as it has shown robust results across all tasks. For comparison to fixed standard splits, we

[3] `https://github.com/microsoft/huggingface-transformers/blob/master/examples/pytorch/language-modeling/run_mlm.py` [last accessed March 5, 2022.]

Pre-training Domain	Model	English	Spanish
General (e.g., Web, News, Wikipedia, ...)	word2vec	80.26	78.20
	Flair	85.15	80.28
	BERT (En)	85.34	77.78
	BETO (Es)	83.57	83.92
	XLM-R	87.13	83.87
Clinical	word2vec	80.98	79.72
	Flair	86.43	80.72
	ClinicalBERT (En)	85.76	76.94
	CLIN-X$_{EN}$	**87.67**	84.57
	CLIN-X$_{ES}$	87.48	**85.37**

Table 4.3: Results for different embeddings and models averaged for the two languages (F_1). Word embeddings are used in a RNN model as in Chapter 4. Transformers are used with a classification layer similar to Devlin et al. (2019).

train the model on different random splits. We see in Table 4.4 that, in particular, ensembles over random splits are a lot better than the standard splits and also all training instances. While the median performance is roughly similar for all methods, the random splits offer a lot more variety in training instances. Thus, the ensemble based on random splits also achieves much higher scores.

4.4.3 Ablation Studies

The lower part of Table 4.4 lists an ablation study of our individual model components. For example, adding cross-sentence context to the transformers boosts performance across all tasks by 0.5 F1 on average. Performing concept extraction on the subword level helps

		Method	English	Spanish
		All training instances	87.83	86.46
Standard	Splits	Median model	87.63	85.16
		Best model	87.85	85.99
		Ensemble	87.95	86.06
Random	Splits	Median model	87.69	86.17
		Best model	88.31	86.85
		Ensemble	**88.78**	**88.15**
Ablation	Study	– BIOSE Labels	88.52	87.13
		– CRF	88.38	85.95
		– Context	87.83	86.84
		– Subword NER	87.38	86.81

Table 4.4: Comparison of training splits with our model architecture and ablation study of the model components averaged for each language (F_1).

even further. This is particularly beneficial considering that no external tokenization is needed, which can be challenging in the clinical domain. The CRF is useful for both languages, though the differences are larger for Spanish, as the two MEDDOPROF tasks have particularly long annotations (2.53 tokens per annotation on average). The same holds for the BIOSE labels, which have the smallest impact of all components but consistently improve upon the standard BIO labels. As each of our proposed methods improves the transformer even further, we use the combination of all methods in the following as our model architecture.

4.4.4 Qualitative Analysis

We provide a qualitative analysis of four sample sentences from the test set of I2B2-2012 in Table 4.5. The analysis shows that our *CLIN-X* model can correctly annotate domain-specific concepts like *orthostatic* (S_1) and *[..] rectures adbominis flap* (S_2) than the general-domain XLM-R model due to its additional domain knowledge. Our architecture helps the model to focus on the target task and prevent the model from annotating false positives like *PO* ("per os"; orally) (S_4). However, it is still able to recognize general-domain words correctly in a clinical context like *wire* (S_3) and identify the correct annotation type (S_2).

Sentence 1 (S_1)	
Gold:	She was not[orthostatic]$_{\text{Problem}}$
XLM-R:	She was not orthostatic
CLIN-X$_{ES}$:	She was not[orthostatic]$_{\text{Problem}}$
CLIN-X$_{ES}$ +OURMODEL:	She was not[orthostatic]$_{\text{Problem}}$
Sentence 2 (S_2)	
Gold:	[Right superiorly based rectures abdominis flap]$_{\text{Treatment}}$
XLM-R:	Right superiorly based rectures abdominis flap
CLIN-X$_{ES}$:	[Right superiorly based rectures abdominis flap]$_{\text{Problem}}$
CLIN-X$_{ES}$ +OURMODEL:	[Right superiorly based rectures abdominis flap]$_{\text{Treatment}}$
Sentence 3 (S_3)	
Gold:	Access-[RIJ]$_{\text{Treatment}}$ changed sterilly over[wire]$_{\text{Treatment}}$
XLM-R:	Access-[RIJ]$_{\text{Treatment}}$ changed sterilly over wire
CLIN-X$_{ES}$:	Access-[RIJ]$_{\text{Treatment}}$ changed sterilly over wire
CLIN-X$_{ES}$ +OURMODEL:	Access-[RIJ]$_{\text{Treatment}}$ changed sterilly over[wire]$_{\text{Treatment}}$
Sentence 4 (S_4)	
Gold:	Plan to restart[Paxil]$_{\text{Treatment}}$ when taking PO
XLM-R:	Plan to restart[Paxil]$_{\text{Treatment}}$ when taking[PO]$_{\text{Treatment}}$
CLIN-X$_{ES}$:	Plan to restart[Paxil]$_{\text{Treatment}}$ when taking[PO]$_{\text{Treatment}}$
CLIN-X$_{ES}$ +OURMODEL:	Plan to restart[Paxil]$_{\text{Treatment}}$ when taking PO

Table 4.5: Qualitative analysis of sample sentences from the I2B2-2012 corpus.

English (I2B2)	2006	2010	2012-C	2012-T	2014
BERT/BETO (monolingual)	94.80	85.25	76.51	75.28	94.86
BERT (multilingual)	94.79	84.91	76.01	76.56	95.34
XLM-R (multilingual)	96.72	87.54	79.63	75.36	96.39
HunFlair (monolingual)	93.48	86.70	78.52	77.16	95.90
ClinicalBERT	94.8	87.8	78.9	76.58	93.0
CLIN-X$_{EN}$	96.25	88.10	79.58	77.70	96.73
CLIN-X$_{ES}$	95.49	87.94	79.58	77.57	96.80
CLIN-X$_{EN}$ +OURMODEL	**98.49**	**89.23**	**80.62**	**78.50**	97.60
CLIN-X$_{ES}$ +OURMODEL	98.30	89.10	80.42	78.48	**97.62**

Spanish	CANTEMIST	MEDDOCAN	M.PROF-N	M.PROF-C	PHARMA.
BERT/BETO (monolingual)	81.30	96.81	79.19	74.59	87.70
BERT (multilingual)	80.94	96.30	76.39	71.84	86.98
XLM-R (multilingual)	82.17	96.76	77.44	74.05	88.92
HunFlair (monolingual)	83.80	96.50	75.16	70.01	88.40
ClinicalBERT	77.18	94.63	65.74	62.85	84.32
NLNDE	85.3	96.96	81.8	79.3	88.6
CLIN-X$_{EN}$	82.80	97.08	78.62	75.05	89.33
CLIN-X$_{ES}$	83.22	97.08	79.54	76.95	90.05
CLIN-X$_{EN}$ +OURMODEL	87.72	97.57	81.36	78.53	**92.36**
CLIN-X$_{ES}$ +OURMODEL	**88.24**	**98.00**	**81.68**	**80.54**	92.27

Table 4.6: Performance of our *CLIN-X* models in comparison to baseline systems and state-of-the-art results (F_1).

4.4.5 Comparison to State-of-the-Art Models

As our results demonstrate, we have proposed a robust model for the clinical domain that works well across the different tasks in both languages. Finally, we compare *CLIN-X* to various transformer models as introduced earlier. We also compare to HunFlair (Weber et al., 2021), the current state of the art for concept extraction in the biomedical domain. We use their model architecture based on clinical Flair and fastText embeddings and train models accordingly on our datasets. In addition, we compare to our NLNDE submissions for the Spanish shared tasks (see Chapter 3 and Section 4.4.6) and the ClinicalBERT by Alsentzer et al. (2019) for the English datasets. The results for each task are shown in Table 4.6. The *CLIN-X* language models in combination with our model architecture outperform the other transformers and HunFlair by a large margin. *CLIN-X* is able to utilize the domain knowledge obtained from the additional pre-training with further improvements from the ensembling over random splits. Even though *CLIN-X* works best in combination with our model architecture, *CLIN-X* based on the standard transformer architecture with a single classification layer already outperforms the existing models on 8 out of 10 tasks.

Team / Model	Subtrack 1 : NER			Subtrack 2 : CLASS		
	Pre.	Rec.	F_1	Pre.	Rec.	F_1
XLM-R	83.9	75.0	79.2	81.2	74.3	77.6
XLM-R$_{ES}$	**86.3**	80.4	**83.2**	82.5	**75.9**	79.1
CLIN-X$_{ES}$	83.8	76.6	80.0	80.7	75.4	77.9
OUR ensemble of all three (submission)	85.5	**78.3**	81.8	**83.0**	**75.9**	**79.3**
MUCIC (Balouchzahi et al., 2021)	81.3	78.8	80.0	77.0	75.5	76.4
SMR-NLP (Siemens AG / LMU Munich)	85.4	75.1	79.9	80.2	69.9	74.7

Table 4.7: Results for MEDDOPROF systems.

4.4.6 *CLIN-X* Model in the MEDDOPROF Shared Task

The *CLIN-X* language model and our previously described model architecture were originally developed for the MEDDOPROF shared task on automatic recognition (subtrack 1) and classification (subtrack 2) of professions and occupations from medical texts in Spanish (Lima-López et al., 2021). For this, we pre-trained the *CLIN-X*$_{ES}$ model on Spanish clinical documents. In addition, we trained a general-domain Model XLM-R$_{ES}$ on various other Spanish texts, as the target documents belong to the clinical domain, but the task itself is concerned with the extraction of general-domain concepts, namely different professions and occupations. In contrast to the training on random splits as in the earlier parts of this chapter, we trained our model on so-called strategic data splits (Wecker et al., 2020). For this, the training documents are clustered using k-Means clustering over document embedding vectors instead of random splits, which creates more challenging data splits. More details are given by Lange et al. (2021a).

The results are shown in Table 4.7. We experimented with the standard XLM-R model, our fine-tuned XLM-R$_{ES}$ on Spanish texts, as well as our domain-adapted *CLIN-X* model. Our final submission was an ensemble of all three models, which won both subtracks of the shared task. This ensemble outperformed all other 14 participating teams and achieved +1.8 and +2.9 F_1 points compared to the second-best team (Balouchzahi et al., 2021).

4.5 Conclusions

In this chapter, we described the newly pre-trained language model *CLIN-X* for the clinical domain. We have shown that *CLIN-X* sets the new state of the art for ten clinical concept extraction tasks in two languages. We demonstrated the positive impact of other model components, such as for ensembles over random splits and cross-sentence context. We think that the release of the *CLIN-X* language model and our model architecture will be a good starting point for many clinical NLP tasks and will enable further research on clinical concept extraction. Moreover, we participated with *CLIN-X* and a general-domain variant in the MEDDOPROF shared task and outperformed all other participating systems, which highlights the importance of domain-adapted systems like *CLIN-X* and robust model architectures.

Chapter 5

Meta-Embeddings for Domain-Robust Input Representations

Combining several embeddings typically improves performance in downstream tasks as different embeddings encode different information, as exemplarily discussed in Chapter 3 for the inclusion of domain knowledge. Moreover, it has been shown that even models using embeddings from transformers still benefit from the inclusion of standard word embeddings. However, the combination of embeddings of different types and dimensions is challenging. In this chapter, we propose feature-based adversarial meta-embeddings (FAME) as an alternative to the simple concatenation. These robust meta-embeddings variants come with an attention function guided by features reflecting word-specific properties, such as shape and frequency. We show that this is beneficial to handle subword-based embeddings. In addition, FAME uses adversarial training to optimize the mappings of differently-sized embeddings to the same space. We demonstrate that FAME works effectively across languages and domains for sequence labeling and sentence classification, in particular in low-resource settings. FAME sets the new state of the art for POS tagging in 27 languages, various NER and concept extraction tasks, and question classification in different domains. It was originally introduced in our publication on robust meta-embedding methods (Lange et al., 2021b).

5.1 Introduction

Recent work on word embeddings and pre-trained language models has shown the large impact of language representations on natural language processing models across tasks and domains (Devlin et al., 2019; Beltagy et al., 2019; Conneau et al., 2020). Nowadays, a large number of different embedding models are available with different characteristics, such as different input granularities (word-based (e.g., Mikolov et al., 2013a; Pennington et al., 2014) vs. subword-based (e.g., Heinzerling and Strube, 2018; Devlin et al., 2019)

vs. character-based (e.g., Lample et al., 2016; Ma and Hovy, 2016; Peters et al., 2018)), or different data used for pre-training (general-world vs. specific domain). Since those characteristics directly influence when embeddings are most effective, combinations of different embedding models are likely to be beneficial (Tsuboi, 2014; Kiela et al., 2018), even when using already powerful large-scale pre-trained language models (Akbik et al., 2018; Yu et al., 2020). Word-based embeddings, for instance, are strong in modeling frequent words, while character-based embeddings can model out-of-vocabulary words. Similarly, domain-specific embeddings can capture in-domain words that do not appear in general domains like news text or Wikipedia articles.

Different word representations can be combined using so-called meta-embeddings. There are several methods available, ranging from concatenation (e.g., Yin and Schütze, 2016), over averaging (e.g., Coates and Bollegala, 2018) to attention–based meta–embeddings (Kiela et al., 2018). However, they all come with shortcomings: Concatenation leads to high-dimensional input vectors and, as a result, requires additional parameters in the first layer of the neural network. Averaging simply merges all information into one vector, not allowing the network to focus on specific embedding types, which might be more effective than others to represent the current word. Attention-based embeddings address this problem by allowing dynamic combinations of embeddings depending on the current input token. However, the calculation of attention weights requires the model to assess the quality of embeddings for a specific word. This is arguably very challenging when embeddings of different input granularities are combined, e.g., subwords and words. Infrequent in-domain tokens, for instance, are hard to detect when using subword-based embeddings as they can model any token. Moreover, both average and attention-based meta-embeddings require mappings of all embeddings into the same space, which can be challenging for a set of embeddings with different dimensions.

In this chapter, we propose feature-based adversarial meta-embeddings (FAME) that (1) align the embedding spaces with adversarial training, and (2) use attention for combining embeddings with a layer that is guided by features reflecting word-specific properties, such as the shape or frequency of the word and, thus, can help the model to assess the quality of the different embeddings. By using attention, we avoid the shortcomings of concatenation (high-dimensional input vectors) and averaging (merging information without focus). Further, our contributions mitigate the challenges of previous attention-based meta-embeddings: In our analysis, we show that the first contribution is especially beneficial when embeddings of different dimensions are combined. The second helps, in particular, when combining word-based with subword-based embeddings.

We conduct experiments across a variety of tasks, languages, and domains, including sequence-labeling tasks (named entity recognition for four languages, concept extraction for two special domains (clinical and materials science), and part-of-speech tagging for 27 languages) and sentence classification tasks (question classification in different domains). Our results and analyses show that FAME outperforms existing meta-embedding methods and that even powerful fine-tuned transformer models can benefit from additional embeddings using our method. In particular, FAME sets the new state of the art for POS tagging

in all 27 languages, for NER in two languages, as well as on all tested concept extraction and two question classification datasets.

In summary, our contributions are meta-embeddings with (i) adversarial training and (ii) a feature-based attention function. (iii) We perform broad experiments, ablation studies, and analyses that demonstrate that our method is highly effective across tasks, domains, and languages, including low-resource settings. (iv) Moreover, we show that even representations from large-scale pre-trained transformer models can benefit from our meta-embeddings approach.

5.2 Related Work

This section surveys related work on meta-embeddings, attention, and adversarial training.

Meta-Embeddings. Previous work has seen performance gains by, for example, combining various types of word embeddings (Tsuboi, 2014) or the same type trained on different corpora (Luo et al., 2014). For the combination, some alternatives have been proposed, such as different input channels of a convolutional neural network (Kim, 2014; Zhang et al., 2016), concatenation followed by dimensionality reduction (Yin and Schütze, 2016) or averaging of embeddings (Coates and Bollegala, 2018), e.g., for combining embeddings from multiple languages (Reid et al., 2020). More recently, auto-encoders (Bollegala and Bao, 2018; Wu et al., 2020) ensembles of sentence encoders (Pörner et al., 2020) and attention-based methods (Kiela et al., 2018) have been introduced. The latter allows a dynamic (input-based) combination of multiple embeddings. Winata et al. (2019a) and Priyadharshini et al. (2020) used similar attention functions to combine embeddings from different languages for NER in code-switching settings. Liu et al. (2021) explored the inclusion of domain-specific semantic structures to improve meta-embeddings in non-standard domains. In this chapter, we follow the idea of attention-based meta-embeddings and propose task-independent methods for improving them.

Extended Attention. Attention has been introduced in the context of machine translation (Bahdanau et al., 2015) and is since then widely used in NLP (i.a., Tai et al., 2015; Xu et al., 2015; Yang et al., 2016; Vaswani et al., 2017). Our approach extends this technique by integrating word features into the attention function. This is similar to extending the source of attention for uncertainty detection (Adel and Schütze, 2017) or relation extraction (Zhang et al., 2017b; Li et al., 2019). However, in contrast to these works, we use task-independent features derived from the token itself. Thus, we can use the same attention function for different tasks.

Adversarial Training. Further, our method is motivated by the usage of adversarial training (Goodfellow et al., 2014b) for creating input representations that are independent of a specific domain or feature. This is related to using adversarial training for domain adaptation (Ganin et al., 2016) or coping with bias or confounding variables (Beutel et al.,

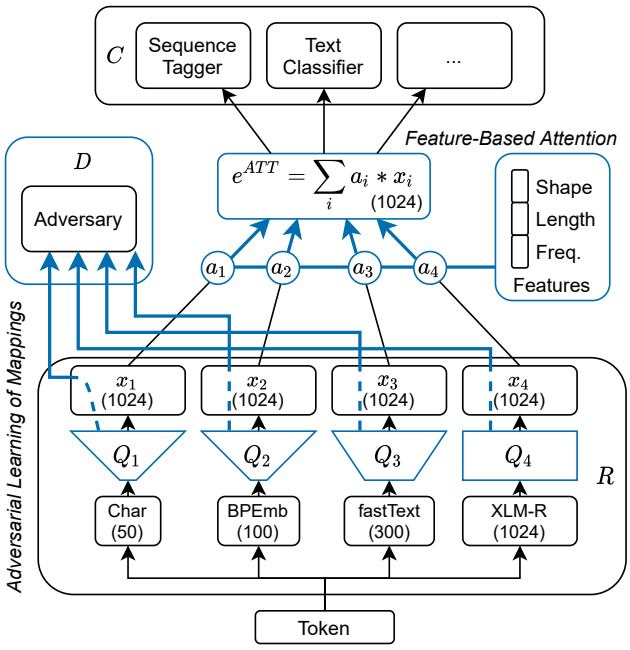


Figure 5.1: Overview of the FAME model architecture. Blue lines highlight our contributions. C (classifier), D (discriminator) and R (input representation) denote the components of adversarial training. The dimensions of intermediate representations are given in parentheses.

2017; Li et al., 2018; Raff and Sylvester, 2018; Zhang et al., 2018; Barrett et al., 2019; McHardy et al., 2019). Following Ganin et al. (2016), we use gradient reversal training in this chapter. Recent studies use adversarial training on the word level to enable cross-lingual transfer from a source to a target language (Zhang et al., 2017a; Keung et al., 2019; Wang et al., 2019a; Bari et al., 2020). In contrast, our discriminator is not binary but multinomial (as in Chen and Cardie (2018)) and allows us to create a common space for embeddings from different granularities. Adversarial training was also used to strengthen non-textual representations, e.g., knowledge graphs (Zeng et al., 2020) or networks (Dai et al., 2019).

5.3 Meta-Embeddings

This section describes the existing attention-based meta-embeddings method and presents our proposed FAME model with feature-based meta-embeddings and adversarial training. The FAME model is depicted in Figure 5.1.

5.3.1 Attention-Based Meta-Embeddings

As discussed in Chapter 3, multiple embeddings can be combined by concatenating all embeddings vectors. However, as some embeddings are more effective in modeling certain words, e.g., domain-specific embeddings for in-domain words, we use attention-based

meta-embeddings that are able to combine different embeddings dynamically as introduced by Kiela et al. (2018) by computing a weighted sum of the embeddings.

Given n embeddings $e_1 \in \mathbb{R}^{E_1}$, ... $e_n \in \mathbb{R}^{E_n}$ of potentially different dimensions E_1, ... E_n, they first need to be mapped to the same space (with E dimensions):

$$x_i = \tanh(Q_i \cdot e_i + b_i) \tag{5.1}$$

with $1 \leq i \leq n$. Note that the mapping parameters $Q_i \in \mathbb{R}^{E \times E_i}$ and $b_i \in \mathbb{R}^E$ are learned for each embedding method during training of the downstream task. Then, attention weights α_i are computed by:

$$\alpha_i = \frac{\exp(V \cdot \tanh(W x_i))}{\sum_{l=1}^{n} \exp(V \cdot \tanh(W x_l))} \tag{5.2}$$

with $W \in \mathbb{R}^{H \times E}$ and $V \in \mathbb{R}^{1 \times H}$ being parameter matrices that are randomly initialized and learned during training. Finally, the embeddings x_i are weighted using the attention weights α_i resulting in the word representation:

$$e^{ATT} = \sum_i \alpha_i \cdot x_i \tag{5.3}$$

This approach requires the model to learn parameters for the mapping function as well as for the attention function. The first might be challenging if the original embeddings have different dimensions, while the latter might be problematic if the embeddings represent inputs from different granularities, such as words vs. subwords. We support this claim experimentally in our analysis in Section 5.7.2.

5.3.2 Feature-Based Attention

Equation 5.2 for calculating attention weights only depends on x_i, the representation of the current word.[1] While this can be enough when only standard word embeddings are used, subword- and character-based embeddings are able to create vectors for out-of-vocabulary inputs, and distinguishing these from tailored vectors for frequent words is challenging without further information (see Section 5.7.2). To allow the model to make an informed decision which embeddings to focus on, we propose to use the features described below as an additional input to the attention function. The word features are represented as a vector $f \in \mathbb{R}^F$ and integrated into the attention function (Equation 5.2) as follows:

$$\alpha_i = \frac{\exp(V \cdot \tanh(W x_i + U f))}{\sum_{l=1}^{n} \exp(V \cdot \tanh(W x_l + U f))} \tag{5.4}$$

with $U \in \mathbb{R}^{H \times F}$ being a parameter matrix that is learned during training.

[1] Kiela et al. (2018) proposed two versions: using the word embeddings or using the hidden states of a bidirectional LSTM encoder. Our observation holds for both of them.

Features. FAME uses the following task-independent features based on word characteristics. Some of these were earlier discussed in Section 2.2.1 as CRF inputs features.

- *Length*: Long words, in particular compounds, are often less frequent in embedding vocabularies, such that the word length can be an indicator for rare or out-of-vocabulary words. We encode the lengths in 20-dimensional one-hot vectors. Words with more than 19 characters share the same vector.
- *Frequency*: High-frequency words can typically be modeled well by word-based embeddings, while low-frequency words are better captured with subword-based embeddings. Moreover, frequency is domain-dependent and can thus help to decide between embeddings from different domains. We estimate the frequency n of a word in the general domain from its rank r in the fastText-based embeddings provided by Grave et al. (2018): $n(r) = k/r$ with $k = 0.1$ following Manning and Schütze (1999). Finally, we group the words into 20 bins as done by Mikolov et al. (2011) and represent their frequency with a 20-dimensional one-hot vector.
- *Word Shape*: Word shapes capture certain linguistic features and are often part of manually designed feature sets, e.g., for CRF classifiers (Lafferty et al., 2001). For example, uncommon word shapes can be indicators for domain-specific words, which can benefit from domain-specific embeddings. We create 12 binary features that capture information on the word shape, including whether the first, any, or all characters are uppercased, alphanumerical, digits, or punctuation marks.
- *Word Shape Embeddings:* In addition, we train word shape embeddings (25 dimensions) similar to Limsopatham and Collier (2016). For this, the shape of each word is converted by replacing letters with *c* or *C* (depending on the capitalization), digits with *n* and punctuation marks with *p*. For instance, *Dec. 12th* would be converted to *Cccp nncc*. The resulting shapes are one-hot encoded, and a trainable randomly initialized linear layer is used to compute the shape representation.

All sparse feature vectors (binary or one-hot encoded) are fed through a linear layer to generate a dense representation. Finally, all features are concatenated into a single feature vector f of 77 dimensions which is used in the attention function as described earlier.

5.3.3 Adversarial Learning of Mappings

The attention-based meta-embeddings require that all embeddings have the same dimension for summation. For this, mapping matrices need to be learned, as only a limited number of embeddings exist for many languages and domains, and there is typically no option only to use embeddings of the same size. To learn effective mappings, we propose to use adversarial training. In particular, FAME adapts gradient-reversal training with three components: the representation module R consisting of the different embedding models and the mapping functions Q to the common embedding space, a discriminator D that tries to

	Dimensions	Fine-tuned?
General Domain		
Character	50	Yes
BPEmb	100	No
fastText	300	No
XLM-R	1024	No / Yes
Domain-specific		
Word	100 (*en*), 300 (*es*)	No
Transformer	768 (*en*)	No / Yes

Table 5.1: Overview of embeddings used in our models.

distinguish the different embeddings from each other, and a downstream classifier C which is either a sequence tagger or a sentence classifier in our experiments (and is described in more detail in Section 5.4).

The input representation is shared between the discriminator and downstream classifier and trained with gradient reversal to fool the discriminator. To be more specific, the discriminator D is a multinomial non-linear classification model with a standard cross-entropy loss function L_D. In our sequence-tagging experiments, the downstream classifier C has a conditional random field (CRF) output layer and is trained with a CRF loss L_C to maximize the log probability of the correct tag sequence (Lample et al., 2016). In our sentence classification experiments, C is a multinomial classifier with cross-entropy loss L_C. Let θ_R, θ_D, θ_C be the parameters of the representation module, discriminator, and downstream classifier, respectively. Gradient reversal training will update the parameters as follows:

$$\theta_D = \theta_D - \eta\lambda\frac{\partial L_D}{\partial \theta_D}; \quad \theta_C = \theta_C - \eta\frac{\partial L_C}{\partial \theta_C}; \quad \theta_R = \theta_R - \eta(\frac{\partial L_C}{\partial \theta_R} - \lambda\frac{\partial L_D}{\partial \theta_R}) \tag{5.5}$$

with η being the learning rate and λ being a hyperparameter to control the discriminator influence.

5.4 Model Architectures

In this section, we present the architectures we use for text classification and sequence tagging. Note that our contribution concerns the input representation layer, which can be used with any NLP model, e.g., also sequence-to-sequence models.

5.4.1 Input Layer

The input to our neural networks is our FAME meta-embeddings layer as described in Section 5.3. Our methodology does not depend on the embedding method, i.e., it can incorporate any token representation. In our experiments, we use the embeddings listed in Table 5.1 based on insights from related work. In particular, Akbik et al. (2018) showed the

Meta-embeddings method	Transformer fine-tuned? No	Yes
General Domain (4 embeddings)		
Concatenation	10.0 / 3.4	543.9 / 539.4
Attention-based meta-emb	4.0 / 4.0	537.9 / 538.9
Feature-based attention	4.0 / 4.0	538.0 / 538.9
Domain-specific (4+2 embeddings)		
Concatenation	14.9 / 5.3	652.2 / 648.2
Attention-based meta-emb	4.9 / 4.9	642.2 / 643.2
Feature-based attention	5.0 / 4.9	642.2 / 643.2
+ Adversarial Discriminator	+1.0 / +1.0	+1.0 / +1.0

Table 5.2: Number of trainable parameters (in million) of our models for sequence labeling / text classification.

advantages of character and fastText embeddings (Bojanowski et al., 2017) and Heinzerling and Strube (2018) showed similar results for character and BPE embeddings. Thus, we decided to use the union (char+fastText +BPE) with a state-of-the-art multilingual Transformer (Conneau et al., 2020, XLM-R). Our character-based embeddings are randomly initialized and accumulated to token embeddings using a bidirectional long short-term memory network (Hochreiter and Schmidhuber, 1997) with 25 hidden units in each direction.

For experiments in non-standard domains, we add domain-specific embeddings, including word embeddings from the clinical domain for English (Pyysalo et al., 2013) and Spanish (Gutiérrez-Fandiño et al., 2021) and the materials science domain (Tshitoyan et al., 2019). Further, we include domain-specific transformer models for experiments on English data, i.e., Clinical BERT (Alsentzer et al., 2019) trained on MIMIC, and SciBERT (Beltagy et al., 2019) trained on academic publications from semantic scholar.[2]

For all experiments, our baselines and proposed models use the same set of embeddings. We experiment with both freezing and fine-tuning the transformer embeddings during training. However, note that fine-tuning the transformer model increases the model size by more than a factor of 100 from 4M trainable parameters to 535M as shown in Table 5.2. This increases computational costs by a large margin. For example, in our experiments, the time for training a single epoch for English NER increases from 3 to 38 minutes.

5.4.2 Models for Sequence Tagging

Our sequence tagger follows a well-known architecture (Lample et al., 2016) as described in Section 3.3 and is based on a BiLSTM-CRF network. Note that we perform sequence tagging on sentence level without cross-sentence context as done, i.a., in Chapter 4 or by Schweter and Akbik (2020).

[2] `https://www.semanticscholar.org/` [last accessed March 5, 2022.]

5.4.3 Models for Text Classification

For sentence classification tasks, we use a BiLSTM sentence encoder. The resulting sentence representation is fed into a linear layer followed by a softmax activation that outputs label probabilities. For natural language inference, i.e., sentence pair classification, premise, and hypothesis are encoded individually. Then, their representations u and v are combined using $[u, v, u * v, |u - v|]$. Again, a linear layer followed by a softmax performs the classification.

5.5 Experimental Setup

We now describe the tasks, datasets, and details of our models and training procedure.

5.5.1 Tasks and Datasets

We use named entity recognition and part-of-speech tagging datasets from different domains and languages for sequence labeling. For NER, we use the CONLL benchmark datasets from the news domain (English/German/Dutch/Spanish) (Tjong Kim Sang, 2002; Sang and Meulder, 2003). In addition, we conduct experiments for concept extraction on two datasets from the clinical domain, the English I2B2-2010 data (Uzuner et al., 2011) and the Spanish PHARMACONER task (Gonzalez-Agirre et al., 2019), as well as experiments on the materials science domain (Friedrich et al., 2020). For POS tagging, we use the universal dependencies treebanks version 1.2 (UPOS tag) and use the 27 languages for which Yasunaga et al. (2018) reported numbers. Moreover, we experiment with three question classifications tasks, namely the TREC corpus (Voorhees and Tice, 1999) with 6 or 50 labels and GARD (Roberts et al., 2014, clinical domain), as well as the SNLI corpus for natural language inference (Bowman et al., 2015).

5.5.2 Hyperparameters and Training

We use hidden sizes of 256 units per direction for all BiLSTMs. The attention layer has a hidden size H of 10. We set the mapping size E to the size of the largest embedding in all experiments, i.e., 1024 dimensions, the size of XLM-R embeddings. The discriminator D has a hidden size of 1024 units and is trained every 10th batch. We perform a hyperparameter search for the λ parameter in $\{1e-4, 1e-5, 1e-6, 1e-7\}$ for models using adversarial training. Note that we use the same hyperparameters for all models and all tasks. Labels for sequence tagging are encoded in BIOSE format.

For training, we use the AdamW optimizer (Loshchilov and Hutter, 2019) with an initial learning rate of $5e-6$. We train the models for a maximum of 100 epochs and select the best model according to the performance using the task's metric on the development set if

Model	*en*	*de*	*es*	*nl*
Akbik et al. (2019)	93.18	88.27	-	90.12
Schweter and Akbik (2020)	93.69	**92.29**	89.93	94.66
Yu et al. (2020)	93.5	90.3	**90.3**	93.7
XLM-R (Conneau et al., 2020)	92.92	85.81	89.72	92.53
FAME (our model)	**94.11**	92.28	89.90	**95.42**

Table 5.3: NER results (F_1).

Model	CLIN$_{en}$	CLIN$_{es}$	SOFC$_{en}$
Alsentzer et al. (2019)	87.7	-	-
Joint Anonymization (Chapter 3)	88.9	91.4	–
CLIN-X Model (Chapter 4)	89.74	92.35	-
Friedrich et al. (2020)	–	–	81.5
FAME (our model)	**90.08**	**92.68**	**83.68**

Table 5.4: Concept extraction results (F_1).

available or using the training loss otherwise. The training was performed on Nvidia Tesla V100 GPUs with 32GB VRAM.[3]

5.6 Results

We now describe our results for the meta-embedding models for the different sequence-labeling tasks and for text classification. Then, we study meta-embeddings for domain-specific transformers and adapt them for subword-level sequence tagging.

5.6.1 Results for Sequence Labeling

We now present the results of our sequence-labeling experiments. All reported numbers are the averages of three runs. Following prior work, we report the micro-F_1 for the NER and clinical corpora, the macro-F_1^A (c.f., Section 2.1.3) for the SOFC corpus, and accuracy for the POS corpora.

Table 5.3 shows the results of on the popular CONLL benchmark datasets for NER in comparison to the state of the art. Our methods consistently improve upon the existing meta-embedding approaches and achieve state-of-the-art performance on 2 out of 4 languages and competitive results on the other, while maintaining a comparably low-dimensional input representation. In total, our model creates a representation of 1,024 dimensions, while other state-of-the-art systems, e.g., Akbik et al. (2019) use up to 8,292 dimensions to represent input tokens. The comparison with XML-R [4] on NER shows that

[3] All experiments ran on a carbon-neutral GPU cluster.

[4] The XLM-R model was evaluated on the 2003 version of the German corpus, while Schweter and Akbik (2020) and our models were evaluated on the revised 2006 version.

	Plank et al. (2016)	Yasunaga et al. (2018)	Heinzerling and Strube (2019)	FAME
bg (Bulgarian)	97.97	98.53	98.7	**99.53**
cs (Czech)	98.24	98.81	98.9	**99.33**
da (Danish)	96.35	96.74	97.0	**99.13**
de (German)	93.38	94.35	94.0	**95.95**
en (English)	95.17	95.82	95.6	**98.09**
es (Spanish)	95.74	96.44	96.5	**97.75**
eu (Basque)	95.51	94.71	95.6	**97.66**
fa (Persian)	97.49	97.51	97.1	**98.68**
fi (Finnish)	95.85	95.40	94.6	**98.67**
fr (French)	96.11	96.63	96.2	**97.19**
he (Hebrew)	96.96	97.43	96.6	**98.00**
hi (Hindi)	97.10	97.21	97.0	**98.35**
hr (Croatian)	96.82	96.32	96.8	**97.96**
id (Indonesian)	93.41	94.03	93.4	**94.24**
it (Italian)	97.95	98.08	98.1	**98.82**
nl (Dutch)	93.30	93.09	93.8	**94.74**
no (Norwegian)	98.03	98.08	98.1	**99.16**
pl (Polish)	97.62	97.57	97.5	**99.05**
pt (Portuguese)	97.90	98.07	98.2	**98.86**
sl (Slovenian)	96.84	98.11	98.0	**99.44**
sv (Swedish)	96.69	96.70	97.3	**99.17**
Avg.	96.40	96.65	96.6	**98.08**
el (Greek)	-	98.24	97.9	**98.89**
et (Estonian)	-	91.32	92.8	**97.07**
ga (Irish)	-	91.11	91.0	**94.27**
hu (Hungarian)	-	94.02	94.0	**97.72**
ro (Romanian)	-	91.46	89.7	**96.64**
ta (Tamil)	-	83.16	88.7	**91.10**
Avg.	-	91.55	92.4	**95.95**

Table 5.5: POS tagging results (accuracy). All models use the gold-standard word segmentations. We use 27 corpora from the universal dependencies (1.2) and predict the UPOS tag. As Yasunaga et al. (2018), we split into high-resource (top) and low-resource languages (bottom).

our FAME method can also improve upon already powerful transformer representations. In domain-specific concept extraction, we outperform related work by 1.5 F_1-points on average. This shows that our approach works across languages and domains. As shown in Table 5.4, our models consistently set the new state of the art for domain-specific concept extraction tasks.

Those effects are also reflected in the POS tagging results, as shown in Table 5.5, even though the differences are smaller for this task. We consistently set the new state of the art for all 27 languages from the UD 1.2 corpus. In particular, we can observe remarkable differences using our method for the low-resource languages.

Model	TREC-6	TREC-50	GARD
Xu et al. (2020)	96.2	92.0	84.9
Roberts et al. (2014)	-	-	80.4
Xia et al. (2018)	98.0	-	-
FAME (our model)	**98.2**	**93.0**	**87.90**

(a) Question classification.

Model	Dev	Test
CAT	86.01±.22	85.43±.11
AVG	86.33±.33	85.57±.31
ATT	86.21±.22	85.64±.35
FAME	**86.57±.12**	**85.89±.18**

(b) Natural language inference.

Table 5.6: Results for the sentence classification tasks (accuracy), namely question classification (left) and NLI (right). For NLI, we compare against the models of Kiela et al. (2018) using four embeddings. Statistical significant differences between FAME and ATT are underlined.

5.6.2 Results for Sentence Classification

The previous experiments evaluated our feature-based adversarial meta-embeddings for various information extraction tasks on the token level. In this section, we use our meta-embeddings method in text classification models and show that this approach is also applicable to sentence-level tasks.

Question Classification. Similar to sequence labeling, our FAME approach outperforms the existing machine-learning models on all three tested sentence classification datasets, as shown in Table 5.6a. This demonstrates that our approach is generally applicable and can be used for different tasks beyond the token level.[5]

Natural Language Inference. To investigate our feature-based attention and adversarial training and show that it can easily be implemented into existing models, we extend the models of Kiela et al. (2018) for NLI with our FAME method.[6]

Table 5.6b provides the results in comparison to the baseline approaches. Our model shows statistically significant differences to the existing meta-embeddings.[7] Similar to Kiela et al. (2018), we observe that the attention-based meta-embeddings (ATT) are not always superior to the unweighted averaging (AVG). However, including our features and the adversarial training lead to consistent improvements.

[5]Note that a rule-based system (Madabushi and Lee, 2016) achieves 97.2% accuracy on TREC-50. However, this requires high manual effort tailored towards this dataset and is not directly comparable to learning-based systems.

[6]We use their code provided at `https://github.com/facebookresearch/DME` [last accessed March 5, 2022.]. Note that our numbers slightly differ from the numbers reported by Kiela et al. (2018) as they used six embeddings. However, the two MT-based embeddings (Hill et al., 2015) are not accessible any longer, as stated in personal correspondence with the authors in 2020.

[7]The state-of-the-art model for SNLI with 91.9 test accuracy is based on fine-tuning BERT (Zhang et al., 2020) but does not use combinations of different embeddings. Thus, we implemented our methods in the model proposed by Kiela et al. (2018)

Model	NER				Concept Extraction			POS (subset)		
	en	*de*	*es*	*nl*	CLIN$_{en}$	CLIN$_{es}$	SOFC$_{en}$	*et*	*ga*	*ta*
FAME (w/ fine-tuning)	94.11	92.28	89.90	95.42	90.08	92.68	83.68	97.07	94.27	91.10
FAME (w/o fine-tuning)	93.43	91.96	88.86	93.28	89.23	91.97	81.85	96.03	91.47	89.58
– features	93.37	91.66	88.37	92.98	89.07	91.42	81.48	95.81	90.20	88.73
– adversarial (ATT)	93.22	91.52	88.16	92.46	88.87	91.33	81.31	95.19	87.79	87.93
– attention (AVG)	92.38	90.14	88.44	92.37	88.69	90.23	80.28	93.20	86.95	87.73
– sum, mapping (CAT)	91.00	90.54	85.40	88.51	87.97	90.66	80.08	91.63	86.32	84.51

Table 5.7: Ablation study results for sequence labeling. We underline our FAME models without fine-tuning for which we found statistically significant differences to the attention-based meta-embeddings (ATT).

5.7 Analysis

We finally analyze the different components of our proposed FAME model by investigating, i.a., ablation studies, attention weights, and low-resource settings.

5.7.1 Ablation Study on Model Components

Table 5.7 provides an ablation study on the different components of our FAME model for exemplary sequence-labeling tasks.

First, we ablate the fine-tuning of the embedding models as we found that this has a large impact on the number of parameters of our models (538M vs. 4M) and, as a result, on the training time (cf., Section 5.4.1). Our results show that fine-tuning does have a positive impact on the performance of our models, but our approach still works very well with frozen embeddings. In particular, our non-finetuned FAME model is competitive to a fine-tuned XLM-R model (see Table 5.3) and outperforms it on 3 out of 4 languages for NER. Second, we ablate our two newly introduced components (features and adversarial training) and find that both of them have a positive impact on the performance of our models across tasks, languages, and domains.

With successively removing components, we obtain models that actually correspond to baseline meta-embeddings, as shown in the second column of the table. Our method without features and adversarial training, for example, corresponds to the baseline attention-based meta-embedding approach (ATT). We are further removing the attention function yields averaging meta-embeddings (AVG). Finally, we also evaluate another baseline meta-embedding alternative, namely concatenation (CAT). Note that concatenation leads to a very high-dimensional input representation and, therefore, requires more parameters in the next neural network layer, which can be inefficient in practice.

Statistical Significance. To show that FAME significantly improves upon the attention-based meta-embeddings, we report statistical significance with paired permutation testing with 2^{20} permutations and a significance level of 0.05. between those two models (using our

dimensions/ granularities	Same (300 dim.) Word	Subword	Different (300 & 100 dim.) Word	Subword
ATT	89.27	88.00	88.60	88.16
+ FEAT	89.28 (+.01)	88.62 (+.62)	88.64 (+.04)	88.42 (+.26)
+ ADV	89.34 (+.07)	88.31 (+.31)	89.23 (+.63)	88.44 (+.28)

Table 5.8: Effect of our proposed methods on embeddings of different granularities (word vs. subword) and dimensions (same vs. different dim.). ATT: attention-based meta-embeddings, FEAT: feature-based attention function, ADV: adversarial training of mapping. We add the differences between our methods and ATT.

Attention function	F_1	(Δ)
no features	88.0	
all features	88.62	(+.62)
– shape	88.65	(+.65)
– frequency	88.61	(+.61)
– length	88.45	(+.45)
– shape embedding	88.34	(+.34)

Table 5.9: Ablation study of the features as used in our FAME models. We test the exclusion of single features from the attention function.

method without fine-tuning for a fair comparison). Table 5.7 shows that we find statistically significant differences in six out of ten settings.

5.7.2 Influence of Embedding Granularities and Dimensions

Next, we perform an analysis to show the effects of our method for embeddings of different dimensions and granularities and support our motivation that our contributions help in those settings. As a testbed, we perform Spanish concept extraction and utilize the embeddings published by Grave et al. (2018) and Gutiérrez-Fandiño et al. (2021) as they allow us to isolate the desired effects nicely.

In particular, they published pairs of embeddings (all having 300 dimensions) that were trained on the same corpora. The first embeddings are standard word embeddings, and the second embeddings are subword embeddings with out-of-vocabulary functionality. As both were trained on the same data, we can isolate the effect of embedding granularities in a first experiment. In addition, Gui et al. (2017) published smaller versions with 100 dimensions that were trained under the same conditions. We use those in a second experiment to analyze the effects of combining embeddings of different dimensions.

The results are shown in Table 5.8. We find that adversarial training becomes particularly important whenever differently-sized embeddings are combined, i.e., when the model needs to learn mappings to larger dimensions.

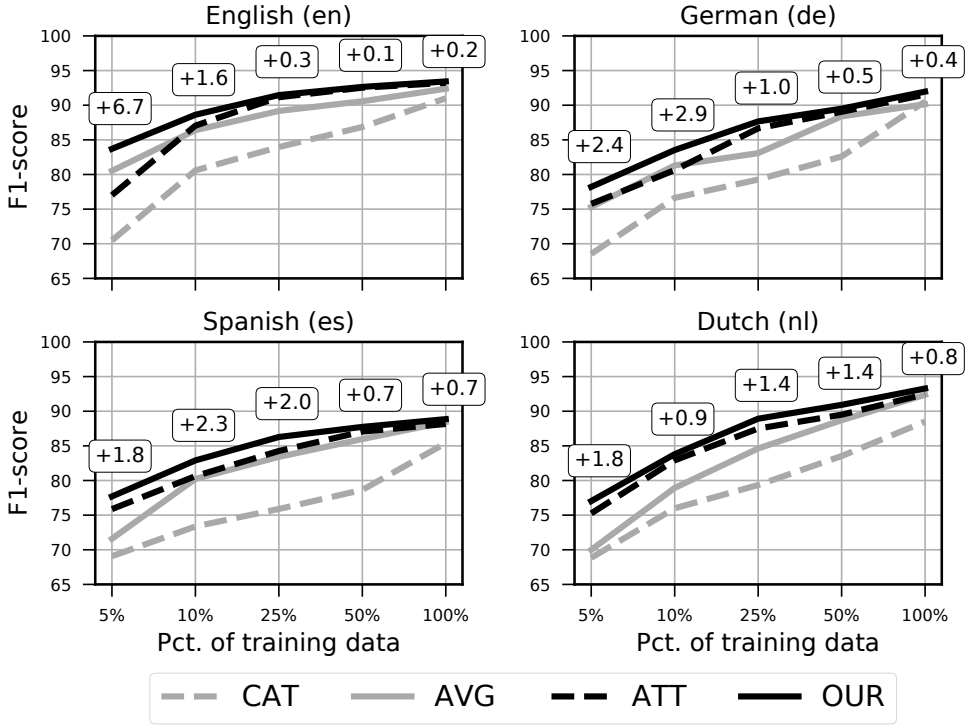


Figure 5.2: Performance for different training set sizes. The highlighted numbers display the difference between our FAME model without fine-tuning and the attention-based meta-embeddings (ATT). Further, we compare to the baseline methods averaging (AVG) and concatenation (CAT) of embeddings.

Further, we see that the inclusion of our proposed features helps substantially in the presence of subword embeddings. The reason might be that with sets of both word-based and subword-based embeddings, it gets harder to decide which embeddings are useful (e.g., word-based embeddings for high-frequency words) and should, thus, get higher attention weights. Our features have been designed in a way to explicitly guide the attention function in those cases, e.g., by indicating the frequency of a word. In addition, Table 5.9 shows an ablation study on our different features for this testbed setting. We see that shape and frequency information are the most important features, as excluding either of them reduces performance the most by more than 0.5 F_1 points.

5.7.3 Application in Low-Resource Settings

As we observed large positive effects of our method for low-resource languages (Table 5.5), we now perform a study to investigate this topic further. We simulate low-resource scenarios by artificially limiting the training data of the CONLL NER corpora to different percentages of all instances. The results are visualized in Figure 5.2. We find that the differences between the standard attention-based meta-embeddings (ATT) and our FAME method get larger with fewer training samples, with up to 6.7 F_1 points for English when 5% of the training data is used, which corresponds to roughly 600 labeled sentences. This

	Input Dim.	News$_{En}$ F_1
Single embeddings		
Character	50	77.02
BPEmb	100	86.37
fastText	300	90.45
XLM-R	1024	89.23
All embeddings		
CAT	1474	91.00
FAME	1024	93.43
Fine-tuned transformer		
XLM-R	1024	92.12
CAT	1474	92.75
FAME	1024	**94.11**

Table 5.10: English NER results for different embeddings and their combinations in our attention-based meta-embeddings. We see, that the combination of multiple embeddings outperforms all models leveraging only single embeddings.

behavior holds for all four languages and highlights the advantages of our method when only limited training data is available. An interesting future research direction is the exploration of FAME for real-world low-resource domains and languages.

5.7.4 Analysis of Embedding Methods

We studied the performance of each embedding method in isolation. The results are shown in Table 5.10 and indicate that fastText and XLM-R embeddings are the best options in this setting. This observation is also reflected in the attention weights assigned by the FAME model (see Figure 5.4). In general, fastText and XLM-R embeddings get assigned the highest weights. This highlights that the attention-based meta-embeddings are able to perform a suitable embedding selection and reduce the need for manual feature selection.

The combination of all embeddings is better than every single embedding, which shows the importance of combining multiple embeddings. In particular, the FAME model outperforms concatenation by a large margin regardless whether the transformer is fine-tuned.

5.7.5 Analysis of Attention Weights

Figure 5.3 provides the change of attention weights from the average for the domain-specific embeddings for a sentence from the clinical domain. It shows that the attention weights for the clinical embeddings are higher for in-domain words, such as “mg”, “PRN” (which stands for “pro re nata”) or “PO” (which refers to “per os”) and lower for general-domain words, such as “every”, “6” or “hours”. Thus, FAME is able to recognize the value of domain-specific embeddings in non-standard domains and assigns attention weights accordingly.

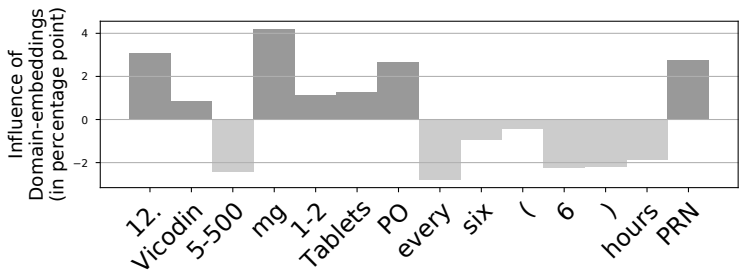


Figure 5.3: Changes in influence of domain-specific embeddings on meta-embeddings. The model prefers domain-specific embeddings for in-domain words.

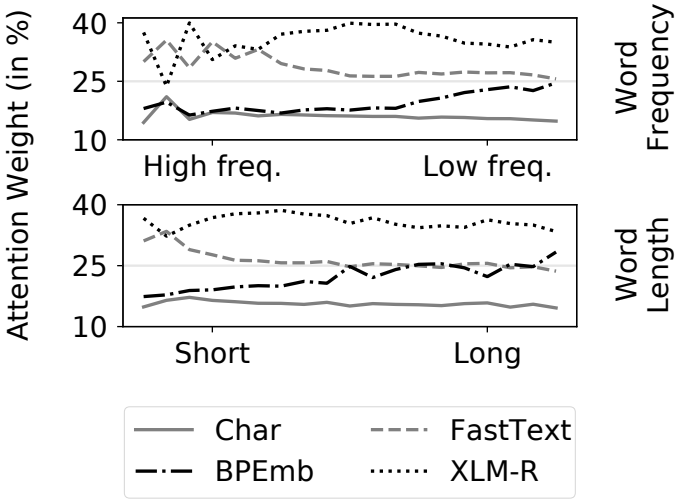


Figure 5.4: Attention weights assigned by the FAME model for the CLIN$_{en}$ corpus grouped by the features word frequency (above) and length (below).

Figure 5.4 shows how attention weights change for frequency and length features as introduced in Section 5.3.2. In particular, it demonstrates that subword-based embeddings (BPEmb and XLM-R) get more important for long and infrequent words, which are usually not well covered in the fixed vocabulary of standard word embeddings.

5.7.6 Analysis of Adversarial Training

We show that adversarial training is also beneficial and boosts performance in a monolingual case when combining multiple embeddings. The embeddings were trained independently from each other. Thus, the individual embedding spaces are clearly separated. Adversarial training shifts all embeddings closer to a common space by scaling, rotating and moving the individual embedding spaces as shown in Figure 5.5, which is important if the average is taken for the attention-based meta-embeddings approach.

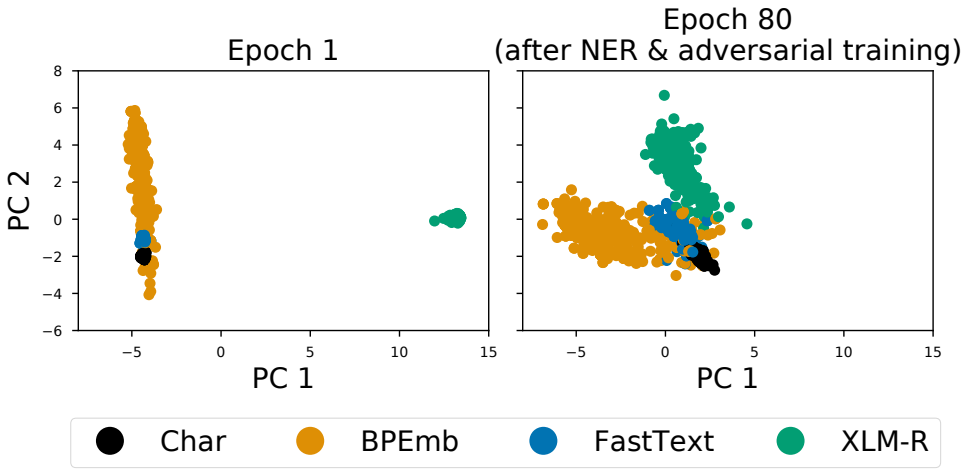


Figure 5.5: The meta-embeddings space before (left) and after NER and adversarial training (right). The embedding mappings are learned with gradient reversal to move the embedding spaces closer together.

	Clinical	Financial	Literature	Materials	CySec (cross-domain)
Clinical BERT	86.8	79.1	60.4	74.8	51.0
Financial BERT	66.1	83.6	37.5	65.8	31.9
Literature BERT	83.8	81.3	72.4	77.1	56.7
Materials BERT	86.2	80.4	66.8	79.6	56.3
Concatenation (All)	87.1	80.9	70.7	79.1	55.5
Meta-embed. (All)	**88.5**	**84.0**	**73.3**	**81.2**	**57.6**

Table 5.11: Results for cross-domain concept extraction with meta-embeddings.

5.7.7 Study: Domain-Specific Transformers

As shown in the previous sections, our meta-embeddings can be used to combine domain-specific and general-domain input representations. In this section, we combine transformer embeddings from various domains in a single model. Namely, we perform experiments on clinical (Uzuner et al., 2011), financial (Alvarado et al., 2015), literature (Bamman et al., 2019) and materials-science corpora (Friedrich et al., 2020), for which domain-specific BERT models exists. We conduct experiments for each BERT model in isolation and their combinations through concatenation and attention-based meta-embeddings. For this, we train standard sequence-labeling models on each of these corpora. In addition, we perform experiments on a cybersecurity corpus (Phandi et al., 2018) without domain-specific BERT models in a cross-domain setting.

The results are given in Table 5.11. Naturally, the domain-specific models perform best in their target domain. In particular, the financial BERT model overfits its target domain, where it achieves an outstanding performance but performs worse in all other domains. While the concatenation is not always better than the domain-specific model, the attention-

Original Sentence	6. Oxycodone-Acetaminophen 5-325 mg Tablets every 4-6 hours.
Spacy	6 . Oxycodone-Acetaminophen 5-325 mg tablets every 4-6 hours .
Native mBPE	0. oxyc od one - acet amin ophen 0-000 mg tablets every 0-0 hours .
Native XLM-R	6. O xy co done - Ac eta mino phen 5 -3 25 mg tablets every 4-6 hours .
Subword Union	6. O x y c o d one - Ac et a min o phen 5 -3 25 mg tablets every 4-6 hours .
Subword Boundaries	6. Oxycodone - Acetaminophen 5-325 mg tablets every 4-6 hours .

Table 5.12: Examples of different tokenization methods our for meta-embeddings models.

based meta-embeddings consistently outperform the concatenation and all single models for all domains, as they are able to dynamically weight embeddings according to their importance for the target domain. This helps to leverage the overfitting models like Financial BERT and more general models. Similar behavior can be observed for the cross-domain experiments for cybersecurity. Here, the meta-embeddings are able to utilize all embeddings and reduce the need for manual embedding selection.

5.7.8 Study: Meta-Embeddings on Subword-Level

The meta-embeddings presented earlier in this chapter depend on external tokenization and are applied on the word level. However, as shown in Chapter 4, sequence-labeling models based on transformers can be improved when operated on the native subword level. Unfortunately, the subword tokenization usually differs for each embedding used in the meta-embeddings layer, and the XLM-R model cannot process the subword tokenization of the byte-pair-encoding embeddings without major adaptations. Thus, the embeddings for subword-based transformers like XLM-R or *CLIN-X* that come with a pre-trained subword tokenizer are computed over words. Then, the subword vectors are aggregated, or only a single subword is taken for the word-level embedding.

Tokenization Methods. Therefore, we investigate the question of how two subword-based embeddings, namely XLM-R (250.000 tokens vocabulary) and multilingual byte-pair encoding embeddings (mBPE, 1 million tokens vocabulary), can be used in meta-embeddings on subword level. A model based on XLM-R and mBPE can, i.a., use the following tokenization schemes:

- **Spacy**: Tokenize the sentence using Spacy (Honnibal et al., 2017). The embedding vectors are generated by averaging all subword embedding vectors for each word.
- **Native mBPE**: Tokenize the sentence according to the multilingual BPE model. This implementation lowercases all characters and replaces numbers with *0*.
- **Native XLM-R**: Tokenize the sentence using XLM-R's pre-trained tokenizer.

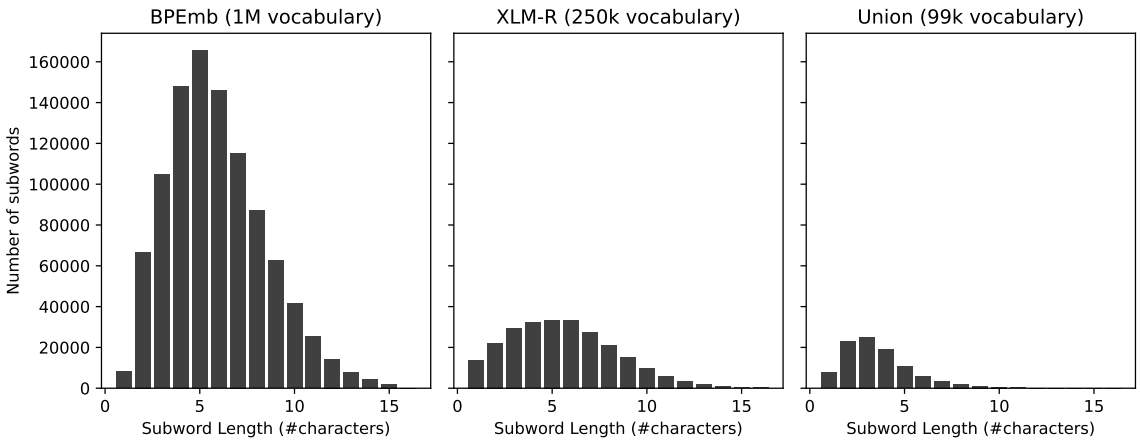


Figure 5.6: Visualization of overlapping subwords contained in XLM-R and mBPE.

Embeddings	Tokenziation	F_1
XLM-R	Spacy	85.60 ± 0.231
mBPE	Spacy	83.10 ± 0.344
XLM-R + mBPE	Spacy	87.72 ± 0.290
XLM-R	Native XLM-R	$\mathbf{87.87 \pm 0.592}$
XLM-R + mBPE	Subword Union	85.27 ± 0.335
XLM-R + mBPE	Subword Boundaries	86.40 ± 0.310

Table 5.13: Results for applying meta-embeddings on subword level using different tokenization strategies.

- **Subword Union**: We can restrict the vocabularies of mBPE and XLM-R to their union, i.e., we only use subwords that can be found in both vocabularies.

- **Subword Boundaries**: Instead of relying on external tokenization methods like Spacy, we can tokenize the sentence based on the common word boundaries generated by both models. For example, both models agree that the *O* of *Oxycodone* starts a new word and that *e* ends this word. As there is no further overlap inside the subwords, we use *Oxycodone* as the resulting token and generate the embedding vectors by averaging the subwords. Note, that for this example, the tokenization is fairly similar to Spacy. However, it allows for more-finegrained subwords when these can be found inside both vocabularies. For example, assuming *Aceta* and *minophen* are covered by the first embedding and *Aceta*, *mino*, *phen* are covered by the second embedding, this method would generate the two subwords *Aceta* and *minophen*.

Word segmentations of all methods for an example sentence are given in Table 5.12.

Experiments. Then, we train an XLM-R model for concept extraction on the PHARMACONER concept extraction tasks (Gonzalez-Agirre et al., 2019) on native subword level

and compare it to a model leveraging meta-embeddings on word level with external tokenization or other methods. We train five models with different random seeds per setting. The results are displayed in Table 5.13.[8] We see that there are almost no differences between the native subword level and meta-embeddings on word level. Both methods improve over transformers on word level, and we can conclude that operating transformers on subword level, as well as adding additional embeddings, improve performance. With this, the question remains of how we can utilize both improvements jointly.

The subword union performs comparable to a single XLM-R embedding but worse than the previous methods, as the union is very restrictive and often leads to a fallback to single characters. This results in a loss of expressiveness as nearly no advantages of subwords are used in this almost-character-based model. As shown in Figure 5.6, the union of both embeddings results in fewer and smaller subwords compared to each individual vocabulary.

Our second method, the detection of overlapping subword boundaries, performs better than the union but does not match the performance of the XLM-R on native subword level as well as the meta-embeddings on word level. Nonetheless, this is the first step of applying meta-embeddings on the subword level, and it removes the need for external tokenization. An interesting future work direction might concern the robust training of subword embeddings, as subword-based methods are not robust w.r.t. to compositionality (Aguilar et al., 2021), i.e., a subword embedding cannot be easily replaced by two or more fine-grained subwords contained in it which limits our approach.

5.8 Conclusions

In this chapter, we proposed feature-based adversarial meta-embeddings (FAME) to combine several embeddings effectively. The features are designed to guide the attention layer when computing the attention weights, in particular for embeddings representing different input granularities, such as subwords or words. Adversarial training helps to learn better mappings when embeddings of different dimensions are combined. We demonstrate the effectiveness of our approach on a variety of sentence classification and sequence-tagging tasks across languages and domains and set the new state of the art for POS tagging in 27 languages, for domain-specific concept extraction on three datasets, for NER in two languages, as well as on two question classification datasets. Our analysis shows that our approach is particularly successful in low-resource settings. Moreover, meta-embeddings can be used to efficiently combine multiple domain-specific transformers in cross-domain settings. A future direction is the evaluation of our method on sequence-to-sequence tasks or document representations.

[8] Note, that in contrast to Table 5.4, we do not finetune the transformer, nor train on the validation set.

Chapter 6

Predicting Auxiliary Embeddings

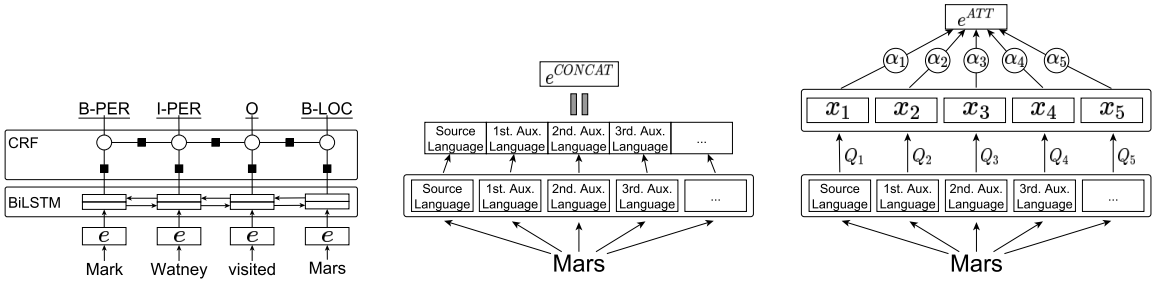
This chapter explores meta-embeddings for combining and including embeddings from other languages, as recent work demonstrated that not only different embeddings from different training methods (see Chapter 5), but also embeddings from related languages could improve the performance of sequence tagging models even for monolingual applications. Specifically, in this chapter, we investigate whether the best auxiliary language can be predicted based on language distances and show that the most related language is not always the best auxiliary language. Further, we show that attention-based meta-embeddings as introduced in Chapter 5 can effectively combine pre-trained embeddings from different languages for sequence tagging. This chapter is based on our publication on auxiliary embeddings (Lange et al., 2020a).

6.1 Introduction

State-of-the-art methods for sequence-tagging tasks, such as named entity recognition and part-of-speech tagging, exploit embeddings as input representation. Recent work suggested including embeddings trained on related languages as auxiliary embeddings to improve model performance: Catalan and Portuguese embeddings, for instance, help NER models on Spanish-English code-switching data (Winata et al., 2019a). This chapter analyzes whether auxiliary embeddings should be chosen from related languages or if embeddings from more distant languages could also help.

For this, we revisit current language distance measures (Gamallo et al., 2017) and adapt them to the embeddings and training data used in our experiments. We investigate whether we can predict the best auxiliary language based on those language distance measures. Our results suggest that no strong correlation exists between language distance and performance and that even less related languages can be a good choice as auxiliary languages.

In our experiments, we explore both available monolingual and multilingual pre-trained byte-pair encoding embeddings (Heinzerling and Strube, 2018) and Flair embeddings em-



(a) BiLSTM-CRF. **(b)** Concatenation. **(c)** Meta Embedding.

Figure 6.1: Overview of our model architecture (left). The embedding combination e can be either computed using the concatenation e^{CONCAT} (middle) or the meta embedding method e^{ATT} (right).

beddings (Akbik et al., 2018). For combining monolingual subword embeddings from different languages, we investigate two different methods: the concatenation of embeddings and the use of attention-based meta-embeddings (Kiela et al., 2018; Winata et al., 2019a).

We perform experiments for NER and POS tagging in five languages and show that meta-embeddings are a promising alternative to the concatenation of additional auxiliary embeddings as they learn to decide on the auxiliary languages in an unsupervised way. Moreover, the inclusion of embeddings from many languages is often beneficial, and meta-embeddings can be effectively used to leverage a larger number of embeddings and achieve new state-of-the-art performance on all five POS tagging tasks. Finally, we propose guidelines to decide which auxiliary languages one should use in which setting.

6.2 Related Work

This section will discuss the related work on auxiliary languages for NLP and language distance measures to select suitable languages. Related work and more details on the combination of different embeddings via meta-embeddings are given in Chapter 5. In this chapter, we compare the inclusion of auxiliary languages via concatenation to the dynamic combination with attention in meta-embeddings.

Auxiliary Languages. Winata et al. (2019a) proposed to include additional embeddings from closely-related languages to improve NER performance in code-switching settings, e.g., it was shown that Catalan and Portuguese embeddings help for Spanish-English NER. In a later work, it was shown that also more distant languages could be beneficial (Winata et al., 2019b), but only tests in the special setting of code-switching NER were performed. So far, no connection between the relatedness of languages and the performance increase was analyzed. In contrast, our work shows that the inclusion of auxiliary languages increases performance in monolingual settings. We analyze whether language distance measures can be used to select the best auxiliary language in advance.

Language Distance Measures. Gamallo et al. (2017) proposed to measure distances between languages by using the perplexity of language models trained on one language and applied to another language. Campos et al. (2020) used a similar method to retrace changes in multilingual diachronic corpora over time. Another popular measure of similarity is based on vocabulary overlap, assuming that similar languages share a large portion of their vocabulary (Brown et al., 2008).

6.3 Model Architectures

We follow the model architectures from Chapters 3 and 5 and use BiLSTM-CRF models for sequence tagging. The main differences relate to the input layer, as we include embeddings from different languages in these experiments.

Embeddings Each input word is represented with a pre-trained word vector. We experiment with byte-pair encoding embeddings (Heinzerling and Strube, 2018) with 300 dimensions and a vocabulary size of 200k tokens for all languages. Second, we experiment with Flair embeddings. For this, we use the embeddings provided by the Flair framework (Akbik et al., 2018) with 2048 dimensions for each language model resulting in a total embedding size of 4096 dimensions. We use these embedding methods, as for both of them pre-trained embeddings are publicly available for all the languages we consider.[1]

Combination of Embeddings As we experiment with multiple word embeddings, we compare two combination methods: a simple **concatenation** e^{CONCAT} and attention-based **meta-embeddings** e^{ATT} as shown in Figure 6.1b and 6.1c. We refer to Section 3.3.1 for a detailed description of the concatenation and to Section 5.3 for the meta-embedding method.

Hyperparameters and Training The bidirectional LSTM has a hidden size of 256 units. For training, we use stochastic gradient descent with a learning rate of 0.1 and a batch size of 64 sentences. The learning rate is halved after three consecutive epochs without improvement on the development set. We apply dropout with a probability of 0.1 after the input layer.

6.4 Experimental Setup

We perform NER and POS experiments in five languages: German (*de*), English (*en*), Spanish (*es*), Finnish (*fi*), and Dutch (*nl*). Note that we assume at least a character overlap to use auxiliary embeddings from another language. Thus, languages with a different character

[1] `https://github.com/flairNLP/flair` and `https://nlp.h-its.org/bpemb/` [last accessed March 5, 2022.]

set, e.g., Asian languages, cannot be used in this setting. Future work could investigate the inclusion of languages with different character sets, e.g., by using bilingual dictionaries or machine translation.

For NER, we use the CONLL 2002/03 datasets (Tjong Kim Sang, 2002; Sang and Meulder, 2003) and the FiNER corpus (Ruokolainen et al., 2020). For POS tagging, we experiment with the universal dependencies treebanks.[2] For each language, we report results for the following methods:

Monolingual (Mono). Only embeddings from the source language were taken for the experiments. This is the baseline setting.

Multilingual (Multi) We also compare our results to multilingual embeddings which have been successfully used in monolingual settings as well (Heinzerling and Strube, 2019). To ensure comparability, we use the multilingual versions of BPEmb and Flair, which were trained simultaneously on 275 and 300 languages, respectively.

Mono + X. A second set of embeddings from a different language X is concatenated with the original monolingual embeddings. While for this, typically embeddings from a related language are chosen, we report results for all language combinations and investigate in particular whether relatedness is necessary for improvement.

Mono + All & Meta-Embeddings. We also experiment with the combination of all embeddings from all languages from our experiments. In this setting, we use all six embeddings (five monolingual embeddings and the multilingual embeddings) and combine them either using concatenation (Mono + All) or meta-embeddings.

We have chosen these settings that are mainly based on monolingual embeddings, as the current state-of-the-art for named entity recognition is based on monolingual Flair embeddings (Akbik et al., 2018). In addition, multilingual embeddings, such as multilingual BERT (Devlin et al., 2019) tend to perform worse than their monolingual counterparts[3] in monolingual experiments. For completeness, we include experiments with multilingual embeddings, as mentioned before.

6.5 Results and Analysis

In this section, we will report our sequence-tagging results and analyze how language similarity measures can be used to select the best auxiliary languages.

[2]We predict the UPOS tag from the following UD v2.0 treebanks: de_gsd, en_ewt, es_gsd, fi_tdt, nl_alpino.

[3]`https://github.com/google-research/bert/blob/master/multilingual.md` [last accessed March 5, 2022.]

NER	*de*	*en*	*es*	*fi*	*nl*
Monolingual	79.78 ± .49	86.78 ± .15	78.99 ± .91	78.00 ± .87	78.91 ± .42
Multilingual	75.37 ± .87	86.52 ± .34	78.33 ± .47	77.41 ± .86	77.49 ± .45
Mono + Multi	81.13 ± .46	**88.01 ± .27**	80.32 ± .50	81.44 ± .36	81.15 ± .43
Mono + *de*	-	87.46 ± .19	79.79 ± .74	80.31 ± .21	81.31 ± .15
Mono + *en*	80.92 ± .29	-	80.48 ± .56	81.22 ± .26	80.84 ± .30
Mono + *es*	80.29 ± .20	87.37 ± .30	-	80.80 ± .83	80.62 ± .39
Mono + *fi*	81.10 ± .36	87.94 ± .17	79.91 ± .82	-	80.65 ± .48
Mono + *nl*	81.25 ± .14	87.38 ± .22	**80.93 ± .25**	80.67 ± .49	-
Mono + All	81.52 ± .33	87.70 ± .06	80.63 ± .34	82.07 ± .33 †	81.73 ± .26 †
Meta-Embeddings	**81.75 ± .50** †	87.87 ± .23	80.84 ± .52	**83.12 ± .12** †	**82.13 ± .50** †
POS	*de*	*en*	*es*	*fi*	*nl*
Monolingual	93.02 ± .11	94.17 ± .09	96.23 ± .04	92.84 ± .13	94.01 ± .21
Multilingual	92.19 ± .20	94.10 ± .06	96.01 ± .07	91.95 ± .11	93.35 ± .22
Mono + Multi	93.40 ± .08	95.11 ± .07	**96.54 ± .03**	94.70 ± .12	94.94 ± .13
Mono + *de*	-	95.11 ± .09	96.43 ± .13	94.43 ± .18	94.70 ± .09
Mono + *en*	93.26 ± .11	-	96.52 ± .06	94.45 ± .14	94.80 ± .12
Mono + *es*	93.31 ± .13	95.03 ± .09	-	94.48 ± .14	94.79 ± .17
Mono + *fi*	93.41 ± .12	94.97 ± .04	96.34 ± .08	-	94.92 ± .13
Mono + *nl*	93.52 ± .10	94.99 ± .08	96.41 ± .07	94.42 ± .08	-
Mono + All	**93.60 ± .14** †	**95.40 ± .04** †	96.46 ± .09	**95.61 ± .08** †	95.31 ± .08
Meta-Embeddings	93.51 ± .08	95.36 ± .10 †	96.48 ± .06	**95.61 ± .11** †	**95.34 ± .14** †

Table 6.1: Results of NER (F_1, above) and POS (Accuracy, below) experiments with BPE embeddings. †: statistically significant to the best Mono + X model; N: closest auxiliary language (d_{MV}); N: best auxiliary language (performance)

6.5.1 Results for Sequence Labeling

Following Reimers and Gurevych (2017), we report all experimental results as the mean of five runs and their standard deviation in Table 6.1 for experiments with byte-pair encoding embeddings. The results with Flair embeddings can be found in Table 6.2 In contrast to the BPE experiments, we do not include multilingual embeddings in the Mono + All and meta-embedding versions of Flair. The reason is prior experiments in which multilingual embeddings led to reduced performance. This is also reflected in the poor performance of the multilingual Flair embeddings alone. It seems that the multilingual BPE embeddings are more effective in downstream tasks than the multilingual Flair embeddings. We performed statistical significance testing to check if the concatenation (Mono + All) and meta-embeddings models are better than the best Mono + X model. We used paired permutation testing with 2^{20} permutations and a significance level of 0.05 and performed the Fischer correction following Dror et al. (2017).[4]

[4]We take the model with median performance on the development set for significance testing.

NER	*de*	*en*	*es*	*fi*	*nl*
Monolingual	82.66 ± .11	89.98 ± .11	85.08 ± .68	83.38 ± .31	85.68 ± .27
Multilingual	66.21 ± .79	82.87 ± .24	77.87 ± .93	73.95 ± .74	77.44 ± .52
Mono + Multi	82.95 ± .21	90.04 ± .11	84.70 ± .50	83.46 ± .37	86.04 ± .28
Mono + *de*	-	90.24 ± .19	85.16 ± .23	84.23 ± .22	85.82 ± .22
Mono + *en*	83.27 ± .36	-	85.53 ± .20	84.10 ± .26	86.73 ± .09
Mono + *es*	82.85 ± .34	90.14 ± .13	-	83.88 ± .31	86.16 ± .09
Mono + *fi*	83.10 ± .45	90.14 ± .09	85.06 ± .64	-	86.14 ± .31
Mono + *nl*	82.79 ± .24	90.18 ± .15	85.77 ± .27	83.65 ± .31	-
Mono + All	83.43 ± .29	90.29 ± .18	85.48 ± .78	84.32 ± .32	86.43 ± .33
Meta-Embeddings	83.90 ± .14 †	90.70 ± .29 †	86.18 ± .35	85.09 ± .30 †	86.58 ± .58
POS	*de*	*en*	*es*	*fi*	*nl*
Monolingual	94.72 ± .07	96.28 ± .05	97.08 ± .03	97.52 ± .03	96.48 ± .11
Multilingual	92.82 ± .20	93.69 ± .07	96.06 ± .13	92.98 ± .10	94.85 ± .11
Mono + Multi	94.72 ± .13	96.29 ± .04	97.04 ± .05	97.52 ± .05	96.77 ± .02
Mono + *de*	-	96.41 ± .07	97.11 ± .08	97.64 ± .04	96.62 ± .06
Mono + *en*	94.71 ± .04	-	97.13 ± .12	97.52 ± .06	96.49 ± .09
Mono + *es*	94.67 ± .06	96.36 ± .03	-	97.48 ± .03	96.61 ± .13
Mono + *fi*	94.65 ± .05	96.38 ± .03	97.14 ± .05	-	96.68 ± .05
Mono + *nl*	94.64 ± .03	96.31 ± .07	97.06 ± .04	97.51 ± .04	-
Mono + All	94.64 ± .10	96.48 ± .05	97.11 ± .04	97.52 ± .06	96.54 ± .20
Meta-Embeddings	94.78 ± .09	96.49 ± .03 †	97.18 ± .07	97.82 ± .03 †	96.83 ± .13 †

Table 6.2: Results of NER (F_1, above) and POS (Accuracy, below) experiments with Flair embeddings. It uses the same markup as Table 6.1.

For meta-embeddings, we found statistically significant differences in 12 out of 20 settings (6 with BPEmb, 6 with Flair) against the best monolingual + X model, while we found statistically significant differences for Mono + All in only 7 out of 20 cases. This suggests that meta-embeddings are superior to monolingual models with one auxiliary language as well as to the concatenation of all embeddings. Further, we found that the combination of monolingual and multilingual byte-pair encoding embeddings is always superior to either monolingual or multilingual embeddings alone for both tasks. Even though the multilingual embeddings have seen many languages during pre-training, they can still benefit from the high performance of monolingual embeddings and vice versa. As the meta-embeddings assign attention weights for each embedding, we can inspect the importance the models give to the different embeddings. An analysis for an example sentence can be found in Section 6.5.3.

Rank	*de*				*en*				*es*				*fi*				*nl*			
	d_P	d_P^*	d_V	d_V^*	d_P	d_P^*	d_V	d_V^*	d_P	d_P^*	d_V	d_V^*	d_P	d_P^*	d_V	d_V^*	d_P	d_P^*	d_V	d_V^*
1	*nl*	*nl*	*en*	*nl*	*nl*	*nl*	*de*	*fi*	*en*	*nl*	*en*	*en*	*de*	*nl*	*en*	*en*	*de*	*de*	*en*	*en*
2	*en*	*en*	*nl*	*en*	*es*	*fi*	*nl*	*nl*	*nl*	*en*	*de*	*nl*	*nl*	*de*	*de*	*nl*	*en*	*en*	*de*	*de*
3	*fi*	*fi*	*es**	*fi*	*de*	*de*	*fi*	*es*	*fi*	*de*	*fi*	*fi*	*en*	*en*	*es**	*de*	*fi*	*fi*	*es**	*es*
4	*es*	*es*	*fi**	*es*	*fi*	*es*	*es*	*de*	*de*	*fi*	*nl*	*de*	*es*	*es*	*nl**	*es*	*es*	*es*	*fi**	*fi*

Table 6.3: Overview of language rankings according to the distance measures used in our experiments. Languages marked with * are ranked the same.

Rank	d_{MV}				
	de	*en*	*es*	*fi*	*nl*
1	*nl*	*nl*	*en*	*en*	*de**
2	*en*	*fi*	*nl*	*de*	*en**
3	*fi*	*de*	*fi*	*nl*	*fi*
4	*es*	*es*	*es*	*es*	*es*

Table 6.4: Language rankings according to the majority voting distance d_{MV} that combines multiple language similarity measures from Table 6.3.

6.5.2 Analysis of Language Distances

To evaluate how useful language distances are for predicting the best auxiliary language, we compare rankings based on language distances and the observed performance rankings based on Table 6.1. For this, we take the language distance from Gamallo et al. (2017), which is based on language modeling perplexity PP of unigram language models LM applied to texts of foreign languages CH. Lower language model perplexities on a foreign dataset indicate higher language relatedness.

$$d_P(L1, L2) = \text{PP}(\text{CH}_{L2}, \text{LM}_{L1}) \tag{6.1}$$

We also test language similarities based on vocabulary overlap with $W(L1|L2)$ being the number of words of $L1$ which are shared with $L2$ and $N(L1)$ the number of words of $L1$ shared with other languages.

$$d_V(L1, L2) = \frac{W(L1|L2) + W(L2|L1)}{2 \cdot \min(N(L1), N(L2))} \tag{6.2}$$

For our experiments, we further adapt d_P to use the perplexity of the Flair forward language models on the test data provided by Gamallo et al. (2017) and call it d_P^*. Similarly, we adapt d_V^* to compute the overlap of words in our training data. Note that both variants, d_P^* and d_V^*, are based on properties from either our model or training data and are, therefore, specific to our setting. Finally, we create a ranking d_{MV} which combines the rankings from

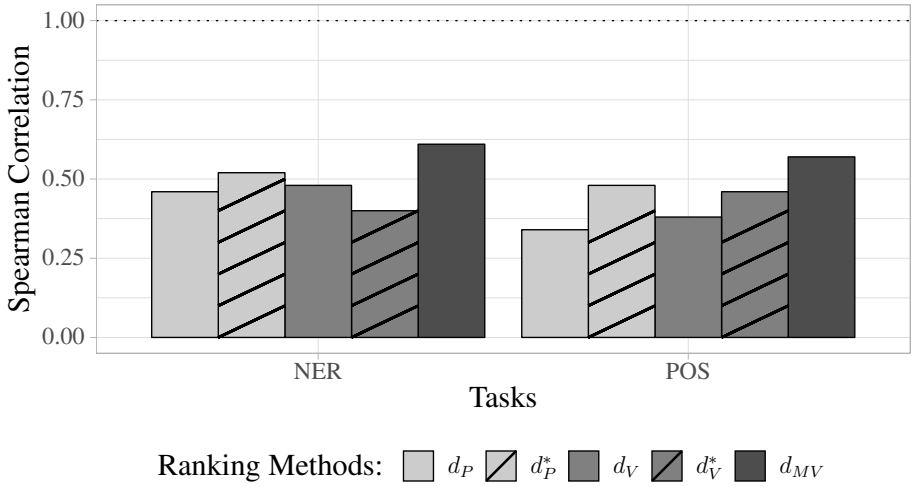


Figure 6.2: Spearman's rank correlation between language distance and model performance rankings for NER and POS tasks for different language distances.

d_P, d_P^*, d_V, d_V^* with majority voting. The ranking of d_{MV} is provided in Table 6.4, the rankings of the individual distance measures are given in Table 6.3.

To analyze the correlation between language distance measures and the performance of our model, we compute Spearman's rank correlation coefficient between the real rankings based on performance and predicted rankings from our language distances. The results are shown in Figure 6.2. We conclude that predicting the auxiliary language ranking is a hard task and see that the most related language is not always the best auxiliary language in practice (cf., Table 6.1). This holds in particular for POS tagging, where the performance differences of models are quite small.

In general, d_P^* shows a higher correlation with model performance than d_P and d_V, indicating that not only word overlap plays a role but also context information. The majority voting d_{MV} achieves the highest correlation and often predicts the best auxiliary language for NER models using byte-pair encoding embeddings. However, the actual ranking of all languages does not match the performance ranking, which results in a relatively low correlation with only a little above 0.5.

6.5.3 Analysis of Attention Weights

As the meta-embeddings assign attention weights for each embedding, we can inspect the importance the models give to the different embeddings. Figure 6.3 shows the assigned weights for an English sentence. In general, the model assigns the most weight to the English embeddings. However, we observe an increased weight for German and the multilingual embedding for the German word *Bayerische*. Even though *Vereinsbank* is also a German word, the model focuses more on English for this word, probably because the subword *bank* has the same meaning in English.

Embedding	Average Weight (All texts)	Now	Bayerische	Vereinsbank	broadens	share	offer	Average Weight (Example)
en	18.0	18.9 (+0.9)	17.6 (-0.4)	20.2 (+2.2)	19.1 (+1,1)	19.5 (+1.5)	20.2 (+2.2)	19.3 (+1.3)
de	16.6	16.9 (+0.3)	18.8 (+2.2)	17.6 (+1.0)	17.5 (+0.9)	16.8 (+0.2)	16.5 (+0.1)	17.4 (+0.8)
Multi	14.7	13.5 (-1.2)	18.9 (+4.2)	15.5 (+0.8)	15.2 (+0.5)	15.0 (+0.3)	15.4 (+0.7)	15.6 (+0.9)
Other (avg.)	16.9	16.9 (+/- 0)	14.9 (-2.0)	15.6 (-1.3)	16.0 (-0.9)	16.2 (-0.7)	16.0 (-0.9)	15.9 (-1.0)

Figure 6.3: Learned attention weights of the meta-embeddings model with byte-pair encoding embeddings for English NER. "Other" refers to the average weights of Spanish, Finish and Dutch embeddings which are less relevant for this example. Darker colors indicate higher relative weights.

6.5.4 Study: Increased Number of Parameters

To investigate whether the performance increase comes from the increased number of parameters rather than the inclusion of more embeddings, we also investigated including the same embedding type twice (Mono + Mono). However, we found that this does not help: The performance is comparable to the monolingual baseline. Thus, the performance increase for Mono + X actually comes from additional information provided by the embeddings of the auxiliary language.

6.5.5 Practical Guide

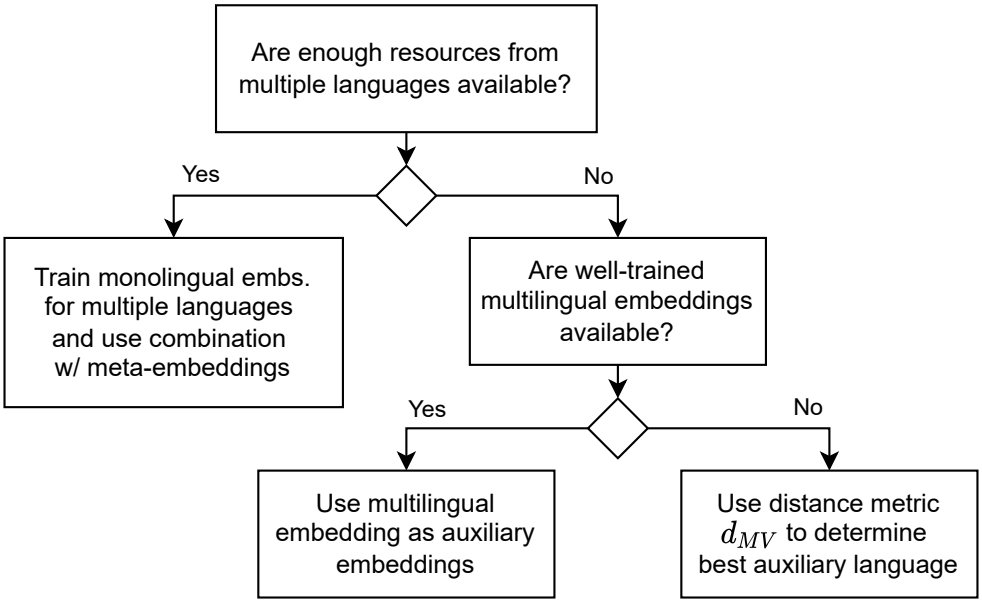


Figure 6.4: Proposal for auxiliary embedding selection.

Finally, we propose a small guide in Figure 6.4 to decide which auxiliary languages one can use to improve performance over monolingual embeddings. Depending on the available amount of data, it is recommended to train multiple monolingual embeddings and combine them using meta-embeddings, which was observed to be the best method in our experiments. If not enough data is available to train monolingual embeddings, the best solution would be the inclusion of multilingual embeddings, assuming the existence of high-quality embeddings, such as multilingual byte-pair encoding embeddings. If none of the above applies, language distance measures, in particular the combination of multiple distances, can help to identify the most promising auxiliary embeddings. Despite not always predicting the best model, the predicted auxiliary language often led to improvements over the monolingual baseline in our experiments.

6.6 Conclusions

In this chapter, we investigated the benefits of auxiliary languages for sequence tagging. We showed that it is hard to predict the best auxiliary language based on language distances. Instead, we showed that meta-embeddings could leverage multiple embeddings effectively for those tasks without the need for manual embedding selection. Finally, we proposed a guide on how to decide which method of including auxiliary languages one should use.

Chapter 7

Predicting Sets of Transfer Sources

Deep neural networks and large language models are known for requiring large amounts of training data. Thus, there is a growing body of work to improve performance in low-resource settings, for example, by transferring knowledge via embeddings from different languages, as shown in the last chapter. This chapter focuses on the transfer of neural models across low-resource tasks and non-standard domains. We introduce new methods to compute similarities between different datasets and predictors to select the most promising transfer sources to select suitable transfer sources in the clinical domain.

In low-resource settings, transfer learning methods, such as our proposed model transfer which will be introduced in this chapter, can help to overcome a lack of labeled data for many tasks and domains. However, predicting useful transfer sources is a challenging problem, as even the most similar sources might lead to unexpected negative transfer results. Thus, ranking methods based on task and text similarity — as suggested in prior work — may not be sufficient to identify promising sources. We propose a new approach to automatically determine which and how many sources should be exploited to tackle this problem. For this, we study the effects of model transfer on sequence labeling across various domains and tasks and show that our methods based on model similarity and support vector machines are able to predict promising sources, resulting in performance increases of up to 24 F_1 points. This chapter is based on our publication on transfer source selection (Lange et al., 2021c).

7.1 Introduction

Only little labeled data is available for many natural language processing applications in non-standard domains. This even holds for high-resource languages like English (Klie et al., 2020). The most popular method to overcome this lack of supervision is transfer learning from high-resource tasks or domains. This includes the usage of resources from similar domains (Ruder and Plank, 2017), domain-specific pretraining on unlabeled text

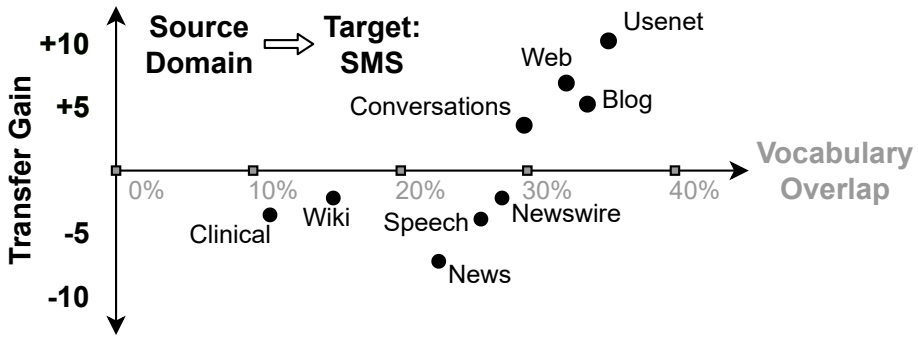


Figure 7.1: Observed transfer gains by transferring models from a source corpus to SMS texts. The domains are sorted by their vocabulary overlap to the target. Positive transfer can be expected for unseen domains with an overlap of at least 30 %.

(Gururangan et al., 2020), and the transfer of trained models to a new domain (Bingel and Søgaard, 2017). While having the choice among various transfer sources can be advantageous, it becomes more challenging to identify the most valuable ones as many sources might lead to negative transfer results, i.e., actually reduce performance (Pruksachatkun et al., 2020).

Current methods to select transfer sources are based on text or task similarity measures (Dai et al., 2020; Schröder and Biemann, 2020). The underlying assumption is that similar texts and tasks can support each other. Existing work predicted the best sources for language model pretraining (Dai et al., 2020), multitask-training (Schröder and Biemann, 2020) or cross-task model transfer (Vu et al., 2020) by analyzing text and task similarity measures or training data (Ruder and Plank, 2017). An example is depicted in Figure 7.1 which shows the correlation of transfer gain and corpus similarity based on vocabulary overlap for transfers from different sources to a fixed target corpus. However, current methods typically consider the text and task similarity in isolation, which limits their application in transfer settings where both the task and the text domain change.

Thus, as a first major contribution, this chapter proposes a new model similarity measure that represents text and task similarity jointly. By learning a mapping between two neural models, it captures the similarity between domain-specific models across tasks. We perform experiments for different transfer settings, namely zero-shot model transfer, supervised domain adaptation, and cross-task transfer across a large set of domains and tasks. Our newly proposed similarity measure successfully predicts the best transfer sources and outperforms existing text and task similarity measures.

As a second major contribution, we introduce a new method to automatically determine which and how many sources should be used in the transfer process, as the transfer can benefit from multiple sources. Our selection method overcomes the limitations of current transfer methods, which solely predict single sources based on rankings. We show the benefits of transfer from sets of sources and demonstrate that support vector machines are

able to predict the best sources across domains and tasks. This improves performance with absolute gains of up to 24 F_1 points and effectively prevents negative transfer.

7.2 Related Work

Domain adaptation and transfer learning are typically performed by transferring information and knowledge from a high-resource to a low-resource domain or task which is lacking labeled data (Daumé III, 2007; Ruder, 2019b). Recent approaches can be divided into two groups: (i) model transfer (Ruder and Plank, 2017), by reusing trained task-specific weights (Vu et al., 2020) or by first adapting models on the target domain before training the downstream task (Gururangan et al., 2020; Rietzler et al., 2020) and (ii) multi-task training (Collobert and Weston, 2008) where multiple tasks are trained jointly by learning shared representations (Peng and Dredze, 2017; Meftah et al., 2020). We follow the first approach in this chapter by training on related domains and tasks and transferring the model to the target domain. This is also called sequential multi-task training (see Section 2.4.2).

For transfer learning, the selection of sources is utterly important to exploit the potential of transfer learning. With the ever-growing body of possible transfer sources, selecting the wrong source can lead to severe negative transfer (Pruksachatkun et al., 2020) and, thus, source selection has to be performed carefully. Text and task similarity measures (Ruder and Plank, 2017; Bingel and Søgaard, 2017) are used to select the best sources for cross-task transfer (Jiang et al., 2020), multi-task transfer (Schröder and Biemann, 2020), cross-lingual transfer (Chen et al., 2019) and language modeling (Dai et al., 2020). Alternatively, neural embeddings for corpora can be compared (Vu et al., 2020). In prior work, the set of domains is usually limited to news and Wikipedia, and the focus is on the single-best source. In contrast, we exploit sources from a larger set of domains and also explore the prediction of sets of sources, as using multiple sources is likely to be beneficial, as also shown by Parvez and Chang (2021) contemporaneously to this work.

7.3 Similarity Measures and Predictors

In this section, we describe the sequence tagger model and similarity measures along with metrics for the evaluation. Finally, we introduce our new model similarity measures and prediction method for sets of transfer sources.

Terminology We consider two dimensions of datasets: the task T, which defines the label set, and the input text coming from a specific domain D. We thus define a dataset as a tuple $\langle T, D \rangle$, and specify in our experiments which of the two dimensions are changed.

7.3.1 Similarity Measures

We apply the following measures to rank sources according to their similarity with the target data.

Baselines. We use the most promising domain similarity measures reported by Dai et al. (2020). The most simple baseline is based on the *Dataset size* (Bingel and Søgaard, 2017). This assumes that large corpora contain general knowledge, which is useful for transfer. However, this measure is static and does not incorporate corpora-specific information. In contrast, the following similarity measures compute similarity scores based on the contents of two corpora,

- *Target vocabulary overlap* is the percentage of unique words from the target corpus covered in the source corpus.

$$TVO(D_s, D_t) = \frac{|V_{D_s} \cap V_{D_t}|}{|V_{D_t}|} \tag{7.1}$$

  where V_D is the vocabulary of D, i.e. the set of unique tokens. In contrast to vocabulary overlap, this is an asymmetric measure. *Annotation overlap* is a special case considering only annotated words.

- We also experiment with the *Language model perplexity* (Baldwin et al., 2013) between two datasets. For this, a language model, in our case a 5-gram LM with Kneser–Ney smoothing (Heafield, 2011) as used by Dai et al. (2020), is trained for each source domain and tested against the target domain. The resulting perplexity gives hints at how similar these domains are, i.e., a lower perplexity indicates similarity between domains.

$$(D_s, D_t) = \sum_{i=1}^{|D_t|} P(D_t^i)^{-\frac{1}{|D_{s_i}|}} \tag{7.2}$$

  with $P(D_t^i)$ being the probabilities assigned by the source language model to sentence i from the target dataset.

- *Jensen-Shannon divergence* (Ruder and Plank, 2017) compares the term distributions between two texts, which are probability distributions that capture the frequency of words. It is similar to vocabulary overlap, as it describes the textual overlap but is based on distributions instead of sets of terms.

$$JSD(t(D_s)||t(D_t)) = \frac{1}{2}(t(D_s)||t(D_t)) + \frac{1}{2}(t(D_t)||t(D_s)) \tag{7.3}$$

where $t(D)$ is the term distribution from dataset D and $D_{KL}(P||Q)$ is the Kullback-Leibler divergence between two probability distributions P and Q:

$$D_{KL}(P||Q) = -\sum_{x \in X} P(x) \log(\frac{Q(x)}{P(x)}) \tag{7.4}$$

- A *Text embedding* (Vu et al., 2020) can be computed by extracting the feature vectors of a neural model. For this, the output of the last layer is averaged over all words in the dataset. This vector then represents the textual domain. The distance between two vectors is computed by using cosine similarity (cos.sim.).

$$TS(D_s, D_t) = \text{cos.sim.}(E_m(D_s), E_m(D_t)) \tag{7.5}$$

  where $E_m(d)$ is the average embedding of dataset d using model m.

- The *Task embedding* (Vu et al., 2020) takes a labeled source dataset and computes a representation based on the Fisher Information Matrix, which captures the change of model parameters w.r.t. the computed loss. This method assumes that similar tasks require similar parameters changes. We use the code released by Vu et al. (2020) to compute task embeddings from the different components of our BERT models and similarly use reciprocal rank fusion (Cormack et al., 2009) to combine these.

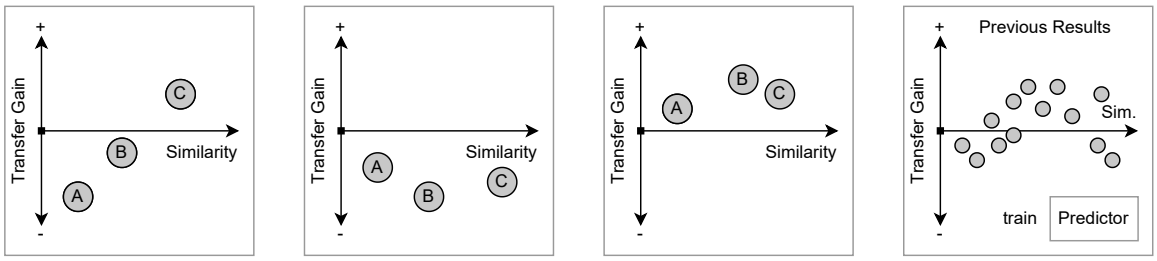
Our Model Similarity. As a new strong method, we propose *Model similarity* that is able to combine domain and task similarity. For this, feature vectors f for a target dataset t are extracted from the last layer of two models m_s, m_t which have been trained on the source and target datasets, respectively. The features are then aligned by a linear transformation W, a learned mapping, between the feature spaces using the Procrustes method (Schönemann, 1966) to minimize their pointwise differences:

$$\arg\min_{W} |W(f(m_s, t)) - f(m_t, t)| \tag{7.6}$$

The resulting transformation W is the optimal mapping between the features $f(m_s, t)$ to $f(m_t, t)$. If both feature spaces are the same, W would be the identity matrix I, i.e., no change is required for the transformation. Larger changes indicate dissimilarity. Thus, our model similarity measure (MoS) between the two models is the distance of the mapping W and the identity matrix I:

$$MoS(m_s, m_t) = |W - I| \tag{7.7}$$

Similar mappings have been used for the alignment of different embedding spaces (Artetxe et al., 2018) as they inherently carry information on the relatedness between models. See Section 2.3.4 for more details.



(a) Top-1 predicts the best source.

(b) Top-1 predicts too many sources.

(c) Top-1 predicts too few sources.

(d) Our predictor is trainable.

Figure 7.2: Illustration of different transfer behaviours. The transfer gain might correlate with the dataset similarity (a). However, the single-source prediction Top-1 might be too liberal (b) or too restrictive (c). To solve this, we train a dynamic predictor model on previous observations (d).

7.3.2 Prediction Methods for Sets of Sources

These similarity measures can be applied to create rankings and select similar datasets. However, they still have a major shortcoming in practice: None of them provides explicit insights when positive or negative transfer can be expected.

Typically, the most similar source is selected for training based on a given similarity measure. This might introduce only a low risk of selecting a negative transfer source, but it also cannot benefit from further positive transfer sources. Examples are given in Figure 7.2. The single-source transfer is straightforward and works well when the similarity values correlate with the transfer gain like in Figure 7.2a. However, this does not always hold in practice. Sometimes, even the most similar sources lead to unexpected negative transfer, and all sources might decrease performance as in Figure 7.2b. Then, a useful prediction method should also predict this behavior and recommends no transfer at all. On the other hand, taking only one transfer source might be too restrictive, and many sources can contribute to positive transfer, as Figure 7.2c shows. Using multiple sources of them can boost performance even further. As a solution to this problem, we propose source predictor models that perform a dynamic selection of transfer sources based on previous observations (cf. Figure 7.2d).

We refer to the selection of the single-best source as Top-1. We also test its extension to an arbitrary selection of the n best sources denoted by Top-n. However, it is unclear how to choose n, and increasing n comes with the risk of including sources that lead to negative transfer results.

As a solution, we propose two methods that predict whether positive transfer is likely for a given distance between datasets: The first method models the prediction as a 3-class classification task, and the second one as a regression task predicting the transfer gain. For classification, we split the transfer gain g into the three classes positive ($g \geq \theta$), neutral ($|g| < \theta$) and negative ($g \leq -\theta$) based on a pre-defined threshold θ. (In our experiments, we set $\theta = 0.5$.) We introduce the neutral class for classification to cope with small transfer

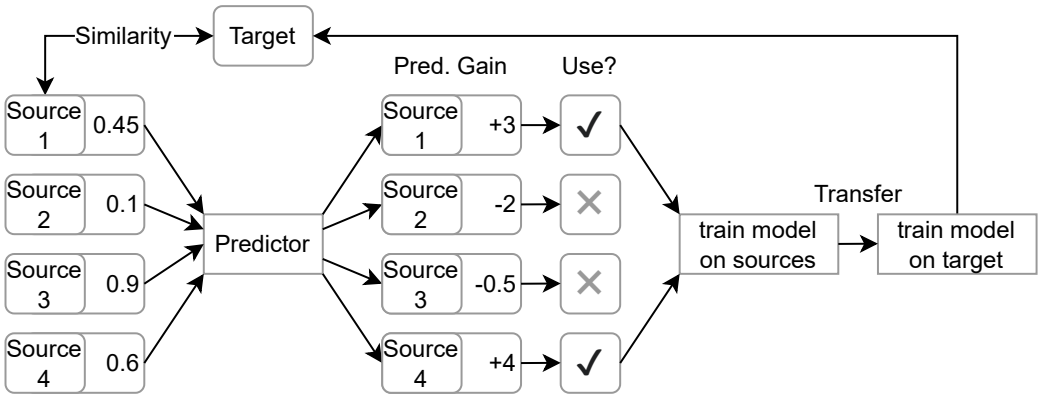


Figure 7.3: Overview of the selection process with our dynamic prediction model.

gains $|g| < \theta$ that do not provide additional information but would increase the training time.

To solve these tasks, we propose to use support vector machines (SVM) for classification (-C) and regression (-R) and compare to k-nearest-neighbour classifiers (k-NN) as well as logistic and linear regression in our experiments.[1] For each method, the input to the model is a similarity value between source and target. The training label is either the observed transfer gain (for regression) or the corresponding class (for classification) for the source-target pair. A trained model can then be used to predict which kind of transfer can be expected, given a new similarity value.[2] The predictions for a target and a set of sources can then be used to select the subset of sources with expected positive transfer. This process is visualized in Figure 7.3.

7.4 Experimental Setup

In this section, we introduce the tasks, datasets and models used in our experiments. Moreover, we describe the different transfer methods we will study.

7.4.1 Tasks and Evaluation Metrics

We perform experiments on 33 datasets for three tasks: Named entity recognition, part-of-speech tagging, and temporal expression extraction (TIME). For TIME, we use the English corpora described by Strötgen and Gertz (2016) and the ACE'05 corpus (Walker et al., 2006) split into domain-specific subcorpora. For POS, we run experiments on the four publicly available universal dependency datasets (Nivre et al., 2016). For NER, we distinguish two cases: shared label sets and different label sets. For NER with shared label sets,

[1] We use sklearn implementations (Pedregosa et al., 2011)

[2] Other similarity measures can be included by modeling each value as a different input dimension. However, we found no significant improvements by including multiple measures.

T	Corpus	Domain D	# Labels	# Train / Dev / Test sentences
NER	CONLL (Sang and Meulder, 2003)	News	9	14,987 / 3,466 / 3,684
	I2B2-CLIN (Uzuner et al., 2011)	Clinical concepts	7	13,052 / 3,263 / 27,625
	I2B2-ANON (Stubbs and Uzuner, 2015)	Clinical anonymization	47	45,443 / 5,439 / 32,587
	WNUT-16 (Strauss et al., 2016)	Twitter posts	21	2,394 / 1,000 / 3,850
	WNUT-17 (Derczynski et al., 2017)	Social media	13	3.394 / 1,009 / 1.287
	WNUT-20 (Tabassum et al., 2020)	Wetlab protocols	37	8.444 / 2,862 / 2,813
	LITBANK (Bamman et al., 2019)	Literature	13	5.549 / 1.388 / 2.973
	SEC (Alvarado et al., 2015)	Financial	9	825 / 207 / 443
	SOFC (Friedrich et al., 2020)	Materials science	9	490 / 123 / 263
NER& POS	GUM-ALL (Zeldes, 2017)	All (GUM)	23/17	3,883 / 960 / 2,060
	GUM-ACAD	Academic	23/17	321 / 81 / 173
	GUM-BIO	Biography	23/17	434 / 106 / 233
	GUM-FICT	Fiction	23/17	576 / 144 / 309
	GUM-INT	Interview	23/17	599 / 150 / 321
	GUM-NEWS	News	23/17	360 / 91 / 194
	GUM-RED	Reddit	23/17	500 / 126 / 269
	GUM-TRAV	Travel	23/17	431 / 108 / 232
	GUM-WHOW	Wikihow	23/17	612 / 154 / 329
POS	EWT (Silveira et al., 2014)	Blog. Email. Social	17	12,514 / 1,998 / 2.074
	LINES (Ahrenberg, 2015)	(non-)Fiction. spoken	17	2,738 / 912 / 914
	PARTUT (Sanguinetti and Bosco, 2014)	Legal. News. Wikipedia	17	1,781 / 156 / 153
Temporal Expressions	TIMEBANK (UzZaman et al., 2013)	News	9	2,557 / 640 / 303
	AQUAINT (UzZaman et al., 2013)	News	9	972 / 243 / 522
	ANCIENT (Strötgen et al., 2014b)	Historical Wikipedia	9	456 / 114 / 245
	WWARS (Mazur and Dale, 2010)	Wikipedia	9	2,788 / 697 / 1,494
	TIME4SMS (Strötgen and Gertz, 2013)	SMS	9	1,674 / 419 / 898
	TIME4SCI (Strötgen and Gertz, 2013)	Clinical	9	461 / 116 / 248
	I2B2-TIME (Sun et al., 2013)	Clinical	9	5,943 / 1,486 / 5,665
	ACE-ALL (Walker et al., 2006)	All (ACE-05)	9	8,958 / 2,241 / 4,802
	ACE-BC	Broadcast conversations	9	1,655 / 414 / 887
	ACE-BN	Broadcast news	9	2,087 / 522 / 1,119
	ACE-CTS	Conversational telephony	9	1,756 / 440 / 942
	ACE-NW	Newswire	9	1,172 / 293 / 628
	ACE-UN	Usenet	9	1,168 / 292 / 626
	ACE-WB	Webblog	9	1,120 / 280 / 600

Table 7.1: Overview of dataset domains and their sizes used in the transfer experiments.

we use the subcorpora from all domains of the GUM (Zeldes, 2017) corpus. For NER with different label sets, we use several publicly available datasets from a wide range of domains, including clinical (I2B2, Stubbs and Uzuner, 2015), social media (WNUT, Strauss et al., 2016) and material science corpora (SOFC, Friedrich et al., 2020).

For TIME tagging and for POS tagging, we use the English corpora described by Strötgen and Gertz (2016) and the four publicly available universal dependencies corpora with the UPOS tag (Nivre et al., 2016), respectively. We convert the TIMEX corpora into the BIO format for sequence tagging. For NER with different label sets, we collected several datasets from a wide range of domains, including clinical (I2B2 Stubbs et al., 2015), social media (WNUT, Strauss et al., 2016) and materials science corpora (SOFC, Friedrich et al., 2020). The GUM (Zeldes, 2017) and ACE'05 (Walker et al., 2006) corpora can be split easily into multiple domains. Thus, we perform experiments for all subcorpora. The GUM corpus has multi-layer annotations and includes named entity annotations as well. We use this to study the effects of NER transfer when the label set is shared. All datasets are listed in Table 7.1 with information on their domain and size with respect to the label set and the number of sentences. We take the last 20% and 10% of the training data as test or development data whenever no split was provided.

The metric for all experiments is micro F_1 for NER and TIME, and accuracy for POS tagging.[3] We use the difference in F_1 to measure transfer effects and also report *transfer gain* (Vu et al., 2020), i.e., the relative improvement of a transferred model compared to the single-task performance. p_t with source s target t as datasets and $p_{s\to t}$ being the transfer performance:

$$g_{s\to t} = \frac{p_{s\to t} - p_t}{p_t} \tag{7.8}$$

In Section 7.5.2, we rank sources according to their similarity to the target. These rankings are evaluated with two metrics, following Vu et al. (2020): (1) the average rank of the best performing model in the predicted ranking denoted by avg rank and (2) the normalized discounted cumulative gain (NDCG, Järvelin and Kekäläinen, 2002). The latter is a ranking measure commonly used in information retrieval, which evaluates the complete ranking. Thus, this metric is more suited for evaluating the whole ranking, while avg rank only considers the top element.

7.4.2 Models for Sequence Labeling

For sequence tagging, we follow Devlin et al. (2019) and use BERT-base-cased as the feature extractor F and a linear mapping to the label space followed by a softmax as the classifier C. All layers of the model are fine-tuned during training.

Models are trained using the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $2e-5$. The training is performed for a maximum of 100 epochs. We apply

[3]Due to the absence of a non-labeled class like 'O' in POS-tagging, accuracy and F_1 are equal in this case and we simply refer to F_1-score in all experiments.

Corpus	Pre	Rec	F_1
CONLL	90.5	91.9	91.2
WNUT-20	78.2	81.0	79.6
WNUT-17	60.1	35.9	44.9
WNUT-16	46.8	44.7	45.7
I2B2-CLIN	82.0	85.8	83.9
I2B2-ANON	94.7	93.2	94.0
SEC	76.7	87.9	81.9
LITBANK	66.1	74.5	70.0
SOFC	73.3	82.8	77.8
GUM-ACAD	46.3	58.8	51.8
GUM-BIO	61.0	72.1	66.1
GUM-FICT	62.8	72.0	67.1
GUM-INT	48.9	58.7	53.4
GUM-NEWS	43.7	52.7	47.8
GUM-RED	50.5	61.9	55.6
GUM-TRAV	37.7	51.0	43.3
GUM-WHOW	40.0	49.0	44.0
GUM-ALL	55.1	64.3	59.4

(a) Named Entity Recognition.

Corpus	Pre	Rec	F_1
TIMEBANK	75.2	76.3	75.7
AQUAINT	77.6	77.6	77.6
ANCIENT	71.8	79.6	75.5
WWARS	87.1	90.7	88.9
TIME4SMS	63.8	68.4	66.0
TIME4SCI	55.9	51.6	53.7
I2B2-TIME	72.2	76.7	74.4
ACE-BC	60.5	64.0	62.2
ACE-BN	60.3	71.5	65.4
ACE-CTS	39.0	55.6	45.8
ACE-NW	76.8	81.9	79.2
ACE-UN	56.5	65.2	60.5
ACE-WB	65.6	69.4	67.5
ACE-ALL	66.9	77.6	71.8

(b) Temporal Expression Extraction.

Corpus	Acc.
PARTUT	96.9
EWT	97.0
LINES	97.5
GUM-ACAD	95.1
GUM-BIO	96.3
GUM-FICT	96.8
GUM-INT	95.5
GUM-NEWS	95.9
GUM-RED	94.6
GUM-TRAV	94.5
GUM-WHOW	94.9
GUM-ALL	96.5

(c) POS Tagging.

Table 7.2: Single task learning performance for the three different tasks.

early stopping after five epochs without change of the F_1-score on the development set. We use the same hyperparameters across all settings. The results for all datasets without transfer are given in Table 7.2.[4]

7.4.3 Transfer Settings

In our experiments, we will study the following three transfer methods:

- **Zero-shot model transfer.** We apply a model trained on a source dataset to a target with the same task but a different domain: $\langle T_i, D_i \rangle \rightarrow \langle T_i, D_j \rangle$.
- **Supervised domain adaptation.** A model trained on a source domain is adapted to a target domain by finetuning its weights on target training data: $\langle T_i, D_i \rangle \rightarrow \langle T_i, D_j \rangle$.
- **Cross-task transfer.** For applying a model to a different task, we replace the classification layer with a randomly initialized layer and adapt it to the new target task: $\langle T_i, D_i \rangle \rightarrow \langle T_j, D_j \rangle$.[5]

[4] All our experiments are run on a carbon-neutral GPU cluster. The model training takes between 5 minutes and 8 hours depending on the dataset size on a single Nvidia Tesla V100 GPU with 32GB VRAM.

[5] We restrict the cross-task transfer to NER targets with different label sets, as the combination of all tasks quickly becomes computationally infeasible given our large number of different settings.

Task	Min.	Avg.	Max.	# Pos.	# Neg.
Zero-Shot Model Transfer					
NER	-57.3 (-37.9)	-17.7 (-10.1)	18.1 (8.0)	7 /64	56 /64
POS	-8.7 (-8.4)	-2.8 (-2.7)	1.6 (1.5)	13 /144	127 /144
TIME	-100.0 (-83.2)	-42.7 (-29.6)	38.6 (17.7)	13 /196	183 /196
Supervised Domain Adaptation					
NER	-5.2 (-2.7)	3.8 (1.9)	14.5 (6.3)	55 /64	8 /64
POS	-0.3 (-0.3)	0.4 (0.4)	1.8 (1.7)	116 /144	9 /144
TIME	-15.3 (-10.1)	3.4 (2.0)	32.7 (15.1)	133 /196	62 /196
Cross-Task Transfer					
NER→NER	-9.1 (-4.1)	-0.2 (-0.2)	6.8 (3.1)	39 /90	46 /90
POS→NER	-5.9 (-4.8)	-0.5 (-0.3)	2.6 (1.2)	42 /120	65 /120
TIME→NER	-7.2 (-3.3)	-0.9 (-0.5)	0.9 (0.6)	35 /150	100 /150

Table 7.3: Statistics on transfer gains (F_1 differences) and the number of positive and negative transfer scenarios for the three transfer settings. The average is aggregated over all domains and the 5 random seeds resulting in 210 task-specific experiments for NER and up to 780 for TIME.

7.5 Results and Analysis

This section first presents the results of the different transfer settings. Then, we analyze how well the previously described dataset and model similarity measures can be used to rank transfer sources. Finally, we evaluate our dynamic prediction methods that predict actual sets of beneficial transfer sources.

7.5.1 Analysis of Transfer Performance

Table 7.3 shows the observed performance gains compared to the single-task performance. For zero-shot model transfer, we observe severe performance drops when transferring out-of-domain models to unseen targets, compared to in-domain models (single-task training). In addition to domain-specific challenges, this setting is impaired by differences in the underlying annotation schemes.[6]

Supervised domain adaptation, i.e., adapting a model to the target domain, improves performance across all settings independent of the source domain. Table 7.3 shows that the average transfer gains are positive for all tasks and that the maximum transfer gain is 32.7 F_1 points for TIME. However, the transfer gains are highly task-depending. For example, there is almost no negative transfer and only limited positive transfer for POS tagging due to a large number of training instances and relatively small changes in domains. In contrast, the transfer gains for TIME range from -15 to +33 F_1 points.

[6]For example, the TIMEX2 (Ferro et al., 2005) and TIMEX3 (Pustejovsky et al., 2005) guidelines disagree about including preceding words in the annotated mentions as "in".

Distance	Model Transfer		Domain Adapt.		Cross -Task		Avg.	
	ρ	N	ρ	N	ρ	N	ρ	N
Vocabulary Overlap	**2.4**	**92.1**	**2.8**	88.9	6.4	84.9	3.9	88.7
Annotation Overlap	**2.4**	91.7	3.1	**89.3**	6.1	85.3	3.9	**89.1**
Dataset size	3.6	86.4	3.8	85.9	7.2	82.3	4.9	84.9
Term Distribution	2.8	90.5	4.2	87.5	6.7	85.2	4.5	87.7
LM Perplexity	3.9	85.6	3.4	88.2	5.9	84.4	4.4	86.1
Text Embedding	4.0	88.1	4.6	85.0	7.1	84.6	5.2	85.9
Task Embedding	4.1	88.5	4.7	84.8	6.6	84.5	5.1	85.6
OUR Model Similarity	2.8	90.8	3.3	88.7	**5.1**	**85.4**	**3.7**	88.3

Table 7.4: Ranking results for different similarity measures in the three transfer settings. Corpus-based measures are listed first and model-based ones below. The values displayed are the average rank of the best model (ρ) and the NDCG-score (N).

The gains for cross-task transfer are smaller than for supervised domain adaptation. While we still observe some performance increases, the average transfer gains are negative for all tasks. This shows that it is likely that the adaptation of models from other tasks will decrease performance. These results demonstrate the need for reliable similarity measures and methods to predict the expected transfer gains given the source task and domain. We will explore them in Section 7.5.2 and Section 7.5.3, respectively.

7.5.2 Results for Similarity-Based Ranking

To evaluate the prospects of different sources for model transfer, we compute the pairwise distances between all datasets using the similarity measures presented in Section 7.3.1 and rank them accordingly.

Table 7.4 shows that the text-based methods vocabulary and annotation overlap are most suited for in-task transfer, i.e., model transfer and domain adaptation, while our model similarity is most useful for cross-task transfer. This shows that task similarity alone is not the most decisive factor for predicting promising transfer sources, and domain similarity is equally or even more important, in particular, when more distant domains are considered. Our model similarity is able to capture both properties and, as a result, outperforms the task embedding in the cross-task setting and performs comparably to the text-based methods in the in-task settings. It is the best similarity measure on average across all transfer settings according to the predicted rank of the top-performing source (avg. rank) and the best neural method according to NDCG.

In general, we find that selecting only the top source(s) based on a ranking from a distance measure, as done in current research, gives no information on whether to expect positive transfer. Thus, we now explore methods to predict sets of promising sources automatically.

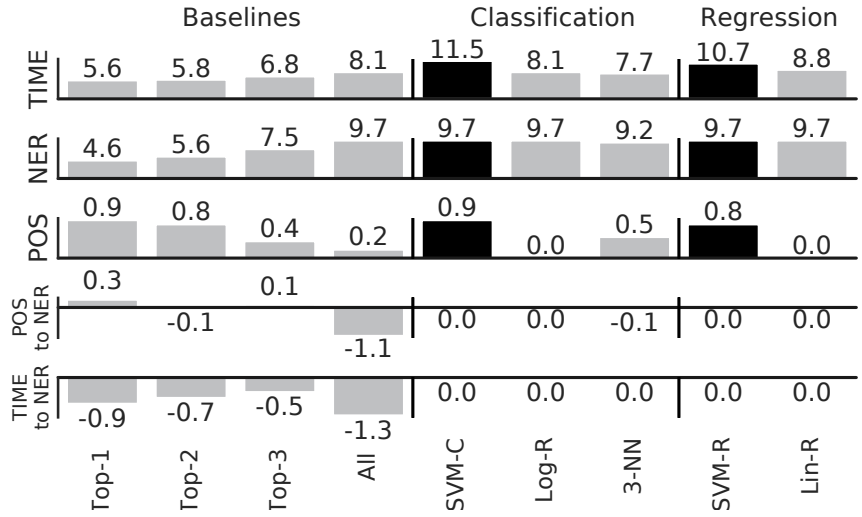


Figure 7.4: Average transfer gains using different classifiers for predicting the sets of most promising sources.

Method	Sources	F_1 (increase)
Single-Task	*No source corpora for pretraining*	60.5 F_1
Top-1	WWARS	+ 10.9
Top-2	WWARS, ACE-BN	+ 11.2
Top-3	WWARS, ACE-BN, TIMEBANK	+ 15.8
All	WWARS, ACE-ALL, TIMEBANK, AQUAINT, I2B2-TIME, ANCIENT, TIME4SCI, TIME4SMS	+ 10.2
SVM (Classifier)	WWARS, ACE-ALL, TIMEBANK, AQUAINT, TIME4SCI, TIME4SMS	**+ 24.0**
Logistic Regression	WWARS, ACE-ALL, TIMEBANK, AQUAINT, ANCIENT, TIME4SCI, TIME4SMS	+ 18.2
k-Nearest-Neighbor	WWARS, ACE-ALL, TIMEBANK, AQUAINT, I2B2-TIME, TIME4SCI, TIME4SMS	+ 17.1
SVM (Regression)	WWARS, ACE-ALL, TIMEBANK, AQUAINT, TIME4SCI, TIME4SMS	**+ 24.0**
Linear Regression	WWARS, ACE-ALL, TIMEBANK, AQUAINT, ANCIENT, TIME4SCI, TIME4SMS	+ 18.2

Table 7.5: Predicted transfer sources for TIME domain adaptation for target ACE-UN.

7.5.3 Results for Prediction of Sets of Sources

We use the methods introduced in Section 7.3.2 to predict the set of most promising sources. Then, we train a model on the combination of the selected sources and adapt it to the target.[7]

The results averaged across the different settings are visualized in Figure 7.4. Again, we observe a task-specific behavior. While NER and TIME targets benefit from training on many sources, POS tagging targets gain the most from using only one or two of the most related source domains. We find that our methods based on SVMs are able to predict this behavior and assign fewer sources for POS targets and more sources for TIME and NER settings. In particular, for TIME settings, our methods SVM-C and -R result in much higher transfer gains compared to the static ranking-based methods and other classifiers or regression models.

[7]We do not explore the NER to NER setting, as we restrict the sources to have the same set of labels. For the other tasks, we trained source combinations that were predicted by at least one model (SVM-R/C, Log-/Lin-R, k-NN) or baseline method (Top-1, Top-2, .., All). Training all possible combinations would be infeasible.

Task	Min.	Avg.	Max.
Supervised Domain Adaptation			
NER	-2.5	4.9	20.3
POS	-0.5	1.0	6.8
TIME	-14.2	17.4	174.5
Cross-Task Transfer			
NER→NER	-13.7	-0.2	30.2
POS→NER	-18.3	-2.5	12.4
TIME→NER	-14.2	-2.0	7.8

Table 7.6: Transfer gains for single-source transfer in a low-resource setting.

For example, transferring multiple sources using our SVM classifier to the ACE usenet target (see Table 7.5) increases performance from 60.5 F_1 for single-task training to 84.5 F_1 (+24.0), which is much higher than the 10.9 points increase when using the single best source or 10.2 points using all available sources.

For the cross-task experiments in the lower part of Figure 7.4, we find that even the inclusion of the single best-ranked model results in a transfer loss of -0.9 points on average for TIME→NER. In this setting, our models correctly adapt to this new challenge and predict an empty set of sources, indicating that no transfer should be performed.

7.6 Study: Low-Resource Transfer

In this section, we describe our transfer experiments in low-resource settings by limiting the training data. For our first experiments in Section 7.6.1, we follow the experimental setup from this chapter and only limit the training data. Moreover, we experiment in Section 7.6.2 with transfer between the ten clinical tasks and our *CLIN-X* language models, as introduced in Chapter 4.

7.6.1 Transfer Learning with Limited Training Data

First, we perform the transfer experiments in a limited setting, where we downsampled all training and development splits to 321 sentences, the size of the smallest corpus (GUM-ACAD) without shuffling. The test set is not changed. With this, the effects of different dataset sizes are reduced. We increased the number of epochs for early stopping to 10, as some of the models trained on the small corpora need a few epochs to get non-zero F_1-scores with the reduced AdamW warmup learning rate. All other hyperparameters and training details remain unchanged. The results are shown in Table 7.6. We can observe much larger positive transfer gains, as the data limitation can drastically reduce the single-task performance. This way, a transfer gain of more than 170 can happen, e.g., for transferring the model trained on ACE-ALL to its subcorpus ACE-CTS, due to a relatively low base score of 16 F_1 in the limited setting. At the same time, the negative transfer can get

English (I2B2)	2006	2010	2012-C	2012-T	2014
CLIN-X$_{EN}$ +OURMODEL	98.49	89.23	80.62	78.50	97.60
CLIN-X$_{ES}$ +OURMODEL	98.30	89.10	80.42	78.48	**97.62**
CLIN-X$_{ES}$ +OURMODEL +Transfer	**98.50**	**89.74**	**80.93**	**79.60**	97.46
Significant Differences	*	*	***	**	*

Spanish	CANTEMIST	MEDDOCAN	M.PROF-N	M.PROF-C	PHARMA.
CLIN-X$_{EN}$ +OURMODEL	87.72	97.57	81.36	78.53	92.36
CLIN-X$_{ES}$ +OURMODEL	**88.24**	**98.00**	81.68	**80.54**	92.27
CLIN-X$_{ES}$ +OURMODEL +Transfer	88.00	97.65	**81.88**	79.38	92.27
Significant Differences				*	

Table 7.7: Performance of our *CLIN-X* models in transfer settings (F_1). We highlight statistically significant differences between *CLIN-X*$_{ES}$ +OURMODEL with and without transfer following the R significant codes: *** p-value ≤ 0.001; ** p-value < 0.01; * p-value < 0.05. See Chapter 4 for a description of the *CLIN-X* language models and our sequence labeling architecture +OURMODEL.

much worse, in particular, for the cross-task settings. While this demonstrates the potential benefits of model transfer, it also highlights the importance of selecting a suitable transfer source, for example, by using our model similarity and dynamic prediction methods for sets of sources.

7.6.2 Transfer Learning in the Clinical Domain

As shown earlier, many NLP tasks suffer from a lack of labeled data which may be overcome by transfer learning. This includes non-standard domains like the clinical domain in particular. On the one hand, this domain has high requirements regarding the removal or masking of protected health information (PHI) of individuals (Uzuner et al., 2007; Stubbs et al., 2015) which is particularly worthy of protection and can prevent data publication. On the other hand, information extraction tasks are often specific to their target domain, and clinical concepts are only found very infrequently outside EHRs, which limits the reusability of existing resources. Possible solutions for the low-resource problem can be multi-task learning (Khan et al., 2020; Mulyar et al., 2021) or transfer learning (Lee et al., 2018; Peng et al., 2019) across similar corpora from the clinical domain. For example, Hofer et al. (2018) showed that few-shot NER in the biomedical domain could be improved by transferring weights trained on a similar task. However, transferring knowledge is particularly challenging in the clinical domain as biomedical NLP models have problems generalizing to new entities (Kim and Kang, 2021). Therefore, one has to carefully select the transfer sources, as discussed in Section 7.5, which can be addressed using our transfer methods.

In this section, we study the effects of transfer learning on our clinical datasets and models as used in Chapter 4.

Tgt.	Src. / Setting	# Training sentences 250	500	1000	2500	7500	All
I2B2-2006	No Transfer	71.24	81.06	84.15	95.49	96.89	98.34
	I2B2-2010	81.55	90.38	89.09	95.61	97.47	96.88
	I2B2-2012-C	79.28	86.5	88.71	96.75	97.92	98.23
	I2B2-2012-T	71.58	80.31	83.29	95.87	**97.97**	97.41
	I2B2-2014	**87.52**	**90.86**	**91.87**	**97.11**	97.95	**98.50**
I2B2-2010	No Transfer	65.38	74.96	82.59	85.54	88.48	89.10
	I2B2-2006	68.90	78.32	82.07	85.70	87.95	88.69
	I2B2-2012-C	**83.99**	**86.25**	**86.88**	**88.46**	**89.34**	**89.74**
	I2B2-2012-T	69.49	74.92	81.31	85.35	88.25	88.65
	I2B2-2014	72.05	79.11	82.49	85.54	87.69	88.80
I2B2-2012-C	No Transfer	69.09	73.21	75.70	78.03	80.36	80.42
	I2B2-2006	68.83	72.14	75.34	77.86	79.25	80.15
	I2B2-2010	**76.39**	**77.98**	**79.44**	**80.90**	**81.65**	**80.93**
	I2B2-2012-T	65.30	69.61	73.30	75.88	80.25	80.12
	I2B2-2014	68.67	72.56	75.39	77.96	79.98	79.83
I2B2-2012-T	No Transfer	67.49	72.67	75.44	78.00	78.33	78.48
	I2B2-2006	68.57	72.49	74.34	77.73	78.43	78.34
	I2B2-2010	68.10	74.04	**78.01**	**78.98**	**79.29**	79.60
	I2B2-2012-C	**70.17**	**75.04**	76.36	78.12	78.54	**80.03**
	I2B2-2014	69.44	72.66	75.04	77.88	78.86	79.36
I2B2-2014	No Transfer	64.96	81.61	85.74	92.70	96.08	**97.62**
	I2B2-2006	**81.50**	**85.76**	**88.96**	**93.51**	96.04	97.46
	I2B2-2010	71.72	83.55	87.81	93.18	**96.14**	97.17
	I2B2-2012-C	71.24	82.97	87.09	93.15	96.13	97.33
	I2B2-2012-T	69.12	81.25	85.08	91.35	96.02	97.00

Table 7.8: Cross-task transfer results for few-shot settings for the English clinical corpora (F_1). The predicted transfer source and the **best** models are highlighted.

First, we test statistical significance between *CLIN-X*$_{ES}$ with and without transfer learning — highlighted with asterisks in Table 7.7. We find that all differences for English are significant, while only one difference for Spanish is significant. This might indicate the complementary relationship of domain adaptation and model transfer learning. As *CLIN-X* was explicitly adapted to Spanish, additional transfer is not necessary for high-resource settings. In contrast, the cross-language domain adaptation for English can still be improved with transfer from related sources, where *CLIN-X*$_{ES}$ +OURMODEL +Transfer has also notably higher performances in 3 out of 5 settings compared to *CLIN-X*$_{EN}$, which is adapted to English.

Tgt.	Src. / Setting	# Training sentences 250	500	1000	2500	7500	All
CANTEMIST	No Transfer	51.68	59.00	67.35	77.15	**84.10**	**88.24**
	MEDDOCAN	**56.48**	59.51	**69.33**	76.57	83.43	88.00
	MEDDOPROF-N	52.06	**59.26**	67.18	**77.27**	83.05	87.74
	MEDDOPROF-C	53.94	55.41	65.71	76.65	83.20	88.00
	PHARMACONER	55.53	59.14	66.78	76.44	83.39	87.95
MEDDOCAN	No Transfer	84.00	92.01	95.28	96.48	97.20	**98.00**
	CANTEMIST	83.61	89.36	95.35	96.75	97.43	97.57
	MEDDOPROF-N	86.99	92.77	93.55	96.15	97.01	97.66
	MEDDOPROF-C	88.70	93.76	95.03	96.32	97.35	97.73
	PHARMACONER	**92.74**	**94.30**	**96.16**	**96.84**	**97.49**	97.65
M.PROF-N	No Transfer	13.99	44.28	51.24	58.95	72.54	81.68
	CANTEMIST	10.01	38.41	50.64	62.66	71.74	79.77
	MEDDOCAN	16.39	45.30	52.89	62.25	73.30	81.38
	MEDDOPROF-C	**61.29**	**68.37**	**72.83**	**72.88**	**78.04**	**81.88**
	PHARMACONER	23.72	44.91	52.90	60.53	73.35	81.07
M.PROF-C	No Transfer	16.46	24.28	47.67	54.66	68.68	**80.54**
	CANTEMIST	10.99	29.73	49.20	52.75	66.57	78.76
	MEDDOCAN	31.83	38.01	53.80	56.46	69.98	79.33
	MEDDOPROF-N	**57.46**	**57.70**	**61.56**	**64.92**	**72.37**	79.38
	PHARMACONER	22.61	35.15	50.50	53.49	69.59	79.08
PHARMA.	Sinlge-Task	67.71	76.38	81.32	87.68	91.31	92.27
	CANTEMIST	60.34	71.77	79.45	86.77	90.61	**92.35**
	MEDDOCAN	**74.48**	76.02	**82.79**	**88.39**	**91.49**	92.27
	MEDDOPROF-N	69.48	**76.44**	78.73	88.60	92.02	91.98
	MEDDOPROF-C	69.25	74.15	80.13	88.27	91.80	92.29

Table 7.9: Cross-task transfer results for few-shot settings for the Spanish clinical corpora (F_1). The predicted transfer source and the **best** models are highlighted.

Second, we simulate low-resource settings where we limit the annotated data of the target dataset between 250 labeled sentences up to 7500 sentences, roughly the size of the smallest corpus. The results are given in Table 7.8 and Table 7.9 for English and Spanish, respectively. Large positive transfer happens in most settings, particularly for the low-resource settings with up to +47.3 F_1 points for MEDDOPROF when only 250 labeled sentences are available. The improvements in the full-data scenario are below 1 F1. However, there is also negative transfer, in particular using I2B2-2012-T and CANTEMIST datasets as transfer sources often result in a negative transfer. The source selection is also crucial in low-resource scenarios, as not every source is equally beneficial. Using our model similarity measure, as introduced in Section 7.3.1, we find good transfer sources for almost all datasets in general and for low-resource scenarios in particular.

7.7 Conclusions

In this chapter, we explored different transfer settings across three sequence-labeling tasks and various domains. Our new model similarity measure based on feature mappings outperforms currently used similarity measures as it is able to capture both task and domain similarity at the same time. We further addressed the automatic selection of sets of sources as well as the challenge of potential negative transfer by proposing a selection method based on support vector machines. We can achieve performance gains of up to 24 F_1 points using this method compared to single-source transfer — as suggested in prior work — using our method. Larger gains are possible in low-resource scenarios with up to 47 F_1 for our experiments in the clinical domain.

Chapter 8

Multilingual Temporal Tagging

The detection and normalization of temporal expressions are challenging tasks and necessary pre-processing steps for many applications, such as knowledge base population, question answering, or information retrieval. Current systems either focus on the extraction of temporal expressions only without normalization or are rule-based, which severely limits the applicability in real-world multilingual settings due to the costly creation of new rules. While there have been some approaches towards normalization methods, which can be combined with ML-based extraction components (e.g., TIMEN (Llorens et al., 2012)), these require quite some effort to adapt and do hardly allow for domain-sensitive strategies for the normalization (cf., (Strötgen and Gertz, 2016)) This chapter investigates both the extraction and normalization of temporal expression with neural models across languages. We improve the extraction by aligning the languages inside a multilingual extraction model. For normalization, we propose a novel neural method for normalizing temporal expressions based on masked language modeling. Our multilingual method outperforms prior rule-based systems in many languages, particularly for low-resource languages with performance improvements of up to 35 F_1 on average compared to a state-of-the-art rule-based system. This chapter is based on our publication for temporal expression extraction (Lange et al., 2020d) and our submission for normalization (Lange et al., 2022b).

8.1 Introduction

The task of temporal tagging consists of the extraction of temporal expressions from texts and their normalization to a standard format (e.g., *May '22*: `2022-05`). More details on this task are given in Section 2.1.2.

In this chapter, we will take a look into both subtasks, the extraction and normalization of temporal expressions in the context of multilingual temporal tagging. Examples of three temporal expressions in different languages and their normalization values are given in Figure 8.1. State-of-the-art systems which natively combine both tasks are rule-based and,

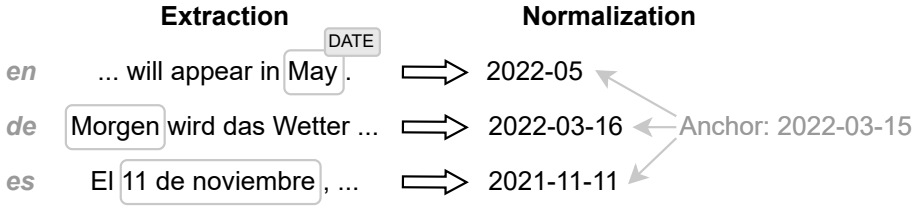


Figure 8.1: Overview of the extraction and normalization process for three temporal expressions from English, German and Spanish example sentences.

therefore, hard to transfer to new languages. Our first main contribution in this chapter is a cross-lingual extraction method based on multilingual transformers. For this, we model the extraction as sequence tagging and train a single model for many languages at once. To further improve the multilingual model, we experiment with adversarial alignment methods by creating a truly multilingual embedding space in the model in order to improve the downstream performance in cross- and multilingual settings.

The temporal expression normalization remains challenging, and no practical solution for the normalization of expressions across languages exists yet. While there are deep learning approaches for the extraction, temporal tagging as a whole is usually solved with highly specific rule-based systems.

Therefore, we propose a new multilingual normalization method that can make use of labeled data from many languages by training a neural transformer model with a masked language modeling (MLM) objective. The MLM training is becoming more popular recently due to language models like BERT (Devlin et al., 2019) and has been used for several different tasks (Sun et al., 2021). In this work, we adopt the MLM objective function for a new purpose: the normalization of expressions. To the best of our knowledge, this is the first work that uses neural networks for the normalization of temporal expression on top of the extraction outputs. For this, as detailed below, we split the normalization task into two steps: normalizing to a context-independent representation and anchoring this representation using the document context for the final disambiguation.

Our experiments in 17 languages demonstrate the robust performance of our multilingual method due to the generalization abilities of neural models. With this, we demonstrate that it is possible to achieve competitive performance with a single multilingual model for many languages at once. Further, we demonstrate that this multilingual model can be transferred to new languages, for which the model has not seen any gold-standard labels during training by applying it to unseen languages in cross-lingual experiments. For both tasks, we outperform the current state of the art for multilingual temporal tagging, HeidelTime (Strötgen and Gertz, 2015), especially for low-resource languages by more than 35 F_1 on average.

The main contributions of this chapter are a new cross-lingual extraction model (Section 8.3.1) and our novel temporal expression normalization method based on neural networks trained with a masked language modeling objective (Section 8.3.2). We conduct

an extensive set of experiments across 17 languages, demonstrating that our multilingual method is robust and works for many languages (Sections 8.5.1 and 8.5.2). Further, we explore different training and decoding strategies for our masked language model (Section 8.5.3).

8.2 Related Work

Temporal Tagging. Temporal tagging is the task of extracting temporal expressions from text and normalizing them to a standard format (Strötgen and Gertz, 2016). A detailed description of this task is given in Section 2.1.2. Most of the research on temporal tagging is focused on English (Strötgen and Gertz, 2016), but also a number of high-quality corpora were created. For example, there exists so-called timebanks i.a., for French (Bittar et al., 2011), Portuguese (Costa and Branco, 2012), and Catalan (Saurı, 2010). Nonetheless, systems for automatic temporal tagging are often language-specific and do not transfer to new languages.

Both temporal tagging tasks, the extraction and normalization, are most often solved with highly specific rule-based systems, such as SUTime (Chang and Manning, 2012) or HeidelTime (Strötgen and Gertz, 2013). However, rule-based methods are hard to transfer to new languages or text domains, as it requires a large manual effort to create new rules specific to the target language. Although work on the automatic generation of rules for English (Ding et al., 2021) or many other languages (Strötgen and Gertz, 2015) exist, the first approach is monolingual and does not transfer to other languages and the rule quality of the second approach typically does not match the high accuracy of hand-crafted rules.

In contrast to rule-based systems, neural networks are known for their ability to generalize to new targets, in particular, for cross- and multilingual applications (Rahimi et al., 2019; Artetxe and Schwenk, 2019). These works show the applicability of neural networks for cross-language extraction. Still, in the context of temporal tagging, recent works have only shown the promising performance of neural networks for the **extraction** of temporal expressions in monolingual settings (Laparra et al., 2018; Xu et al., 2019) by modeling the extraction as sequence labeling. Their application in multilingual settings remains an open research problem that is only addressed by rule-based systems.

Similarly, the **normalization** of temporal expressions is only solved using rule-based methods so far, e.g., as done by Llorens et al. (2012) or Ning et al. (2018). Alternatives to strictly rule-based systems are context-free grammars (Bethard, 2013; Lee et al., 2014) which are independent of the extraction method. However, these are even more specific towards a certain language and can hardly generalize to new languages or language families. Bethard and Parker (2016) proposed the SCATE format, which allows a more fine-grained extraction of temporal expressions. Based on this, Laparra et al. (2018) used neural networks for the extraction and a rule-based procedure for the normalization for English. While this method, in theory, is extensible to multilingual applications, no annotated data is available in the SCATE format for other languages, as it is mostly incompatible with the

dominant TimeML (Pustejovsky et al., 2005) annotation format. In contrast, our normalization approach is fully compatible with TimeML and can leverage many existing training resources.

Masked Language Modeling. The self-supervised MLM paradigm has gained a lot of attention recently (Sun et al., 2021) due to popular language models like BERT (Devlin et al., 2019). This led to active research on using MLM to solve further tasks like text classification (Brown et al., 2020), named entity recognition (Ma et al., 2021) and relation extraction (Han et al., 2021). In general, MLM is a bidirectional language modeling task. Given the left and right context of a masked word, the original word under the mask has to be predicted. For example, given a sentence with at least one mask "*The topic of Chapter 3 was **MASK** extraction*," an MLM model has to predict the most probable replacements for the masks. Likely words in this context are "*information*" or "*concept*". In this chapter, we train a transformer with an MLM objective to normalize a temporal expression to its context-independent representation.

Multilingual Embeddings. Recently, it became popular to train embedding models on resources from many languages jointly (Conneau and Lample, 2019; Conneau et al., 2020). For example, multilingual BERT (Devlin et al., 2019) was trained on Wikipedia articles from more than 100 languages. Although performance improvements show the possibility to use multilingual BERT in monolingual (Hakala and Pyysalo, 2019), multilingual (Tsai et al., 2019) and cross-lingual settings (Wu and Dredze, 2019), it has been questioned whether multilingual BERT is truly multilingual (Pires et al., 2019; Singh et al., 2019; Libovický et al., 2020). Therefore, we will investigate the benefits of aligning its embeddings in our experiments. More information on multilingual language models is given in Section 2.3.4.

Aligning Embedding Spaces. A well-known method to create multilingual embedding spaces is the alignment of monolingual embeddings (Mikolov et al., 2013b; Joulin et al., 2018). Smith et al. (2017) proposed to align embedding spaces by creating orthogonal transformation matrices based on bilingual dictionaries, which we use as a baseline alignment method for word embeddings. We provide a detailed description in Section 2.3.4.

It was shown that BERT can also benefit from alignment, i.a., in cross-lingual (Schuster et al., 2019; Liu et al., 2019a) or multilingual settings (Cao et al., 2020b). In contrast to prior work, we experiment with aligning BERT using adversarial training, which is related to using adversarial training for domain adaptation (Ganin et al., 2016), coping with bias or confounding variables (Li et al., 2018; Raff and Sylvester, 2018; Zhang et al., 2018; Barrett et al., 2019; McHardy et al., 2019) or transferring models from a source to a target language (Zhang et al., 2017a; Keung et al., 2019; Wang et al., 2019b). Similar to Chen and Cardie (2018) and our adversarial meta-embeddings approach (cf., Chapter 5), we use a multinomial discriminator in our setting.

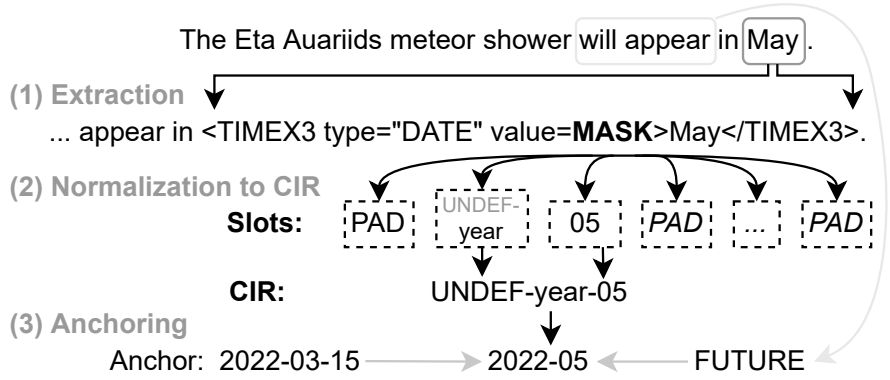


Figure 8.2: Overview of our 3-step pipeline for temporal tagging consisting of the extraction of temporal expressions (step 1), the normalization to a context-independent representation (CIR) using a slot-based masked language model (step 2) and the anchoring given a reference time (step 3).

8.3 Model Architectures

We propose to solve the task of multilingual temporal tagging in three steps as shown in Figure 8.2: (1) Extraction of temporal expressions and their types using a novel multilingual sequence tagger; (2) Normalization to a CIR for each temporal expression with our novel MLM-based normalization model; (3) Anchoring of CIRs given a reference time, e.g., using HeidelTime rules.

Our main contributions are neural models for the first and second subtasks, To the best of our knowledge, the second subtask, the normalization to a context-independent representation, has not been addressed with neural networks before. In this section, we detail all components of our approach. Information on the models that we apply for the third subtask as well as an ablation study of all different model components are given in Section 8.5.

8.3.1 Models for Extraction

We model the task of extracting temporal expressions as a sequence-tagging problem and explore the performance of state-of-the-art transformers like XLM-R and BERT, as well as recurrent neural networks with fastText embeddings. In particular, we train multilingual models that process many languages at once. In addition, we propose an unsupervised alignment approach based on adversarial training and compare it to two baseline approaches to create and improve the multilingual embedding spaces. Figure 8.3 provides an overview of the system. The different components are described in detail in the following.

Classifier. For the extraction of temporal expressions, we experiment with the XLM-R transformer as a classifier C similar to Conneau et al. (2020). Here, the transformer outputs

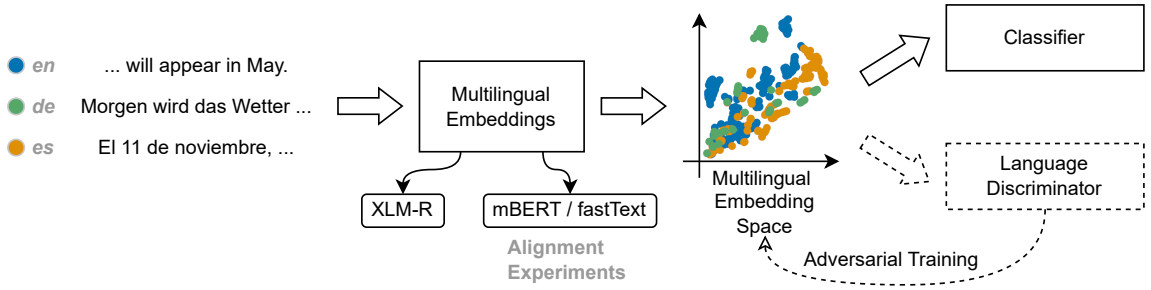


Figure 8.3: Overview of our extraction model based on a joint multilingual embeddings space. We propose to use adversarial training to align the language-specific embeddings.

are directly mapped to the label space, and a softmax function is used to compute the label probabilities instead of a CRF.

In addition, we study the creation of multilingual embedding spaces via alignment methods. For this, we use recurrent neural networks, e.g., as discussed in Chapter 3 and train BiLSTM-CRF models. As input, we experiment with two embedding methods: (i) monolingual pre-trained fastText word embeddings (Bojanowski et al., 2017),[1] and (ii) multilingual BERT (Devlin et al., 2019).[2] For BERT, we use the averaged output of the last four layers as input to the BiLSTM and fine-tune the whole model during the training of temporal information extraction. The alignment methods for these models are discussed in the following.

Alignment of Embeddings. We propose an unsupervised approach based on adversarial training to align multilingual embeddings in a common space and compare it with two approaches from related work based on linear transformation matrices.

For these **baseline alignments**, embedding spaces are typically aligned using a linear transformation based on bilingual dictionaries. We follow the work from Smith et al. (2017) and align embedding spaces based on orthogonal transformation matrices (see Section 2.3.4). These matrices can either be constructed in an unsupervised way by using words that appear in the vocabularies from both languages, i.e., equal words that can be identified using string matching, or in a supervised way based on real-world dictionaries (Mikolov et al., 2013b; Joulin et al., 2018). For the latter method, we build dictionaries based on translations from wiktionary.[3] For both methods, we reduce the vocabularies to the most frequent 5k words per language and treat English as the pivot language, i.e., we align all languages pairwise to English.

[1] `https://fasttext.cc/docs/en/crawl-vectors.html` [last accessed March 5, 2022.]

[2] `https://github.com/google-research/bert/blob/master/multilingual.md` [last accessed March 5, 2022.]

[3] `https://github.com/open-dsl-dict/wiktionary-dict` [last accessed March 5, 2022.]

Adversarial Alignment. We propose to use gradient reversal training to align embeddings from different (sub)spaces in an unsupervised way. Note that neither dictionaries nor other language resources are needed for this approach, making it applicable to zero- or low-resource scenarios. In particular, we extend the extraction model C with a discriminator D. Both model parts are trained alternately in a multi-task fashion. The *feature extractor* R is shared among them and consists of the embedding layer E, followed by a non-linear mapping:

$$R(x) = \tanh(W^\top E(x)) \tag{8.1}$$

with x being the current word, $W \in \mathbb{R}^{S \times S}$ and S being the embedding dimensionality.

The *discriminator* D is a multinomial non-linear classifier consisting of one hidden layer with ReLU activation (Hahnloser et al., 2000):

$$D(x) = \text{softmax}(T^\top \text{ReLU}(V^\top R(x))) \tag{8.2}$$

with $V \in \mathbb{R}^{S \times H}$, $T \in \mathbb{R}^{H \times O}$, H being a hyperparameter and O the number of different languages. The adversarial training using R, C, and D is performed similarly to the adversarial training for meta-embeddings in Chapter 5. With this, the discriminator is optimized for predicting the correct origin language of a given sentence, but at the same time, the feature extractor gets updated with gradient reversal, such that the language detection becomes more challenging and the discriminator cannot easily distinguish the word representations from different languages.

8.3.2 Models for Normalization

In the following section, we detail the normalization model. For this, masked language modeling and context-independent representations (CIRs, cf., Section 2.1.2) play a central role.

Masked Language Modeling. We model the task of assigning CIRs to temporal expressions as masked language modeling. In particular, we add TimeML annotations as inline information to the text sequences and mask the value field for prediction, e.g., "... <TIMEX3 type="DATE" value="**MASK**">yesterday</TIMEX3> ...". Note that those annotations could be the ground-truth annotations when applying the model on gold temporal expressions or predicted temporal expressions when using the model in the 3-step pipeline as described above. For this, we train a transformer model for CIR prediction using the masked language modeling objective. While this approach is general enough to be used to predict the normalized values directly, we saw in our ablation study (see Section 8.5.3) that state-of-the-art transformer models are not able to perform this task yet, even when given the anchor time as an additional input. Therefore, we replace the value field with the CIR instead (cf., Section 2.1.2).

Slot-Based Value Representation. Using only a single mask token for the whole CIR would require the model to store all possible CIRs in its vocabulary. Since it is not possible to enumerate, e.g., all possible dates, we model the CIRs as a fixed-length sequence of slots. We define 11 slots and use regular expressions to assign slot values in the training data. Figure 8.2 shows an example for the CIR `UNDEF-year-05` that is represented as the slots `[PAD], [year], [05], [PAD], ..., [PAD]`. Details on the slots and expressions are given below. To cover the full vocabulary of CIRs, we introduce 200 new tokens to the XML-R model. In our experiments, we compare this approach of using the pre-defined slots with subtokens from the XLM-R tokenizer for the CIRs.

We use the following 11 slots to represent CIRs values.[4] These slots are then used for masking during training and inference with our normalization model (which basically is a masked language model).

- **SB**: This slot can contain BC information of years (e.g., as in `BC4000` for the year 4000 BC) or the duration markers `P` and `PT`. Moreover, mathematical operations like `PLUS` are covered as used in relative expression involving offset computations (e.g., `this-day-plus-2` for the day after tomorrow) and holiday names (`EasterSunday`).

- **SD1, SD2**: These slots are used to represent 4-digit year numbers (**SD1** = `20` and **SD2** = `22` for the year 2022) to reduce the number of possible 4-digit numbers from 9999 to 99. This helps to generalize to unseen years as fewer parameters have to be learned. In addition, we use **SD1** to mark reference expressions like `PAST_REF`. For underspecifed expressions like `UNDEF-this-day`, the term `this` is stored in **SD1** and `day` in **SD2**. Moreover, **SD1** and **SD2** are used to store numbers of durations.

- **SD3, SD4**: Analogously to **SD1** and **SD2** that are used to store year information, **SD3** is used for months and **SD4** for days.

- **ST1, ST2, ST3**: Temporal information from expressions of type `TIME` that are smaller than day granularity are stored in the **ST** slots. For example, the hour information of `24:00` and the daytime information, such as `EV` is stored in **ST1**. Information on minutes and seconds is stored in **ST2** and **ST3**, respectively. Moreover, these slots are used to cover additional units in durations, such as in `P1D2H` (1 day and 2 hours).

- **SA1, SA2, SA3**: Finally, some CIRs include function calls which can be augmented with arguments that we store in the **SA** slots. For example, the argument `2` of `this-day-plus-2` is stored in **SA1**. Other function calls are used to compute days with respect to holidays like `EaserSunday` or specific weekdays.

Examples: The following examples show temporal expressions, their corresponding CIRs, and the tokenization into our slots. Note that there is no need to capture terms like

[4] Note that our CIRs describe a superset of TimeML

UNDEF in our slots as the presence of words like this, next or last in a CIR implies the existence of UNDEF in the CIR. This information can be reconstructed when obtaining a CIR from our slots. This also includes "-" to separate numbers as in YYYY-MM-DD values, REF in reference expressions and T for time information. We use the following format to give examples for our CIR conversion: *Text* → *CIR* → *Slot Sequence*

- *Now ...*
  → PRESENT_REF
  → **SD1**=PRESENT

- ... for *1000 days ...*
  → P1000D
  → **SB**=P, **SD1**=10, **SD2**=10, **SD4**=D

- ... for *one and a half day ...*
  → P1D12H
  → **SB**=P, **SD1**=1, **SD4**=D, **ST1**=12, **ST2**=H

- ... in *1000 BC ...*
  → BC1000
  → **SB**=BC, **SD1**=10, **SD2**=00

- ... on the *morning of March 15, 2022 ...*
  → 2022-03-15TMO
  → **SD1**=20, **SD2**=22, **SD3**=03, **SD4**=15, **ST1**=MO

- On *March 15*, ...
  → UNDEF-year-03-15
  → **SD1**=year, **SD3**=03, **SD4**=15

- ... *the day after tomorrow ...*
  → UNDEF-this-day-PLUS-2
  → **SB**=PLUS, **SD1**=this, **SD2**=day, **SA1**=2

- ... at *Pentecost*[5] ...
  → UNDEF-year-00-00 funcDateCalc(EasterSunday(YEAR, 49))
  → **SB**=EasterSunday, **SD1**=year, **SD2**=00, **SA1**=49

Note that slots can be optional depending on the temporal expression. For example, the value 2022 representing the year 2022 would only require **SD1** and **SD2**. All other slots are set to a padding value [PAD] then, which allows a fixed-sized representation of CIRs that can be predicted with our masked language model.

[5] In Christian communities, the holiday of Pentecost is celebrated 49 days after Easer Sunday.

Curriculum Learning. Our slot-based representation with 11 slots per CIR results in 11 masks. To train the model on this task, we apply curriculum learning in the first half of the training. In particular, we mask only a single slot of the CIR and steadily increase the number of masks up to the maximum of 11. For the second half of the training, the masking is applied to all slots. Following the masking strategy of recent MLMs, we do not only mask the value, but also mask different parts of the sentence to allow our model to learn dependencies between the value and its context and vice versa. More specifically, we mask the value slots for 70% of sentences, annotated tokens for 15%, types with 10%, and other text parts with 5%. Note that we also experimented without curriculum learning by masking all 11 slots directly. The results are given in Section 8.5.3.

Inference and Decoding. For inference, we first add 11 masks (i.e., one per slot) as value placeholders that need to be predicted to the input sentence. Then, we use the masked language model to predict the most probable sequence of slots for the CIR. To get the final value, we apply sequential left-to-right decoding of all masks by iteratively decoding the left-most mask and replacing the mask with its predicted value until all masks are resolved. We compare two alternative decoding strategies: (i) Decoding all masks simultaneously. (ii) Training a conditional random field model that takes the logits as input and uses the Viterbi algorithm to determine the most probable sequence of predictions. See Section 2.2.1 for more details on CRFs and their decoding.

8.4 Experimental Setup

This section describes our experimental setup. We will provide information on the gold-standard dataests used for our multilingual temporal tagging experiments, as well as the creation of our weakly-supervised training data. Finally, we provide details on our models.

8.4.1 Datasets and Metrics

Our experiments will be performed on corpora in 17 language annotated either following the TIMEX3 or TIMEX2 standard. More details on the corpora and their sources are given in the next paragraphs. We divide the languages into high- and low-resource languages depending on whether manually created HeidelTime rules are available for the respective language.

For evaluation, we use the TEMPEVAL-3 evaluation script (UzZaman et al., 2013) and report strict and relaxed extraction F_1-score for complete and partial overlap to gold standard annotations, respectively. We also report the type F_1-score for the classification into the four temporal types: DATE, TIME, DURATION, and SET and the value F_1 if applicable.

Corpus	Language	#Annotations (train / test)	Reference
Corpora only used for evaluation			
KRAUTS-DIEZEIT	German (*de*)	_ / 493	(Strötgen et al., 2018)
TEMPEVAL-3 (platinum)	English (*en*)	_ / 137	(UzZaman et al., 2013)
KOMPAS (test)	Indonesian (*id*)	_ / 192	(Mirza, 2015)
TIMEBANKCA	Catalan (*ca*)	_ / 1383	(Saurı, 2010)
ESTTIMEML	Estonian (*et*)	_ / 622	(Orasmaa, 2014)
EUSTIMEML	Basque (*eu*)	_ / 112	(Altuna et al., 2020)
FR TIMEBANK	French (*fr*)	_ / 423	(Bittar et al., 2011)
RO TIMEBANK	Romanian (*ro*)	_ / 151	(Forascu and Tufis, 2012)
PT-TIMEBANK (test)	Portuguese (*pt*)	_ / 151	(Costa and Branco, 2012)
WIKIWARS-EL (test)	Greek (*el*)	_ / 414	(Kapernaros, 2020)
Corpora split into train and test sets			
MEANTIME (*it*)	Italian (*it*)	229 / 244	(Minard et al., 2016)
MEANTIME (*nl*)	Dutch (*nl*)	221 / 259	(Minard et al., 2016)
TEMPEVAL-3 (*es*)	Spanish (*es*)	730 / 551	(UzZaman et al., 2013)
POLEVAL-2019	Polish (*pl*)	633 / 6011	(Kocon et al., 2019)
WIKIWARS	English (*en*)	1378 / 1251	(Mazur and Dale, 2010)
WIKIWARS-DE	German (*de*)	1510 / 684	(Strötgen and Gertz, 2011)
WIKIWARS-HR	Croatian (*hr*)	724 / 677	(Skukan et al., 2014)
WIKIWARS-UA	Ukrainian (*ua*)	454 / 2237	(Grabar and Hamon, 2019)
WIKIWARS-VI	Vietnamese (*vi*)	118 / 101	(Strötgen et al., 2014a)
Corpora only used for training			
KRAUTS-DOLOMITEN	German (*de*)	388 / _	(Strötgen et al., 2018)
MEANTIME (*en*)	English (*en*)	472 / _	(Minard et al., 2016)
TEMPEVAL-3 (train, *en*)	English (*en*)	1240 / _	(UzZaman et al., 2013)
PT-TIMEBANK (train)	Portuguese (*pt*)	1127 / _	(Costa and Branco, 2012)
WIKIWARS-EL (train)	Greek (*el*)	1496 / _	(Kapernaros, 2020)

Table 8.1: Overview of gold-standard datasets for temporal tagging.

Gold-Standard Training and Evaluation Data. Our models are evaluated on gold-standard corpora in up to 17 languages. Corpus statistics and details on the training and test splits are shown in Table 8.1.

Weakly-Supervised Training Data for Normalization. For training the normalization model, we create a large-scale weakly-supervised dataset covering 87 languages.[6] Reasons are that (i) existing gold training data is too small to cover the wide range of different values and (ii) CIRs are not part of existing annotations. For all languages, we take the data from GlobalVoices[7] (news-style documents) and Wikipedia[8] (narrative-style documents), use Spacy for tokenization and HeidelTime for the annotation with temporal expressions.

[6]The set of 87 languages is the intersection of languages covered by HeidelTime, our data, and the XLM-R language model that we use for our normalization models.

[7]`https://globalvoices.org/` [last accessed March 5, 2022.]

[8]`https://en.wikipedia.org/wiki/List_of_Wikipedias` [last accessed March 5, 2022.]

Rank	Lang	#Ann.	Rank	Lang	#Ann.	Rank	Lang	#Ann.	Rank	Lang	#Ann.
1	*de*	870897	23	*sv*	13705	45	*ja*	3696	67	*am*	776
2	*en*	542087	24	*id*	13031	46	*br*	3582	68	*ku*	557
3	*fr*	284871	25	*da*	12919	47	*uz*	3361	69	*so*	506
4	*ar*	280446	26	*fy*	12852	48	*th*	3162	70	*yi*	485
5	*es*	250871	27	*pl*	11283	49	*cs*	3096	71	*ko*	483
6	*pt*	215209	28	*fa*	11041	50	*ga*	2799	72	*si*	442
7	*it*	199236	29	*eu*	10992	51	*mn*	2778	73	*ps*	403
8	*nl*	194944	30	*ne*	10750	52	*gd*	2772	74	*lo*	354
9	*ru*	122884	31	*ms*	10017	53	*lt*	2734	75	*km*	350
10	*zh*	105421	32	*mg*	9271	54	*mr*	2623	76	*su*	335
11	*hr*	50233	33	*kk*	8080	55	*la*	1876	77	*lv*	323
12	*ro*	33545	34	*hi*	7762	56	*ua*	1673	78	*as*	299
13	*vi*	22048	35	*eo*	7353	57	*hy*	1642	79	*ug*	283
14	*af*	21081	36	*ur*	6228	58	*ta*	1556	80	*sd*	278
15	*mk*	19539	37	*hu*	5871	59	*my*	1103	81	*gu*	258
16	*tr*	19532	38	*sq*	5760	60	*ml*	1079	82	*ha*	205
17	*gl*	17416	39	*sk*	5172	61	*kn*	1029	83	*sl*	125
18	*ca*	16747	40	*sr*	4276	62	*fi*	1017	84	*yo*	102
19	*bn*	16284	41	*ka*	4247	63	*oa*	979	85	*sa*	24
20	*cy*	14738	42	*el*	4217	64	*jv*	968	86	*or*	19
21	*bg*	14550	43	*he*	4057	65	*ky*	926	87	*xh*	3
22	*et*	13948	44	*sw*	3979	66	*is*	804			

Table 8.2: Overview of our weakly-supervised training data with the number of annotations per language. This data was annotated with our adapted HeidelTime model and the boundaries, types and ICRs of temporal expressions are labeled.

A Note on Adopting HeidelTime. In our experiments, we use a modified version of HeidelTime. First, we implement a new UIMA collection reader based on Spacy as an alternative to the TreeTagger that has a restrictive license. This results in a slightly different sentence segmentation and tokenization and, thus, minor differences in performance. For example, the original HeidelTime achieves 63.47 F_1 on the Portuguese test data, while our Spacy version achieves 63.24 F_1 as one additional false-positive expression was annotated due to different sentence boundaries. Second, we changed the HeidelTime output to display the internal CIRs for the TimeML values, such that we can create our weakly-supervised training data. The resulting number and quality of annotations are highly dependent on the amount of available data for that language and the quality of HeidelTime's rules. Details on the weakly-supervised data are given in Table 8.2. We filtered sentences without annotations.

Regular Expressions for CIR Extraction. We will now describe the six regular expressions used to split CIR values from HeidelTime outputs into our slots to create the weakly-supervised training data. For readability, we define the following groups to cap-

ture temporal units and other fixed names. Note that these are used across languages. For example, the German expression *Montag* would still be represented with `monday`.

```
UNITS = (H|D|DE|DT|M|C|Y|C|CE|W|WE|Qu|Q|S)
UNITS_F = (day|month|year|decade|century|week|
    weekend|quarter|hour|minute|second)
DAYTIME = (NI|AF|MO|EV|MD|MI)
SPECIAL = (SP|SU|FA|AU|WI| H1|H2|Q1|Q2|Q3|Q4|H|Q)
NAMES = (monday|tuesday|wednesday|
    thursday|friday|saturday|sunday|
    january|february|march|april|may|june|july|
    august|september|october|november|december)
```

$D1$**: References.** The first regular expression $D1$ is used to capture simple reference expressions that refer to uncertain points in time. $DX(n)$ marks the n-th group captured by the regular expression DX.

```
D1 = (PRESENT|PAST|FUTURE)_REF
Slots: SD1=D1(1)
```

$D2$**: Explicit Dates.** The regular expression $D2$ detects explicit values that do not need further normalization, such as days in the `YYYY-DD-MM` format, e.g., `2022-03-15`.

```
D2 =(BC)?(\d\d?|XX)?(\d\d|XX)? (?:-(W)?(\d\d?|XX|SPECIAL))?
    (?:-(\d\d?|XX|WE))?\)? (?:T(\d\d|X|DAYTIME|XX)?
    (?::(\d\d))?(?:(?::|-)(\d\d))?)?

Slots: SB=D2(1), SD1=D2(2), SD2=D2(3), SD3=D2(5),
    SD4=D2(6), ST1=D2(7), ST2=D2(8), ST3=D2(9)|D2(4)
```

$P1$**: Durations.** The third regular expression $P1$ detects expressions of `type` DURATION, e.g., `P1D2H`. These are defined as `P<number><unit>` for units of at least day granularity and `PT<number><unit>` for smaller granularities. We capture up to two different units `P1D2H` (1 day and 2 hours) but ignore further units that are theoretically defined in the TimeML specifications but do not often occur in practice (those did not occur at all in our datasets).

```
P1 = (P|PT)(\d\d?|X|XX)(\d\d|\.)?(\d\d?)?)?(UNITS)?
    (\d\d?)?(UNITS)?

Slots: SB=P1(1), SD1=P1(2), SD2=P1(3), SD3=P1(4), SD4=P1(6),
    ST1=P1(5), ST2=P1(7)
```

$D3$**: Relative Dates.** While the previous regular expressions $D1$, $D2$, and $P1$ follow the TimeML specifications and capture fully normalized expressions, i.e., anchored values, the following regular expressions capture CIRs as used internally by HeidelTime. They represent relative expressions that need to be anchored.

$D3$ detects relative expressions with respect to a certain point in time, such as `this-day-plus-2` (the day after tomorrow).

```
D3 = UNDEF-(this|next|last|REF| REFUNIT|REFDATE)?-?
     (UNITS_F|SPECIAL)?-??(NAMES|SPECIAL)|XX|\d\d?)?
     (?:-?(\d\d?|XX))?(?:-(PLUS|MINUS|LESS)-(\d\d?)-?
     (\d\d?)?-?(\d\d?)?)?\)?(?:T(\d\d?|X|DAYTIME|XX)?
     (?::(\d\d?|XX))?(?:(?::|-) (\d\d|XX))?)

Slots: SB=D3(5), SD1=D3(1),  SD2=D3(2), SD3=D3(3),
     SD4=D3(4), ST1=D3(9), ST2=D3(10), ST3=D3(11),
     SA1=D3(6), SA2=D3(7), SA3=D3(8)
```

$D4$**: Relative Dates (coarse).** $D4$ captures underspecified expressions like *May* that is missing year information and would be represented with the CIR `UNDEF-year-05`.

```
D4 = UNDEF-(year|decade|century?)-?(\d\d?|X)?-?(\d\d?|X)?-?
     (\d\d?|X|SPECIAL)?\)?(?:T(\d\d?|X|DAYTIME)?
     (?::(\d\d?|XX))?(?:(?::|-)(\d\d|XX))?)?

Slots: SD1=D4(1), SD2=D4(2), SD3=D4(3), SD4=D4(4),
     ST1=D4(5), ST2=D4(6), ST3=D4(7)
```

$D5$**: Holidays and functions.** Finally, $D5$ covers special functions used by HeidelTime. These functions are used to compute days with respect to weekdays and moveable feast like `EasterSunday` that refer to different days depending on the year. For example, the earliest possible date of Easter Sunday is March 22, and the latest is April 25 in the Gregorian calendar.[9] The concrete date is then computed by an external function given a year.[10]

```
D5 = (UNDEF-year|UNDEF-this-year|UNDEF-century\d\d|\d\d\d\d)
     -(\d\d)-00 funcDateCalc\((
         EasterSunday|EasterSundayOrthodox|
         WeekdayRelativeTo| ShroveTideOrthodox)
     \(YEAR(?:(?:-(\d\d)))?(?:-(\d\d))(?:,\s?(-?\d\d?))?
     (?:,\s?(-?\d\d?))?(?:, (true|false))?\)\)

Slots: SB=D5(3), SD1=D4(1), SD2=D4(2), SD3=D4(5),
     SD4=D4(7), ST1=D4(3), SA1=D5(6), SA2=D5(7), SA3=D5(8)
```

[9] `https://en.wikipedia.org/wiki/List_of_dates_for_Easter` [last accessed March 5, 2022.]

[10] `https://www.linuxtopia.org/online_books/programming_books/python_programming/python_ch38.html` [last accessed March 5, 2022.]

8.4.2 Model Settings

This section will describe the experimental setups for the extraction and normalization of temporal expressions with our neural networks. We also give further details on the alignment methods for the multilingual extraction models and the anchoring of context-independent representations.

Extraction Models. We model the extraction as a sequence-labeling problem as described in Section 8.3.1. For this, we convert the annotated corpora from TimeML format into the tokenized BIO format using Spacy tokenizers.[11]

Our first model is based on the multilingual XLM-R transformer (Conneau et al., 2020) with a linear layer for classification. For the monolingual extraction (Mono), we train one model per language on the language-specific gold-standard resources if available or the weakly-supervised data otherwise. For the multilingual setting (Multi), we train a single model on the combined training resources of all 17 languages as given in Table 8.1. The models are trained for a fixed number of three epochs.

Moreover, we explore the adversarial training for embedding alignment. For this, we use BiLSTM-CRF models based on BERT or fastText embeddings. These multilingual models are trained using the Portuguese TimeBank (Costa and Branco, 2012) and TEMPEVAL-3 (UzZaman et al., 2013) for Spanish and English (TimeBank subset). To demonstrate that our model is able to generalize to unseen languages, we perform tests using the French (Bittar et al., 2011), Catalan (Saurı, 2010) and Basque TimeBanks (Altuna et al., 2016) and the DIEZEIT subset of the German KRAUTS corpus (Strötgen et al., 2018).

We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $1e-5$ for the BiLSTM-CRF model part and $1e-6$ for BERT. These modes are trained for a maximum of 50 epochs using early stopping on the development set. The BiLSTM has a hidden size of 128 units per direction. The labels are encoded in the BIO format. For regularization, we apply dropout with a rate of 10% after the input embeddings. The discriminator for adversarial training has a hidden size H of 100 units and is trained after every 10th batch of the sequence tagger with λ set to 0.001.

Normalization Model In the following section, we will give details on our normalization model based on the multilingual XLM-R transformer (Conneau et al., 2020). For the **normalization to CIRs**, we train our proposed model with masked language modeling on the weakly supervised data (see Section 8.3). In our experiments, we evaluate this model in combination with the multilingual extraction model (Multi+OUR) as well as in combination with the gold boundaries for temporal expressions (Gold+OUR), which serves as an upper bound.

[11] `https://spacy.io/` [last accessed March 5, 2022.]

Task	Metric	HT	fastText-LSTM unaligned	fastText-LSTM aligned w/o Dict.	fastText-LSTM aligned w/ Dict	fastText-LSTM aligned w/ AT	BERT-LSTM unaligned	BERT-LSTM aligned w/ AT	XLM-R Mono (gold)	XLM-R Multi (17 lang.)
en	strict	81.78	68.36	69.10	70.80	75.63 †	73.09	74.80 †	**85.7**	82.0
	relaxed	90.71	79.14	79.03	81.21	82.03 †	84.34	86.61 †	**92.3**	88.9
	type	83.27	72.13	72.18	73.32	72.85 †	75.50	79.53 †	**86.5**	82.8
es	strict	85.87	75.67	76.53	77.44	79.64 †	79.11	79.55	**89.6**	89.3
	relaxed	90.13	82.43	82.45	82.47	84.46 †	84.12	85.71	**94.5**	94.2
	type	87.47	78.07	78.46	78.24	80.88 †	80.22	80.11	**91.4**	90.0
pt	strict	71.59	70.36	70.20	70.48	72.41	74.52	75.47	**79.0**	76.6
	relaxed	81.18	76.77	75.86	76.29	78.15	80.75	81.51	**82.4**	81.3
	type	76.75	72.29	71.50	72.26	73.84	75.47	76.23	**79.0**	78.1

Table 8.3: Results for multilingual models trained on English, Spanish and Portuguese data jointly. † highlights aligned models with statistical significant differences to the unaligned model (paired permutation test, p=0.05). HT stands for HeidelTime. Monorefers to the XLM-R finetuned on gold-standard monolingual data and Multifor the variant trained on 17 jointly.

For the **anchoring of CIRs**, we use rules similar to the HeidelTime rules.[12] In particular, anchor dates can be given by the document creation time or by previous temporal expressions (Strötgen and Gertz, 2016).

8.5 Results and Analysis

This section will provide the results of our extraction and normalization experiments. We compare our model to HeidelTime (Strötgen and Gertz, 2013).

8.5.1 Results for Extraction

The results for our multilingual extraction experiments with embedding alignments are provided in Table 8.3. We train three models with different random seeds and report the performance of the model with median performance on the combined development set of all three languages for the BERT and fastText-based models. Moreover, we compare to the XLM-R model from our normalization pipeline that was trained in a pure monolingual fashion (Mono) or on a larger set of multilingual resources from 17 languages (Multi).

The effects of aligning fastText embeddings are clearly visible in Table 8.3. The supervised alignment using a dictionary is always superior compared to the unsupervised

[12]More precisely, we use a slightly modified version of HeidelTime's SPECIFYAMBIGUOUSVALUESSTRING function, which incorporates tense information of the context play a critical role. We compute that using morphological features from Spacy (`https://spacy.io/usage/linguistic-features#morphology` [last accessed March 5, 2022.])

Task	Metric	HeidelTime -Auto	BERT-LSTM unaligned	BERT-LSTM aligned w/ AT	XLM-R Mono (weak)	XLM-R Multi (17 languages)
	strict	52.35	60.12	62.58	**82.5**	82.4
fr	relaxed	72.02	74.23	75.46	88.1	**89.8**
	type	68.70	61.96	62.07	**79.7**	76.9
	strict	38.87	63.34	66.53	**75.4**	70.9
de	relaxed	52.11	76.51	77.82	**85.9**	82.6
	type	50.15	66.95	69.04	**80.6**	76.2
	strict	28.11	63.24	64.21	29.5	**77.3**
ca	relaxed	62.81	74.95	77.00	64.3	**87.8**
	type	60.84	65.66	67.85	62.3	**82.5**
	strict	22.54	43.96	47.87	0.0	**59.7**
eu	relaxed	26.76	61.54	63.83	0.0	**70.2**
	type	23.94	57.14	58.51	0.0	**66.0**

Table 8.4: Results for the unsupervised cross-lingual extraction. We compare to HeidelTime with automatically generated resources, which resembles a similar setting. Here, monolingual refers to the XLM-R finetuned on weakly-supervised monolingual data.

alignment without a dictionary or the unaligned embeddings. Our proposed adversarial alignment (w/ AT) leads to the best results across languages. The performance of BERT is close to the best fastText model. Aligning BERT with adversarial training also increases performance. The improvements are smaller compared to fastText but still statistically significant for English.

Table 8.4 provides transfer results of the models with BERT embeddings to languages without labeled training data.[13] It outperforms the state-of-the-art HeidelTime models by a large margin. The impressive performance of the multilingual BERT in the cross-lingual setting can be explained by the fact that the model has seen many sentences in our target languages during the pre-training phase, which can now be effectively leveraged in this new setting.

Our XLM-R model from the pipeline experiments outperforms the aligned LSTM-based extraction models for the high-resource experiments. We assume that this model achieves better results because the XLM-R model is often considered superior compared to mBERT in multilingual settings (Conneau et al., 2020). Moreove, the fine-tuning training of XLM-R might be better than the feature-based BiLSTM approach we use for BERT, and the XLM-R model was trained on a much larger set of languages in our experiments. Note that the monolingual XLM-R model outperforms the current state of the art for English (Lee et al., 2014) who achieve 83.1/91.4/85.4 for strict/relaxed/type F_1.

[13] The results of the fastText models were considerably lower for cross-lingual transfer.

	HeidelTime				Mono+OUR				Multi+OUR				Gold+OUR
	Str.	Rel.	Type	Val.	Str.	Rel.	Type	Val.	Str.	Rel.	Type	Val.	Val.
avg.	54.4	65.6	60.9	52.5	61.7	73.8	70.9	55.5	**75.0**	**85.8**	**80.4**	**64.0**	73.9
High-resource languages													
de (N)	69.7	79.3	75.4	**62.4**	**75.4**	**85.9**	**80.6**	61.5	70.9	82.6	76.2	59.5	73.2
de (W)	88.5	94.3	89.0	84.8	**89.6**	**97.0**	**96.0**	83.8	88.9	96.7	95.4	**85.7**	87.3
en (N)	81.8	90.7	83.3	**78.1**	**85.7**	**92.3**	**86.5**	72.5	82.0	88.9	82.8	70.5	78.3
en (W)	90.6	94.3	90.6	**94.3**	93.1	96.6	**93.1**	89.7	**94.7**	**98.3**	87.7	94.2	94.2
es (N)	83.7	90.2	86.1	**80.9**	**89.6**	**94.5**	**91.4**	79.0	89.3	94.2	90.0	77.1	84.4
et (N)	42.4	57.4	51.3	44.0	3.3	28.0	24.4	9.6	**55.5**	**78.0**	**72.0**	**45.2**	64.8
fr (N)	**85.6**	**90.6**	**82.3**	**73.3**	82.5	88.1	79.7	67.9	82.4	89.8	76.9	61.4	68.0
hr (W)	**93.3**	**95.8**	**94.6**	**85.7**	84.1	90.8	89.5	74.6	86.3	91.7	90.1	75.7	84.7
it (N)	**84.4**	**92.9**	**83.5**	**74.1**	69.8	81.4	73.7	60.4	76.8	82.4	78.4	67.2	75.3
nl (N)	54.0	**91.3**	79.0	44.4	61.4	73.0	67.2	42.7	**76.0**	82.7	**81.4**	**53.5**	64.6
pt (N)	71.3	80.9	76.5	63.2	**87.1**	**91.2**	85.0	**68.7**	**87.1**	91.1	**86.5**	**68.7**	76.6
vi (W)	**92.6**	89.5	**96.6**	**91.6**	87.6	85	89.8	83.5	91.5	**93.8**	92.6	90.8	91.2
avg.	78.2	87.3	82.4	**73.1**	75.8	83.6	79.7	66.2	**81.8**	**89.2**	**84.2**	70.8	78.6
Low-resource languages													
ca (N)	28.1	62.8	61.1	43.6	29.5	64.3	62.3	40.2	**77.3**	**87.8**	**82.5**	**59.7**	68.1
el (W)	2.2	4.9	4.9	1.3	47.0	88.2	86.1	64.6	**81.7**	**92.0**	**90.2**	**70.6**	83.7
eu (N)	22.5	26.8	23.9	18.3	0.0	0.0	0.0	0.0	**59.7**	**70.2**	**66.0**	**45.0**	50.4
id (N)	19.7	54.7	44.5	40.1	17.4	39.7	30.6	25.6	**49.7**	**79.5**	**63.9**	**46.9**	64.8
pl (N)	18.8	27.2	16.5	11.2	86.1	**92.5**	87.6	58.7	**86.7**	92.2	**87.7**	**59.0**	66.0
ro (N)	3.2	19.5	16.7	5.5	3.8	22.6	37.0	7.7	**9.8**	**47.2**	**39.1**	**19.7**	54.6
ua (W)	1.6	2.8	2.2	1.2	**80.2**	90.6	87.5	63.6	79.4	**90.7**	**88.8**	**65.4**	74.5
avg.	13.7	28.4	24.3	17.3	37.7	56.8	55.9	37.2	**63.5**	**79.9**	**74.0**	**52.3**	66.0

Table 8.5: Overview of the extraction (Strict, Relaxed, Type F_1) and normalization results (value F_1) for our models and HeidelTime. The gold extractions (Gold+OUR) simulate a pipeline with perfect extractions. (N) and (W) refer to news articles and Wikipedia, respectively.

As the XLM-R models performed best, we focus on this model type and evaluate it on more languages, including more low-resource languages. The extended extraction results are visualized in Figure 8.4 and detailed information is provided in Table 8.5. In general, multilingual extraction outperforms monolingual extraction, probably because the model is able to use knowledge from different languages. Our multilingual model achieves +2 F_1 for high-resource and +51 F_1 for low-resource languages compared to HeidelTime. Note that HeidelTime with automatically created rules has a poor performance for some low-resource languages (*el*, *ro*, *ua*). This is similar to the observations of Grabar and Hamon (2019) who found that "[e]xploitation of this automatically built system produced no results when applied to the Ukrainian data" (Grabar and Hamon, 2019, p. 3). For these languages, the automatically generated are not good enough in practice to perform temporal tagging, which underlines the need for multilingual systems like our pipeline model.

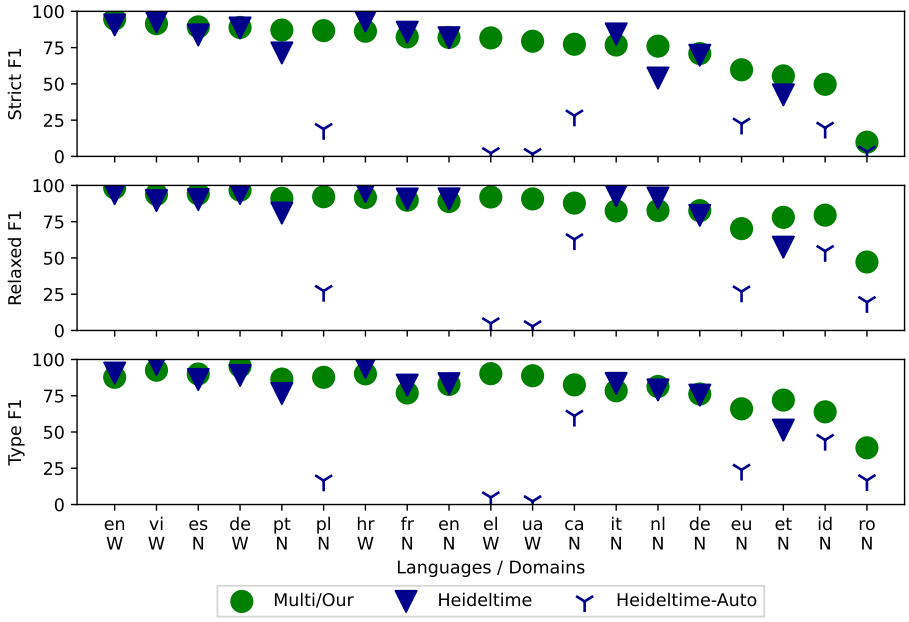


Figure 8.4: Extraction results for temporal expressions in 17 languages.

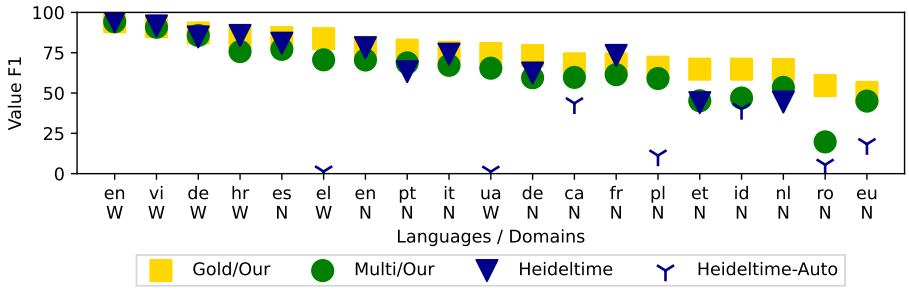


Figure 8.5: Normalization results for temporal expressions in 17 languages.

8.5.2 Results for Normalization

The normalization results are given in Figure 8.5 and Table 8.5. Our multilingual masked language model matches HeidelTime's performance rather close for high-resource languages and outperforms it for low-resource languages with an increase of 35 F_1 points on average. Our method can generalize well across languages but loses some accuracy in monolingual settings, as we use a single multilingual model that was optimized for many languages at once.[14]

[14]The rather low performance of our models and HeidelTime for the high-resource languages Estonian (*et*) and Dutch (*nl*) can be explained by the poor data quality of these corpora. An inter-annotator agreement of 44 F_1 was reported for the Estonian corpus (Orasmaa, 2014), which is close to our results. The Dutch data was translated from English and automatically annotated via cross-lingual projections (Minard et al., 2016), which may reduce the annotation quality. Note that only the first five sentences for each document were annotated for the MEANTIME corpora (*it*and *nl*). We restricted our evaluation to these annotated parts accordingly.

	News		Wiki		Low-Resource	
	de	*en*	*de*	*en*	*ca*	*eu*
OUR	60.3	**72.0**	**85.7**	91.2	**59.7**	**47.1**
Decoding Strategy (OUR uses Sequential)						
w/ Simultaneous	59.5	70.5	85.5	91.2	**59.7**	45.0
w/ Viterbi	**61.5**	71.3	85.4	91.2	**59.7**	44.0
Value Representation						
w/o OUR Slots	58.9	70.0	81.0	87.7	57.4	27.3
w/o OUR CIR	56.5	59.0	64.2	45.6	34.2	21.5
Training Strategy						
w/o Curriculum	59.3	70.5	83.2	**91.3**	57.3	30.5
Training Data (OUR uses Weak)						
Weak + Gold	55.6	65.8	84.5	90.8	-	-
only Gold	11.0	13.4	11.5	15.6	-	-
only Monolingual	57.8	67.4	**85.6**	87,7	28.4	7.3

Table 8.6: Ablation study for our model components (value F_1). All rows ablate for a single effect compared to OUR.

Finally, we can give an upper bound for our normalization method by predicting values on the gold extractions. Using these extractions in our pipeline model increases the average performance from 64.0 F_1 up to 73.9 F_1 by almost ten points. This shows that the extraction part still offers room for future improvements that our normalization model is able to use.

8.5.3 Ablation Studies of Normalization Model

As our proposed normalization model consists of multiple components, we now investigate their individual effects in more detail. The results are given in Table 8.6.

Decoding Strategies. First, we test different decoding strategies as described in Section 8.3. We find that sequential decoding works best. However, it also requires more computation time. A cheaper alternative with only minor performance decreases is the simultaneous decoding of all masks.

Value Representations. Second, we analyze the impact of different value representations by comparing our proposed approach with CIR and slot tokenization to (i) tokenization of values using the standard XLM-R tokenizer instead of pre-defined slots (w/o OUR Slots), and (ii) training a model to directly predict the anchored value without CIRs in between (w/o OUR CIR). For (i), we find that our slot method has major advantages when processing narrative texts, such as Wikipedia, due to the higher amount of relative expressions (cf., Table 8.7), that are tokenized into many subtokens (up to 34, instead of 11 when using our slots). For (ii), we add the document creation time to the input of the model so that the

	de	*en*
News	**67.1** / 32.9	**52.3** / 47.7
Wiki	47.6 / **52.4**	44.2 / **55.8**

Table 8.7: Distribution of explicit / relative values according to HeidelTime (in %).

model has all necessary information to predict the fully normalized value directly instead of a CIR. However, we find that current transformers are not able to correctly incorporate this information and mostly predict a memorized, incorrect value. Thus, using CIRs as an intermediate step is important for neural temporal tagging.

Training Strategy and Data. Finally, we investigate the *training strategy* and *training data*. Our curriculum learning has advantages for low-resource languages as it reduces the training complexity, which helps for the difficult adaptation to languages with few resources. Weakly-supervised training data is required, as the amount of gold-standard data is too small to train the MLM model. Finetuning the trained MLM model further on gold data (Weak+Gold) decreases performance slightly. Training the model only on monolingual data also decreases performance, highlighting the prospects of our multilingual approach.

8.6 Conclusions

In this chapter, we explored multilingual temporal tagging. In particular, we addressed both of its subtasks: the extraction and normalization of temporal expressions with neural networks. For this, we proposed multilingual extraction models that leverage training resources from many languages and that set the new state of the art for multilingual extraction of temporal expressions. In addition, we explored the alignment of languages inside multilingual transformers and word embeddings and demonstrated that these methods are a step forward towards truly multilingual models.

Moreover, we have introduced a new method for normalizing temporal expressions by training transformers models with on masked language modeling to predict context-independent representations in our a new slot-based prediction scheme. We were able to train a single multilingual model for the task with this approach — the first neural normalization method for temporal expressions. We evaluated our method in 17 languages and set the new state of the art in low-resource languages with massive improvements of 35 F_1 points on average. The success of our method demonstrates the potential of neural networks for temporal normalization, and we are convinced that it will enable future research on this topic. A promising research direction is the joint modeling of extraction, normalization, and resolution.

Chapter 9

Summary and Outlook

The field of information extraction was significantly reshaped since the broad adoption of deep-learning methods for natural language processing. Major advantages of these neural approaches are new task-agnostic representation models that can be used to solve many tasks with minimal architectural changes. However, these models require large-scale pre-training, which opened up many exciting research directions concerning the applicability of neural networks in low-resource settings, including non-standard domains and languages.

9.1 Summary and Conclusions

In this thesis, we targeted various information extraction tasks in low-resource domains and languages and explored different representation learning methods to enable deep learning in these challenging settings, including robust model architectures, advanced training methods, and the potential of transfer learning. Our proposed methods allow us to create or leverage robust input representations for information extraction tasks in non-standard domains or languages. We will summarize our main contributions in the following:

(1) Domain-specific knowledge contained in word representation methods has a positive effect on information extraction models in non-standard domains. We demonstrated that deep learning models can be greatly improved in these domains by incorporating domain-specific knowledge from the target domain, either via fine-tuning on documents from the target domain (Chapter 4), joint training of multiple tasks, or combining domain-specific and general-domain representations (Chapter 3).

(2) The combination of different word representations does not only help to infer domain knowledge but can also be used to capture other method-specific properties of these representations, such as the strength of word-level embeddings and the flexibility of subword-based embeddings. We made important contributions by studying meta-embedding meth-

ods that can dynamically incorporate the advantages of many embeddings and by proposing a novel architecture to improve performance in low-resource settings (Chapter 5). We further showed the applicability of meta-embeddings to combine embeddings from different languages (Chapter 6).

(3) The knowledge contained in pre-trained languages models, in particular specific domain knowledge, helps to address problems across tasks in non-standard domains. Its transfer enables the creation of task-specific models when the training data is limited assuming that transfer sources are carefully selected. We propose a new similarity measure that outperforms the existing similarity measures, in particular for ranking cross-task transfer sources. Moreover, we showed that the source selection process should be dynamic with respect to the number of transfer sources — an aspect not considered in prior approaches. We proposed dynamic selection models that are able to predict sets of helpful transfer sources and avoid negative transfer (Chapter 7).

(4) Pre-trained language models can also be transferred across languages by training on many languages jointly and applying the models to unseen languages. We exemplarily showed this for temporal tagging by training multilingual extraction and normalization models. For this, we proposed the first neural method for temporal expression normalization by using a masked language modeling training objective and context-independent representations and explored the adversarial alignment of multilingual models (Chapter 8).

(5) We performed broad evaluations across a total of 30 languages and 16 domains, for instance in the clinical domain, to demonstrate the robustness of our methods for various sequence tagging and classification tasks.[1] Our models set the new state of the art for many of these datasets. We have won two international shared tasks on Spanish clinical NLP, highlighting the language- and domain-agnostic applicability of our models, as we participated as neither language nor domain experts and outperformed various approaches by teams with domain or language experts.

9.2 Outlook and Discussion

Despite various contributions detailed in this thesis and promising advances in the field in general, information extraction, in particular in low-resource domains and languages, is far from being a solved problem. For example, Ruder (2019a) named low-resource NLP one of the four biggest open problems in research. In this section, we will discuss current challenges and outline future opportunities that we see in the field of low-resource information extraction.

[1] Languages: *bg*, *ca*, *cs*, *da*, *de*, *el*, *en*, *es*, *et*, *eu*, *fa*, *fi*, *fr*, *ga*, *he*, *hi*, *hr*, *hu*, *id*, *it*, *nl*, *no*, *pl*, *pt*, *ro*, *sl*, *sv*, *ta*, *ua*, *vi* (30 in total). Domains: academic publications, biographies, clinical, conversations, cybersecurity, emails, financial, how-to guides, literature & fiction, news, sms, social media (including Twitter and Reddit), travel guides, webblogs, wetlab protocols, wikipedia (16 in total).

Adapting Language Models to New Domains and Languages Pre-trained language models are trained on a diverse set of domains and languages. However, as of today, the languages covered during the pre-training are not equally well represented in the resulting model, which reduces the possible performance for low-resource languages significantly (Pires et al., 2019). Solutions, such as our adversarial training for domain-robust meta-embedding or multilingual alignment, can help to utilize the existing resources in a single model across domains and languages without increasing the amount of training data or the number of model parameters. Nonetheless, these methods only improve the model for languages and domains known at the time of training. More dynamic solutions are required for integrating new tasks after the training, as the field of information extraction will likely keep expanding into new tasks, domains, and languages, and the training of new models from scratch is expensive. Promising research on this problem includes the extension of existing models for new languages (Pfeiffer et al., 2021) or domains, e.g., by using compositional approaches as done by Gururangan et al. (2021) to add new domains quickly without touching the majority of model parameters. We think that exploring this research direction will not only help to reduce the economic and ecological impacts (Strubell et al., 2019; Schwartz et al., 2020; Bender et al., 2021) by reusing existing resources, but will also help to tackle information extraction in new languages and domains, which could not have been addressed otherwise due to resource limitations.

Combinations of Methods. Another challenging future research direction concerns the combination of different methods. For example, we have shown in Chapter 5 that meta-embeddings on word level and domain-adapted transformers on subword level both greatly improve over the standard models by leveraging domain knowledge in two different ways. Nonetheless, their combination still does not further improve performance in practice, which highlights the need for future research in this direction. We found similar effects for model transfer over clinical datasets and clinical pre-training of general-domain representations in Chapter 7. However, this is not limited to the methods discussed in this thesis, and includes other methods like data augmentation (Wei and Zou, 2019; Dai and Adel, 2020; Feng et al., 2021) or distant supervision (Mintz et al., 2009; Adelani et al., 2020). One assumption is that these methods are complementary, as hypothesized by Longpre et al. (2020) for the case of data augmentation and LM pre-training. Future work might explore the relationships between these methods and analyze how they can be combined efficiently to benefit from all their advantages.

Multi-task and Pipeline Training. One more promising direction is the joint modeling of many tasks. For example, the joint modeling of NLP pipelines, such as our anonymization pipeline (see Chapter 3), achieved not only performance improvements but also allowed a structured information flow, e.g., to perform model-internal anonymization. More work in this direction could possibly also includes future research on a joint model for temporal tagging based on our methods (see Chapter 8). In addition, the end-to-end learning of many tasks in a single model instead of using pipeline structures has been addressed re-

cently in larger scales for dozens of tasks and demonstrated good results for many different tasks (Raffel et al., 2020), languages (Xue et al., 2021) and modalities, e.g., text and images (Radford et al., 2021; Lin et al., 2021). However, all of these models require large-scale datasets for each task. Therefore, an interesting future direction is multi-task training in non-standard domains that considers the existing resource constraints. Such a model has to be trained in a clever way to learn the more general shared knowledge without forgetting task-specific information.

Summary. The observations made in this thesis and the above-discussed directions highlight the importance of research on domain and language-robust information extraction. We made many vital contributions in this field by taking significant steps towards robust input representations in non-standard domains and languages with our proposed methods. We are optimistic that the successes of our techniques can be generalized to other low-resource scenarios which were not part of our comprehensive evaluations and that higher-level tasks relying on these information extraction models will equally benefit.

List of Figures

List of Tables

Bibliography

Heike Adel and Hinrich Schütze. Exploring different dimensions of attention for uncertainty detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 22–34. Association for Computational Linguistics, 2017.

David Ifeoluwa Adelani, Michael A. Hedderich, Dawei Zhu, Esther van den Berg, and Dietrich Klakow. Distant supervision and noisy label learning for low resource named entity recognition: A study on hausa and yorùbá. *Workshop on Practical Machine Learning for Developing Countries at ICLR'20, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. DocBERT: BERT for document classification. *CoRR*, abs/1904.08398, 2019.

Gustavo Aguilar, Bryan McCann, Tong Niu, Nazneen Rajani, Nitish Shirish Keskar, and Thamar Solorio. Char2subword: Extending the subword embedding space using robust character compositionality. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 1640–1651. Association for Computational Linguistics, 2021.

Lars Ahrenberg. Converting an english-swedish parallel treebank to universal dependencies. In *Proceedings of the Third International Conference on Dependency Linguistics, DepLing 2015, August 24-26 2015, Uppsala University, Uppsala, Sweden*, pages 10–19. Uppsala University, Department of Linguistics and Philology, 2015.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649. Association for Computational Linguistics, 2018.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 724–728. Association for Computational Linguistics, 2019.

Jesujoba O. Alabi, Kwabena Amponsah-Kaakyire, David Ifeoluwa Adelani, and Cristina España-Bonet. Massive vs. curated embeddings for low-resourced languages: the case of yorùbá and twi. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 2754–2762. European Language Resources Association, 2020.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

Begoña Altuna, María Jesús Aranzabe, and Arantza Díaz de Ilarraza. Adapting TimeML to basque: Event annotation. In *Computational Linguistics and Intelligent Text Processing - 17th International Conference, CICLing 2016, Konya, Turkey, April 3-9, 2016, Revised Selected Papers, Part II*, volume 9624 of *Lecture Notes in Computer Science*, pages 565–577. Springer, 2016.

Begoña Altuna, María Jesús Aranzabe, and Arantza Díaz de Ilarraza. Eustimeml: A markup language for temporal information in basque. *Research in Corpus Linguistics*, 8(1): 86–104, May 2020.

Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. Domain adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop, ALTA 2015, Parramatta, Australia, December 8 - 9, 2015*, pages 84–90. Association for Computer Linguistics, 2015.

Alan R. Aronson. Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. In *AMIA 2001, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 3-7, 2001*, page 17. American Medical Association, 2001.

Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610, 2019.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2289–2294. Association for Computer Linguistics, 2016.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5012–5019. AAAI Press, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. How noisy social media text, how diffrnt social media sources? In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 356–364. Asian Federation of Natural Language Processing / Association for Computer Linguistics, 2013.

Fazlourrahman Balouchzahi, Grigori Sidorov, and Hosahalli Lakshmaiah Shashirekha. ADOP fert-automatic detection of occupations and profession in medical texts using flair and BERT. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021), XXXVII International Conference of the Spanish Society for Natural Language Processing., Málaga, Spain, September, 2021*, volume 2943 of *CEUR Workshop Proceedings*, pages 747–757. CEUR-WS.org, 2021.

David Bamman, Sejal Popat, and Sheng Shen. An annotated dataset of literary entities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2138–2144. Association for Computational Linguistics, 2019.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. Self-supervised meta-learning for few-shot natural language classification tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 522–534. Association for Computational Linguistics, 2020.

M. Saiful Bari, Shafiq R. Joty, and Prathyusha Jwalapuram. Zero-resource cross-lingual named entity recognition. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7415–7423. AAAI Press, 2020.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 238–247. Association for Computer Linguistics, 2014.

Maria Barrett, Yova Kementchedjhieva, Yanai Elazar, Desmond Elliott, and Anders Søgaard. Adversarial removal of demographic attributes revisited. 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, Hong Kong, China, November 3-7, 2019*, pages 6329–6334. Association for Computational Linguistics, 2019.

David S. Batista. Named-entity evaluation metrics based on entity-level. *Blogpost*, 05 2018. URL `https://www.davidsbatista.net/blog/2018/05/09/Named_Entity_Evaluation/`. [last accessed March 5, 2022.].

Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

Maximiliana Behnke and Kenneth Heafield. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2664–2674. Association for Computational Linguistics, 2020.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. 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, Hong Kong, China, November 3-7, 2019*, pages 3613–3618. Association for Computational Linguistics, 2019.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, volume 19, pages 137–144. MIT Press, 2006.

Emily M. Bender. The benderrule: On naming the languages we study and why it matters. *The Gradient*, 2019. URL `https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/`. [last accessed March 5, 2022.].

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *FAccT '21:*

2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021, pages 610–623. Association for Computing Machinery, 2021.

Steven Bethard. A synchronous context free grammar for time normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA*, pages 821–826. Association for Computational Linguistics, 2013.

Steven Bethard and Jonathan Parker. A semantically compositional annotation scheme for time normalization. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association, 2016.

Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H. Chi. Data decisions and theoretical implications when adversarially learning fair representations. *CoRR*, abs/1707.00075, 2017.

Kasturi Bhattacharjee, Miguel Ballesteros, Rishita Anubhai, Smaranda Muresan, Jie Ma, Faisal Ladhak, and Yaser Al-Onaizan. To BERT or not to BERT: comparing task-specific and task-agnostic semi-supervised approaches for sequence tagging. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7927–7934. Association for Computational Linguistics, 2020.

Elan Van Biljon, Arnu Pretorius, and Julia Kreutzer. On optimal transformer depth for low-resource language translation. *CoRR*, abs/2004.04418, 2020.

Joachim Bingel and Anders Søgaard. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 164–169. Association for Computational Linguistics, 2017.

André Bittar, Pascal Amsili, Pascal Denis, and Laurence Danlos. French TimeBank: An ISO-TimeML annotated reference corpus. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 130–134. Association for Computer Linguistics, 2011.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146, 2017.

Danushka Bollegala and Cong Bao. Learning word meta-embeddings by autoencoding. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1650–1661. Association for Computational Linguistics, 2018.

Andrew Borthwick. *A Maximum Entropy Approach To Named Entity Recognition*. PhD thesis, New York University, January 06 1999.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. Association for Computer Linguistics, 2015.

Cecil H. Brown, Eric W. Holman, Søren Wichmann, and Viveka Velupillai. Automated classification of the world's languages: a description of the method and preliminary results. *Language Typology and Universals*, 61(4):285–308, 2008.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Curran Associates, 2020.

José Ramom Pichel Campos, Pablo Gamallo Otero, and Iñaki Alegría Loinaz. Measuring diachronic language distance using perplexity: Application to english, portuguese, and spanish. *Nat. Lang. Eng.*, 26(4):433–454, 2020.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. HyperCore: Hyperbolic and co-graph representation for automatic ICD coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3105–3114. Association for Computational Linguistics, 2020a.

Steven Cao, Nikita Kitaev, and Dan Klein. Multilingual alignment of contextual word representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b.

Bob Carpenter. Coding chunkers as taggers: IO, BIO, BMEWO, and BMEWO+. *LingPipe Blog*, 10 2009. URL `https://lingpipe-blog.com/2009/10/14/`. [last accessed March 5, 2022.].

Angel X. Chang and Christopher D. Manning. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 3735–3740. European Language Resources Association, 2012.

Xilun Chen and Claire Cardie. Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1226–1240. Association for Computational Linguistics, 2018.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3098–3112. Association for Computational Linguistics, 2019.

Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. Low-resource domain adaptation for compositional task-oriented semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5090–5100. Association for Computational Linguistics, 2020.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pages 103–111. Association for Computational Linguistics, 2014.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. 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, Hong Kong, China, November 3-7, 2019*, pages 4067–4080. Association for Computational Linguistics, 2019.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1914–1925. Association for Computational Linguistics, 2018.

Joshua Coates and Danushka Bollegala. Frustratingly easy meta-embedding - computing meta-embeddings by averaging source word embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA,*

June 1-6, 2018, Volume 2 (Short Papers), pages 194–198. Association for Computational Linguistics, 2018.

Michael Collins. Log-linear models, MEMMs, and CRFs. *Columbia University lecture*, 2015. [last accessed March 5, 2022.].

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. Association for Computing Machinery, 2008.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067. Curran Associates, 2019.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics, 2020.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 758–759. Association for Computing Machinery, 2009.

Francisco Costa and António Branco. TimeBankPT: A TimeML annotated corpus of portuguese. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 3727–3734. European Language Resources Association, 2012.

Jan Christian Blaise Cruz and Charibeth Cheng. Evaluating language model finetuning techniques for low-resource languages. *CoRR*, abs/1907.00409, 2019.

Quanyu Dai, Xiao Shen, Liang Zhang, Qiang Li, and Dan Wang. Adversarial training methods for network embedding. In *The World Wide Web Conference, WWW 2019,*

San Francisco, CA, USA, May 13-17, 2019, pages 329–339. Association for Computing Machinery, 2019.

Xiang Dai. Recognizing complex entity mentions: A review and future directions. In Vered Shwartz, Jeniya Tabassum, Rob Voigt, Wanxiang Che, Marie-Catherine de Marneffe, and Malvina Nissim, editors, *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, Student Research Workshop*, pages 37–44. Association for Computational Linguistics, 2018.

Xiang Dai and Heike Adel. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 3861–3867. International Committee on Computational Linguistics, 2020.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cécile Paris. Cost-effective selection of pretraining data: A case study of pretraining BERT on social media. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1675–1681. Association for Computational Linguistics, 2020.

Hal Daumé III. Frustratingly easy domain adaptation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. Association for Computer Linguistics, 2007.

Berry de Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. NRC at i2b2: one challenge, three practical tasks, nine statistical systems, hundreds of clinical records, millions of useful features. In *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2*. National Research Council Canada, 2010.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 140–147. Association for Computational Linguistics, 2017.

Franck Dernoncourt, Ji Young Lee, Özlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *J. Am. Medical Informatics Assoc.*, 24 (3):596–606, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26 (3):297–302, 1945.

Wentao Ding, Jianhao Chen, Jinmao Li, and Yuzhong Qu. Automatic rule generation for time expression normalization. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3135–3144. Association for Computational Linguistics, 2021.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. Investigating meta-learning algorithms for low-resource natural language understanding tasks. 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, Hong Kong, China, November 3-7, 2019*, pages 1192–1197. Association for Computational Linguistics, 2019.

Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Rotem Dror, Gili Baumer, Marina Bogomolov, and Roi Reichart. Replicability analysis for natural language processing: Testing significance with multiple datasets. *Trans. Assoc. Comput. Linguistics*, 5:471–486, 2017.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1383–1392. Association for Computational Linguistics, 2018.

Jeffrey L. Elman. Finding structure in time. *Cogn. Sci.*, 14(2):179–211, 1990.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 968–988. Association for Computational Linguistics, 2021.

Lisa Ferro, Laurie Gerber, Inderjeet Mani, Beth Sundheim, and George Wilson. TIDES 2005 standard for the annotation of temporal expressions. *The MITRE Corporation*, 2005.

Clément Feutry, Pablo Piantanida, Yoshua Bengio, and Pierre Duhamel. Learning anonymized representations with adversarial neural networks. *CoRR*, abs/1802.09386, 2018.

Jenny Rose Finkel and Christopher D. Manning. Joint parsing and named entity recognition. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA*, pages 326–334. Association for Computer Linguistics, 2009.

Jenny Rose Finkel, Shipra Dingare, Huy Nguyen, Malvina Nissim, Christopher D. Manning, and Gail Sinclair. Exploiting context for biomedical entity recognition: From syntax to the web. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, NLPBA/BioNLP 2004, Geneva, Switzerland, August 28-29, 2004*. The COLING 2004 Organizing Committee, 2004.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017.

John Rupert Firth. Ethnographic analysis and language with reference to malinowski's views. *Man and Culture: an evaluation of the work of Bronislaw Malinowski*, pages 93–118, 1957.

Corina Forascu and Dan Tufis. Romanian timebank: An annotated parallel corpus for temporal information. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 3762–3766. European Language Resources Association, 2012.

Annemarie Friedrich, Heike Adel, Federico Tomazic, Johannes Hingerl, Renou Benteau, Anika Marusczyk, and Lukas Lange. The sofc-exp corpus and neural approaches to information extraction in the materials science domain. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1255–1268. Association for Computational Linguistics, 2020.

Max Friedrich, Arne Köhn, Gregor Wiedemann, and Chris Biemann. Adversarial learning of privacy-preserving text representations for de-identification of medical records. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5829–5839. Association for Computational Linguistics, 2019.

Pablo Gamallo, José Ramom Pichel, and Iñaki Alegria. From language identification to language distance. *Physica A: Statistical Mechanics and its Applications*, 484:152–162, 2017.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 513–520. Omnipress, 2011.

Christoph Goller and Andreas Küchler. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96), Washington, DC, USA, June 3-6, 1996*, pages 347–352. Institute of Electrical and Electronics Engineers, 1996.

Aitor Gonzalez-Agirre, Montserrat Marimon, Ander Intxaurrondo, Obdulia Rabal, Marta Villegas, and Martin Krallinger. Pharmaconer: Pharmacological substances, compounds and proteins named entity recognition track. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 1–10. Association for Computational Linguistics, 2019.

Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradient-based neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014a.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680. Curran Associates, 2014b.

Kyle Gorman and Steven Bedrick. We need to talk about standard splits. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2786–2791. Association for Computational Linguistics, 2019.

Natalia Grabar and Thierry Hamon. WikiWars-UA: Ukrainian corpus annotated with temporal expressions. *Computational Linguistics and Intelligent Systems*, 2:22–31, 2019.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomás Mikolov. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association, 2018.

Daniel Grießhaber, Ngoc Thang Vu, and Johannes Maucher. Low-resource text classification using domain-adversarial learning. *Comput. Speech Lang.*, 62:101056, 2020.

Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuanjing Huang. Part-of-speech tagging for twitter with adversarial neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copen-*

hagen, Denmark, September 9-11, 2017, pages 2411–2420. Association for Computational Linguistics, 2017.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8342–8360. Association for Computational Linguistics, 2020.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. Demix layers: Disentangling domains for modular language modeling. *CoRR*, abs/2108.05036, 2021.

Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, Marc Pàmies, Joan Llop-Palao, Joaquín Silveira-Ocampo, Casimiro Pio Carrino, Aitor Gonzalez-Agirre, Carme Armentano-Oller, Carlos Rodríguez Penagos, and Marta Villegas. Spanish language models. *CoRR*, abs/2107.07253, 2021.

Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789), 2000.

Kai Hakala and Sampo Pyysalo. Biomedical named entity recognition with multilingual BERT. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 56–61. Association for Computational Linguistics, 2019.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259, 2021.

Fadi Hassan, Mohammed Jabreel, Najlaa Maaroof, David Sánchez, Josep Domingo-Ferrer, and Antonio Moreno. Recrf: Spanish medical document anonymization using automatically-crafted rules and CRF. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, volume 2421 of *CEUR Workshop Proceedings*, pages 727–734. CEUR-WS.org, 2019.

R Brian Haynes, K Ann McKibbon, Nancy L Wilczynski, Stephen D Walter, and Stephen R Werre. Optimal search strategies for retrieving scientifically strong studies of treatment from medline: analytical survey. *The BMJ*, 330:1179, 2005.

Han He and Jinho D. Choi. The stem cell hypothesis: Dilemma behind multi-task learning with transformer encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5555–5577. Association for Computational Linguistics, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. Institute of Electrical and Electronics Engineers, 2016.

Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 187–197. Association for Computational Linguistics, 2011.

Michael A. Hedderich, David Ifeoluwa Adelani, Dawei Zhu, Jesujoba O. Alabi, Udia Markus, and Dietrich Klakow. Transfer learning and distant supervision for multilingual transformer models: A study on african languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2580–2591. Association for Computational Linguistics, 2020.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2545–2568. Association for Computational Linguistics, 2021a.

Michael A. Hedderich, Lukas Lange, and Dietrich Klakow. ANEA: distant supervision for low-resource named entity recognition. In *Practical Machine Learning for Developing Countries Workshop at ICLR 2021, Online, May 3-7, 2021*, 2021b.

Benjamin Heinzerling and Michael Strube. Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association, 2018.

Benjamin Heinzerling and Michael Strube. Sequence tagging with contextual and non-contextual subword representations: A multilingual evaluation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 273–291. Association for Computational Linguistics, 2019.

Wahed Hemati and Alexander Mehler. LSTMVoter: chemical named entity recognition using a conglomerate of sequence labeling tools. *J. Cheminformatics*, 11(1):3:1–3:7, 2019.

Felix Hill, Kyunghyun Cho, Sébastien Jean, Coline Devin, and Yoshua Bengio. Embedding word similarity with neural machine translation. In Yoshua Bengio and Yann LeCun, ed-

itors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9 (8):1735–1780, 1997.

Maximilian Hofer, Andrey Kormilitzin, Paul Goldberg, and Alejo J. Nevado-Holgado. Few-shot learning for named entity recognition in medical text. *CoRR*, abs/1811.05468, 2018.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK*, pages 782–792. Association for Computational Linguistics, 2011.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Boyd Adriane. spacy: Industrial-strength natural language processing in python. To appear, 2017.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339. Association for Computational Linguistics, 2018.

Tsung-Yuan Hsu, Chi-Liang Liu, and Hung-yi Lee. Zero-shot reading comprehension by cross-lingual transfer learning with multi-lingual language representation model. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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, Hong Kong, China, November 3-7, 2019*, pages 5932–5939. Association for Computational Linguistics, 2019.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR, 2020.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. ClinicalBERT: Modeling clinical notes and predicting hospital readmission. *CoRR*, abs/1904.05342, 2019.

Yuyun Huang and Jinhua Du. Self-attention enhanced CNNs and collaborative curriculum learning for distantly supervised relation extraction. 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, Hong Kong,*

China, November 3-7, 2019, pages 389–398. Association for Computational Linguistics, 2019.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Søgaard. DaNE: A named entity resource for Danish. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4597–4604. European Language Resources Association, 2020.

Ander Intxaurrondo. SPACCC (spanish clinical case corpus) tokenizer, March 2019.

Ayush Jain and Meenachi Ganesamoorty. NukeBERT: A pre-trained language model for low resource nuclear domain. *CoRR*, abs/2003.13821, 2020.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3543–3556. Association for Computational Linguistics, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

Kasper Jensen, Cristina Soguero-Ruiz, Karl Oyvind Mikalsen, Rolv-Ole Lindsetmo, Irene Kouskoumvekaki, Mark Girolami, Stein Olav Skrovseth, and Knut Magne Augestad. Analysis of free text in electronic health records for identification of cancer patient trajectories. *Scientific reports*, 7:46226, 2017.

Jing Jiang. Information extraction from text. In *Mining Text Data*, pages 11–41. Springer, 2012.

Zhengbao Jiang, Wei Xu, Jun Araki, and Graham Neubig. Generalizing natural language analysis through span-relation representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2120–2133. Association for Computational Linguistics, 2020.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G

Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Michael I. Jordan. Chapter 25 - serial order: A parallel distributed processing approach. In *Neural-Network Models of Cognition*, volume 121 of *Advances in Psychology*, pages 471–495. North-Holland, 1997.

Armand Joulin, Piotr Bojanowski, Tomás Mikolov, Hervé Jégou, and Edouard Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2979–2984. Association for Computational Linguistics, 2018.

Kohei Kajiyama, Hiromasa Horiguchi, Takashi Okumura, Mizuki Morita, and Yoshinobu Kano. De-identifying free text of Japanese dummy electronic health records. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis, Louhi@EMNLP 2018, Brussels, Belgium, October 31 - November 4, 2018*, pages 65–70. Association for Computational Linguistics, 2018.

Emmanouil I. Kapernaros. Extending the temporal tagger heideltime for the greek language. Master's thesis, National and Kapodistrian University of Athens, 2020.

Sarvnaz Karimi, Xiang Dai, Hamed Hassanzadeh, and Anthony N. Nguyen. Automatic diagnosis coding of radiology reports: A comparison of deep learning and conventional classification methods. In *BioNLP 2017, Vancouver, Canada, August 4, 2017*, pages 328–332. Association for Computational Linguistics, 2017.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. Low-resource deep entity resolution with transfer and active learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5851–5861. Association for Computational Linguistics, 2019.

Phillip Keung, Yichao Lu, and Vikas Bhardwaj. Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and NER. 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, Hong Kong, China, November 3-7, 2019*, pages 1355–1360. Association for Computational Linguistics, 2019.

Muhammad Raza Khan, Morteza Ziyadi, and Mohamed Abdelhady. MT-BioNER: Multitask learning for biomedical named entity recognition using deep bidirectional transformers. *CoRR*, abs/2001.08904, 2020.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. Dynamic meta-embeddings for improved sentence representations. In *Proceedings of the 2018 Conference on Empirical*

Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 1466–1477. Association for Computational Linguistics, 2018.

Hyunjae Kim and Jaewoo Kang. How do your biomedical named entity models generalize to novel entities? *CoRR*, abs/2101.00160, 2021.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. Cross-lingual transfer learning for POS tagging without cross-lingual resources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2832–2838. Association for Computational Linguistics, 2017.

Yoon Kim. Convolutional neural networks for sentence classification. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, October 25-29, 2014*, pages 1746–1751. Association for Computational Linguistics, 2014.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. Improving sentence compression by learning to predict gaze. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1528–1533. Association for Computer Linguistics, 2016.

Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. From zero to hero: Human-in-the-loop entity linking in low resource domains. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6982–6993. Association for Computational Linguistics, 2020.

Roman Klinger and Katrin Tomanek. Classical probabilistic models and conditional random fields. *Algorithm Engineering Report, TR07-2-013*, 2007.

Jan Kocon, Marcin Oleksy, Tomasz Bernas, and Michal Marcinczuk. Results of the poleval 2019 shared task 1: Recognition and normalization of temporal expressions. *Proceedings of the PolEval2019 Workshop, Warsaw, Poland, May 31, 2019*, page 9, 2019.

Ilia Korvigo, Maxim Holmatov, Anatolii Zaikovskii, and Mikhail Skoblov. Putting hands to rest: efficient deep CNN-RNN architecture for chemical named entity recognition with no hand-crafted rules. *J. Cheminformatics*, 10(1):28, 2018.

Robin Kurtz, Stephan Oepen, and Marco Kuhlmann. End-to-end negation resolution as graph parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies, IWPT 2020, Online, July 9, 2020*, pages 14–24. Association for Computational Linguistics, 2020.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann, 2001.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270. Association for Computer Linguistics, 2016.

Lukas Lange, Heike Adel, and Jannik Strötgen. NLNDE: The neither-language-nor-domain-experts' way of spanish medical document de-identification. In *Proceedings of The Iberian Languages Evaluation Forum (IberLEF 2019)*, CEUR Workshop Proceedings, 2019a.

Lukas Lange, Heike Adel, and Jannik Strötgen. NLNDE: enhancing neural sequence taggers with attention and noisy channel for robust pharmacological entity detection. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 26–32. Association for Computational Linguistics, 2019b.

Lukas Lange, Michael A. Hedderich, and Dietrich Klakow. Feature-dependent confusion matrices for low-resource NER labelingvwith noisy labels. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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, Hong Kong, China, November 3-7, 2019*, pages 3552–3557. Association for Computational Linguistics, 2019c.

Lukas Lange, Heike Adel, and Jannik Strötgen. On the choice of auxiliary languages for improved sequence tagging. In *Proceedings of the 5th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2020, Online, July 9, 2020*, pages 95–102. Association for Computational Linguistics, 2020a.

Lukas Lange, Heike Adel, and Jannik Strötgen. Closing the gap: Joint de-identification and concept extraction in the clinical domain. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6945–6952. Association for Computational Linguistics, 2020b.

Lukas Lange, Xiang Dai, Heike Adel, and Jannik Strötgen. NLNDE at CANTEMIST: neural sequence labeling and parsing approaches for clinical concept extraction. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020*, volume 2664 of *CEUR Workshop Proceedings*, pages 335–346. CEUR-WS.org, 2020c.

Lukas Lange, Anastasiia Iurshina, Heike Adel, and Jannik Strötgen. Adversarial alignment of multilingual models for extracting temporal expressions from text. In *Proceedings of the 5th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2020, Online, July 9, 2020*, pages 103–109. Association for Computational Linguistics, 2020d.

Lukas Lange, Heike Adel, and Jannik Strötgen. Boosting transformers for job expression extraction and classification in a low-resource setting. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021), XXXVII International Conference of the Spanish Society for Natural Language Processing., Málaga, Spain, September, 2021*, volume 2943 of *CEUR Workshop Proceedings*, pages 737–746. CEUR-WS.org, 2021a.

Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. FAME: Feature-based adversarial meta-embeddings for robust input representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8382–8395. Association for Computational Linguistics, 2021b.

Lukas Lange, Jannik Strötgen, Heike Adel, and Dietrich Klakow. To share or not to share: Predicting sets of sources for model transfer learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8744–8753. Association for Computational Linguistics, 2021c.

Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. Clin-x: pre-trained language models and a study on cross-task transfer for concept extraction in the clinical domain. *(under submission at Oxford Bioinformatics)*, 2022a.

Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. Multilingual normalization of temporal expressions with masked language models. *(under submission at the ACL Rolling Review)*, 2022b.

Egoitz Laparra, Dongfang Xu, and Steven Bethard. From characters to time intervals: New paradigms for evaluation and neural parsing of time normalizations. *Trans. Assoc. Comput. Linguistics*, 6:343–356, 2018.

Anne Lauscher, Vinit Ravishankar, Ivan Vulic, and Goran Glavas. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4483–4499. Association for Computational Linguistics, 2020.

Robert Leaman and Zhiyong Lu. Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinform.*, 32(18):2839–2846, 2016.

Robert Leaman, Rezarta Islamaj Dogan, and Zhiyong Lu. Dnorm: disease name normalization with pairwise learning to rank. *Bioinform.*, 29(22):2909–2917, 2013.

Robert Leaman, Ritu Khare, and Zhiyong Lu. Challenges in clinical natural language processing for automated disorder normalization. *J. Biomed. Informatics*, 57:28–37, 2015a.

Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. tmchem: a high performance approach for chemical named entity recognition and normalization. *J. Cheminformatics*, 7(S-1): S3, 2015b.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. Transfer learning for named-entity recognition with neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association, 2018.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4):1234–1240, 2020.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1437–1447. Association for Computer Linguistics, 2014.

Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710, 1966.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.*, 34(1):50–70, 2022.

Pengfei Li, Kezhi Mao, Xuefeng Yang, and Qi Li. Improving relation extraction with knowledge-attention. 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, Hong Kong, China, November 3-7, 2019*, pages 229–239. Association for Computational Linguistics, 2019.

Yitong Li, Timothy Baldwin, and Trevor Cohn. Towards robust and privacy-preserving text representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 25–30. Association for Computational Linguistics, 2018.

Jindrich Libovický, Rudolf Rosa, and Alexander Fraser. On the language neutrality of pre-trained multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1663–1674. Association for Computational Linguistics, 2020.

Salvador Lima-López, Eulàlia Farré-Maduell, Antonio Miranda-Escalada, Vicent Brivá-Iglesias, and Martin Krallinger. NLP applied to occupational health: MEDDOPROF shared task at iberlef 2021 on automatic recognition, classification and normalization of professions and occupations from medical texts. *Proces. del Leng. Natural*, 67:243–256, 2021.

Nut Limsopatham and Nigel Collier. Bidirectional LSTM for named entity recognition in twitter messages. In *Proceedings of the 2nd Workshop on Noisy User-generated Text, NUT@COLING 2016, Osaka, Japan, December 11, 2016*, pages 145–152. The COLING 2016 Organizing Committee, 2016.

Junyang Lin, Rui Men, An Yang, Chang Zhou, Yichang Zhang, Peng Wang, Jingren Zhou, Jie Tang, and Hongxia Yang. M6: multi-modality-to-multi-modality multitask mega-transformer for unified pretraining. In *Proceedings of The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 3251–3261. Association for Computing Machinery, 2021.

Ying Lin, Shengqi Yang, Veselin Stoyanov, and Heng Ji. A multi-lingual multi-task architecture for low-resource sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 799–809. Association for Computational Linguistics, 2018.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3125–3135. Association for Computational Linguistics, 2019.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1–10. Association for Computational Linguistics, 2017a.

Qian Liu, Jie Lu, Guangquan Zhang, Tao Shen, Zhihan Zhang, and Heyan Huang. Domain-specific meta-embedding with latent semantic structures. *Inf. Sci.*, 555:410–423, 2021.

Qianchu Liu, Diana McCarthy, Ivan Vulic, and Anna Korhonen. Investigating cross-lingual alignment methods for contextualized embeddings with token-level evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pages 33–43. Association for Computational Linguistics, 2019a.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019b.

Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. De-identification of clinical notes via recurrent neural network and conditional random field. *J. Biomed. Inform.*, 75:S34 – S42, 2017b.

Hector Llorens, Leon Derczynski, Robert J. Gaizauskas, and Estela Saquete. TIMEN: an open temporal expression normalisation resource. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 3044–3051. European Language Resources Association, 2012.

Shayne Longpre, Yu Wang, and Chris DuBois. How effective is task-agnostic data augmentation for pretrained transformers? In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4401–4411. Association for Computational Linguistics, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Yinxia Lou, Yue Zhang, Tao Qian, Fei Li, Shufeng Xiong, and Donghong Ji. A transition-based joint model for disease named entity recognition and normalization. *Bioinform.*, 33(15):2363–2371, 2017.

Yong Luo, Jian Tang, Jun Yan, Chao Xu, and Zheng Chen. Pre-trained multi-view word embedding using two-side neural network. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1982–1988. AAAI Press, 2014.

Jouni Luoma and Sampo Pyysalo. Exploring cross-sentence contexts for named entity recognition with BERT. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 904–914. International Committee on Computational Linguistics, 2020.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. Template-free prompt tuning for few-shot NER. *CoRR*, abs/2109.13532, 2021.

Xuezhe Ma and Eduard H. Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. Association for Computer Linguistics, 2016.

Harish Tayyar Madabushi and Mark Lee. High accuracy rule-based question classification using question syntax and semantics. In *COLING 2016, 26th International Conference*

on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, pages 1220–1230. Association for Computer Linguistics, 2016.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999. ISBN 978-0-262-13360-9.

Montserrat Marimon, Aitor Gonzalez-Agirre, Ander Intxaurrondo, Heidy Rodriguez, Jose Lopez Martin, Marta Villegas, and Martin Krallinger. Automatic de-identification of medical texts in spanish: the MEDDOCAN track, corpus, guidelines, methods and evaluation of results. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, volume 2421 of *CEUR Workshop Proceedings*, pages 618–638. CEUR-WS.org, 2019.

Pawel P. Mazur and Robert Dale. WikiWars: A new corpus for research on temporal expressions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA*, pages 913–922. Association for Computational Linguistics, 2010.

Robert McHardy, Heike Adel, and Roman Klinger. Adversarial training for satire detection: Controlling for confounding variables. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 660–665. Association for Computational Linguistics, 2019.

Sara Meftah, Nasredine Semmar, Mohamed-Ayoub Tahiri, Youssef Tamaazousti, Hassane Essafi, and Fatiha Sadat. Multi-task supervised pretraining for neural domain adaptation. In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media, SocialNLP@ACL 2020, Online, July 10, 2020*, pages 61–71. Association for Computational Linguistics, 2020.

Oren Melamud, Mihaela A. Bornea, and Ken Barker. Combining unsupervised pre-training and annotator rationales to improve low-shot text classification. 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, Hong Kong, China, November 3-7, 2019*, pages 3882–3891. Association for Computational Linguistics, 2019.

Charles E Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. Elsevier, 1978.

Tomás Mikolov, Stefan Kombrink, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, pages 5528–5531. Institute of Electrical and Electronics Engineers, 2011.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a.

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013b.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119. Curran Associates, 2013c.

Tomás Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association, 2018.

Anne-Lyse Minard, Manuela Speranza, Ruben Urizar, Begoña Altuna, Marieke van Erp, Anneleen Schoen, and Chantal van Son. MEANTIME, the NewsReader multilingual event and time corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*, pages 4417–4422, Portorož, Slovenia, 2016. European Language Resources Association.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 1003–1011. Association for Computer Linguistics, 2009.

Antonio Miranda-Escalada, Eulàlia Farré, and Martin Krallinger. Named entity recognition, concept normalization and clinical coding: Overview of the cantemist track for cancer text mining in spanish, corpus, guidelines, methods and results. In *Proceedings of the*

Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020, volume 2664 of *CEUR Workshop Proceedings*, pages 303–323. CEUR-WS.org, 2020.

Paramita Mirza. Recognizing and normalizing temporal expressions in indonesian texts. In Kôiti Hasida and Ayu Purwarianti, editors, *Computational Linguistics - 14th International Conference of the Pacific Association for Computaitonal Linguistics, PACLING 2015, Bali, Indonesia, May 19-21, 2015, Revised Selected Papers*, volume 593 of *Communications in Computer and Information Science*, pages 135–147. Springer, 2015.

Makoto Miwa and Yutaka Sasaki. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, October 25-29, 2014*, pages 1858–1869. Association for Computational Linguistics, 2014.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1101–1111. Association for Computational Linguistics, 2018.

Benjamin Müller, Benoît Sagot, and Djamé Seddah. Can multilingual language models transfer to an unseen dialect? A case study on north african arabizi. *CoRR*, abs/2005.00318, 2020.

Andriy Mulyar, Özlem Uzuner, and Bridget T. McInnes. Mt-clinical BERT: scaling clinical information extraction with multitask learning. *J. Am. Medical Informatics Assoc.*, 28 (10):2108–2115, 2021.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1), 2007.

Usman Naseem, Matloob Khushi, Vinay Reddy, Sakthivel Rajendran, Imran Razzak, and Jinman Kim. Bioalbert: A simple and effective pre-trained language model for biomedical named entity recognition. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pages 1–7. Institute of Electrical and Electronics Engineers, 2021.

Aurélie Névéol, Aude Robert, Francesco Grippo, Claire Morgand, Chiara Orsi, Laszlo Pelikan, Lionel Ramadier, Grégoire Rey, and Pierre Zweigenbaum. CLEF ehealth 2018 multilingual information extraction task overview: ICD10 coding of death certificates in french, hungarian and italian. In *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018*, volume 2125 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018.

Denis Newman-Griffis and Ayah Zirikly. Embedding transfer for low-resource medical named entity recognition: A case study on patient mobility. In *Proceedings of the BioNLP 2018 workshop, Melbourne, Australia, July 19, 2018*, pages 1–11. Association for Computational Linguistics, 2018.

Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth. CogCompTime: A tool for understanding time in natural language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 72–77. Association for Computational Linguistics, 2018.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan T. McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association, 2016.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis M. Tyers, and Daniel Zeman. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4034–4043. European Language Resources Association, 2020.

Juri Opitz and Sebastian Burst. Macro F1 and macro F1. *CoRR*, abs/1911.03347, 2019.

Siim Orasmaa. Towards an integration of syntactic and temporal annotations in estonian. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 1259–1266. European Language Resources Association, 2014.

Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Networks Learn. Syst.*, 32 (2):604–624, 2021.

Aitor García Pablos, Naiara Pérez, and Montse Cuadros. Vicomtech at CANTEMIST 2020. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020*, volume 2664 of *CEUR Workshop Proceedings*, pages 489–498. CEUR-WS.org, 2020.

Md. Rizwan Parvez and Kai-Wei Chang. Evaluating the values of sources in transfer learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*

2021, Online, June 6-11, 2021, pages 5084–5116. Association for Computational Linguistics, 2021.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

Nanyun Peng and Mark Dredze. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 91–100. Association for Computational Linguistics, 2017.

Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of BERT and elmo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 58–65. Association for Computational Linguistics, 2019.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. Association for Computational Linguistics, 2014.

Naiara Pérez, Laura García-Sardiña, Manex Serras, and Arantza del Pozo. Vicomtech at MEDDOCAN: medical document anonymization. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, volume 2421 of *CEUR Workshop Proceedings*, pages 696–703. CEUR-WS.org, 2019.

Martín Pérez-Pérez, Obdulia Rabal, Gael Pérez-Rodríguez, Miguel Vazquez, Florentino Fdez-Riverola, Julen Oyarzábal, A Valencia, Anália Lourenço, and Martin Krallinger. Evaluation of chemical and gene/protein entity recognition systems at biocreative v. 5: the CEMP and GPRO patents tracks. In *Proceedings of the BioCreative Workshop*, volume 5, pages 3–11. University of Minho, 2017.

John P. Pestian, Chris Brew, Pawel Matykiewicz, D. J. Hovermale, Neil Johnson, K. Bretonnel Cohen, and Wlodzislaw Duch. A shared task involving multi-label classification of clinical free text. In *Biological, translational, and clinical language processing, BioNLP@ACL 2007, Prague, Czech Republic, June 29, 2007*, pages 97–104. Association for Computational Linguistics, 2007.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018.

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10186–10203. Association for Computational Linguistics, 2021.

Minh C. Phan, Aixin Sun, and Yi Tay. Robust representation learning of biomedical names. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3275–3285. Association for Computational Linguistics, 2019.

Peter Phandi, Amila Silva, and Wei Lu. SemEval-2018 task 8: Semantic extraction from CybersecURity REports using natural language processing (SecureNLP). In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 697–706. Association for Computational Linguistics, 2018.

Michael Phi. Illustrated guide to LSTM's and GRU's: A step by step explanation. *Towards Data Science*, 09 2018. URL `https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21`. [last accessed March 5, 2022.].

Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4996–5001. Association for Computational Linguistics, 2019.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. Association for Computer Linguistics, 2016.

Nina Pörner, Ulli Waltinger, and Hinrich Schütze. Sentence meta-embeddings for unsupervised semantic textual similarity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7027–7034. Association for Computational Linguistics, 2020.

David M. W. Powers. Evaluation: From predcision, recall and f-factor to roc, informedness, markedness & correlation. *J. Mach. Learn. Technol.*, 2(1):37–63, 2011.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. When BERT plays the lottery, all tickets are winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3208–3229. Association for Computational Linguistics, 2020.

Ruba Priyadharshini, Bharathi R. Chakravarthi, Mani Vegupatti, and John P. McCrae. Named entity recognition for code-mixed indian corpus using meta embedding. In *Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, March 6-7, 2020*, pages 68–72. Institute of Electrical and Electronics Engineers, 2020.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5231–5247. Association for Computational Linguistics, 2020.

James Pustejovsky, Robert Ingria, Roser Saurí, José M. Castaño, Jessica Littman, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Inderjeet Mani. The specification language TimeML. In *The Language of Time - A Reader*, pages 545–558. Oxford University Press, 2005.

Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakosk, and Sophia Ananiadou. Distributional semantics resources for biomedical text processing. In *Proceedings of The International Symposium on Languages in Biology and Medicine (LBM) 2013*, pages 39–44. Oxford University Press, 2013.

Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. [last accessed March 5, 2022.].

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

Edward Raff and Jared Sylvester. Gradient reversal against discrimination: A fair neural network learning approach. In *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*, pages 189–198. Institute of Electrical and Electronics Engineers, 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

Afshin Rahimi, Yuan Li, and Trevor Cohn. Massively multilingual transfer for NER. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 151–164. Association for Computational Linguistics, 2019.

Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP - A survey. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6838–6855. International Committee on Computational Linguistics, 2020.

Lance A. Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora, VLC@ACL 1995, Cambridge, Massachusetts, USA, June 30, 1995*. Springer, 1995.

Lev-Arie Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In Suzanne Stevenson and Xavier Carreras, editors, *Proceedings of the Thirteenth Conference on Computational Natural Language Learning, CoNLL 2009, Boulder, Colorado, USA, June 4-5, 2009*, pages 147–155. Association for Computational Linguistics, 2009.

Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. Combining pretrained high-resource embeddings and subword representations for low-resource languages. In *AfricaNLP Workshop at ICLR'20, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Nils Reimers and Iryna Gurevych. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 338–348. Association for Computational Linguistics, 2017.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4933–4941. European Language Resources Association, 2020.

Kirk Roberts, Halil Kilicoglu, Marcelo Fiszman, and Dina Demner-Fushman. Automatically classifying question types for consumer health questions. In *AMIA 2014, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 15-19, 2014*, page 1018. American Medical Informatics Association, 2014.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1119–1129. Association for Computer Linguistics, 2015.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.

Sebastian Ruder. The 4 biggest open problems in NLP. *Personal Blog*, 01 2019a. URL `https://ruder.io/4-biggest-open-problems-in-nlp`. [last accessed March 5, 2022.].

Sebastian Ruder. *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University of Ireland, Galway, 2019b.

Sebastian Ruder and Barbara Plank. Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 372–382. Association for Computational Linguistics, 2017.

Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. A finnish news corpus for named entity recognition. *Lang. Resour. Evaluation*, 54(1):247–272, 2020.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. Association for Computational Linguistics, 2003.

Erik F. Tjong Kim Sang and Jorn Veenstra. Representing text chunks. In *EACL 1999, 9th Conference of the European Chapter of the Association for Computational Linguistics, June 8-12, 1999, University of Bergen, Bergen, Norway*, pages 173–179. Association for Computer Linguistics, 1999.

Manuela Sanguinetti and Cristina Bosco. Converting the parallel treebank ParTUT in Universal Stanford Dependencies. In *Proceedings of the First Italian Conference on Computational Linguistics (CLiC-it 2014)*, pages 316–321, Pisa, Italy, 2014. Pisa University Press.

Roser Saurı. Annotating temporal relations in catalan and spanish timeml annotation guidelines. Technical report, Technical report, Technical Report BM 2010-04, Barcelona Media, 2010.

Timo Schick, Helmut Schmid, and Hinrich Schütze. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5569–5578. International Committee on Computational Linguistics, 2020.

Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1), 1966.

Fynn Schröder and Chris Biemann. Estimating the influence of auxiliary tasks for multitask learning of sequence tagging tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2971–2985. Association for Computational Linguistics, 2020.

Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, 45(11):2673–2681, 1997.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1599–1613. Association for Computational Linguistics, 2019.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *Commun. ACM*, 63(12):54–63, 2020.

Stefan Schweter and Alan Akbik. FLERT: document-level features for named entity recognition. *CoRR*, abs/2011.06993, 2020.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2931–2951. Association for Computational Linguistics, 2019.

Boaz Shmueli. A tale of two macro-F1's. *Towards Data Science*, 08 2019. URL `https://towardsdatascience.com/a-tale-of-two-macro-f1s-8811ddcf8f04`. [last accessed March 5, 2022.].

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel R. Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. A gold standard dependency corpus for english. In *Proceedings of the Ninth International Conference on Language*

Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014, pages 2897–2904. European Language Resources Association, 2014.

Jasdeep Singh, Bryan McCann, Richard Socher, and Caiming Xiong. BERT is not an interlingua and the bias of tokenization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP, DeepLo@EMNLP-IJCNLP 2019, Hong Kong, China, November 3, 2019*, pages 47–55. Association for Computational Linguistics, 2019.

Luka Skukan, Goran Glavaš, and Jan Šnajder. HeidelTime.hr: extracting and normalizing temporal expressions in Croatian. In *Proceedings of the 9th Slovenian Language Technologies Conferences (IS-LT 2014), Ljubljana, Slovenia, October 9-10, 2014*, pages 99–103, 2014.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Felipe Soares, Marta Villegas, Aitor Gonzalez-Agirre, Martin Krallinger, and Jordi Armengol-Estapé. Medical word embeddings for Spanish: Development and evaluation. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop, ClinicalNLP@NAACL-HLT 2019, Minneapolis, MN, USA, June 7, 2019*, pages 124–133, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. Association for Computer Linguistics, 2016.

Th A Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, 5:1–34, 1948.

Alessandro Sperduti and Antonina Starita. Supervised neural networks for the classification of structures. *IEEE Trans. Neural Networks*, 8(3):714–735, 1997.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2377–2385. Curran Associates, 2015.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd*

Workshop on Noisy User-generated Text, NUT@COLING 2016, Osaka, Japan, December 11, 2016, pages 138–144. The COLING 2016 Organizing Committee, 2016.

Jannik Strötgen and Michael Gertz. WikiWarsDE: A german corpus of narratives annotated with temporal expressions. In *Proceedings of the conference of the German society for computational linguistics and language technology (GSCL 2011)*, pages 129–134. German Society for Computational Linguistics and Language Technology, 2011.

Jannik Strötgen and Michael Gertz. Multilingual and cross-domain temporal tagging. *Lang. Resour. Evaluation*, 47(2):269–298, 2013.

Jannik Strötgen and Michael Gertz. A baseline temporal tagger for all languages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 541–547. Association for Computer Linguistics, 2015.

Jannik Strötgen and Michael Gertz. *Domain-Sensitive Temporal Tagging*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2016. ISBN 978-1-627-05459-1.

Jannik Strötgen, Ayser Armiti, Tran Van Canh, Julian Zell, and Michael Gertz. Time for more languages: Temporal tagging of arabic, italian, spanish, and vietnamese. *ACM Trans. Asian Lang. Inf. Process.*, 13(1):1:1–1:21, 2014a.

Jannik Strötgen, Thomas Bögel, Julian Zell, Ayser Armiti, Tran Van Canh, and Michael Gertz. Extending heideltime for temporal expressions referring to historic dates. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 2390–2397. European Language Resources Association, 2014b.

Jannik Strötgen, Anne-Lyse Minard, Lukas Lange, Manuela Speranza, and Bernardo Magnini. KRAUTS: A german temporally annotated news corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association, 2018.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3645–3650. Association for Computational Linguistics, 2019.

Amber Stubbs and Özlem Uzuner. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *J. Biomed. Informatics*, 58:S20–S29, 2015.

Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *J. Biomed. Informatics*, 58:S11–S19, 2015.

Cong Sun and Zhihao Yang. Transfer learning in biomedical named entity recognition: An evaluation of BERT in the pharmaconer task. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 100–104. Association for Computational Linguistics, 2019.

Tianxiang Sun, Xiangyang Liu, Xipeng Qiu, and Xuanjing Huang. Paradigm shift in natural language processing. *CoRR*, abs/2109.12575, 2021.

Weiyi Sun, Anna Rumshisky, and Özlem Uzuner. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *J. Am. Medical Informatics Assoc.*, 20(5):806–813, 2013.

Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, 2:93–128, 2006.

Jeniya Tabassum, Wei Xu, and Alan Ritter. WNUT-2020 task 1 overview: Extracting entities and relations from wet lab protocols. In *Proceedings of the Sixth Workshop on Noisy User-generated Text, W-NUT@EMNLP 2020 Online, November 19, 2020*, pages 260–267. Association for Computational Linguistics, 2020.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566. Association for Computer Linguistics, 2015.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601. Association for Computational Linguistics, 2019.

Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning, CoNLL 2002, Held in cooperation with COLING 2002, Taipei, Taiwan, 2002*. Association for Computational Linguistics, 2002.

Julien Tourille, Matthieu Doutreligne, Olivier Ferret, Aurélie Névéol, Nicolas Paris, and Xavier Tannier. Evaluation of a sequence tagging tool for biomedical texts. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis, Louhi@EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 193–203. Association for Computational Linguistics, 2018.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. Small and practical BERT models for sequence labeling. 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, Hong Kong, China, November 3-7, 2019*, pages 3630–3634. Association for Computational Linguistics, 2019.

Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nat.*, 571(7763): 95–98, 2019.

Yuta Tsuboi. Neural networks leverage corpus-wide information for part-of-speech tagging. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar*, pages 938–950. Association for Computer Linguistics, 2014.

Özlem Uzuner, Yuan Luo, and Peter Szolovits. Viewpoint paper: Evaluating the state-of-the-art in automatic de-identification. *J. Am. Medical Informatics Assoc.*, 14(5):550–563, 2007.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *J. Am. Medical Informatics Assoc.*, 18(5):552–556, 2011.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James F. Allen, Marc Verhagen, and James Pustejovsky. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013*, pages 1–9. Association for Computer Linguistics, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008. Curran Associates, 2017.

Marta Villegas, Ander Intxaurrondo, Autor Gonzalez-Agirre, Montserrat Marimon, and Martin Krallinger. The MeSpEN resource for english-spanish medical machine translation and terminologies: census of parallel corpora, glossaries and term translations. *Multilingual Biomedical Text Processing Workshop at LREC 2018, Miyazaki, Japan, May 8, 2018*, 2018.

Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory*, 13(2):260–269, 1967.

Ellen M. Voorhees and Dawn M. Tice. The TREC-8 question answering track evaluation. In *Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*. National Institute of Standards and Technology, 1999.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7882–7926. Association for Computational Linguistics, 2020.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. ACE 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45, 2006.

Hao Wang, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li. Learning with noisy labels for sentence-level sentiment classification. 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, Hong Kong, China, November 3-7, 2019*, pages 6285–6291. Association for Computational Linguistics, 2019a.

Haozhou Wang, James Henderson, and Paola Merlo. Weakly-supervised concept-based adversarial learning for cross-lingual word embeddings. 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, Hong Kong, China, November 3-7, 2019*, pages 4418–4429. Association for Computational Linguistics, 2019b.

Leon Weber, Jannes Münchmeyer, Tim Rocktäschel, Maryam Habibi, and Ulf Leser. HUNER: improving biomedical NER with pretraining. *Bioinform.*, 36(1):295–302, 2020.

Leon Weber, Mario Sänger, Jannes Münchmeyer, Maryam Habibi, Ulf Leser, and Alan Akbik. Hunflair: an easy-to-use tool for state-of-the-art biomedical named entity recognition. *Bioinform.*, 37(17):2792–2794, 2021.

Hanna Wecker, Annemarie Friedrich, and Heike Adel. ClusterDataSplit: Exploring challenging clustering-based data splits for model performance evaluation. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, Eval4NLP@EMNLP 2020, Online, November 19, 2020*, pages 155–163. Association for Computational Linguistics, 2020.

Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. 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, Hong Kong, China, November 3-7, 2019, pages 6381–6387. Association for Computational Linguistics, 2019.

Paul J. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356, 1988.

World Health Organization WHO et al. *ICD-O: International classification of diseases for oncology*. World Health Organization, 1976.

World Health Organization WHO et al. *International Statistical Classification of Diseases and Related Health Problems: Alphabetical index*, volume 3. World Health Organization, 2004.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. 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, Hong Kong, China, November 3-7, 2019*, pages 11–20. Association for Computational Linguistics, 2019.

Genta Indra Winata, Zhaojiang Lin, and Pascale Fung. Learning multilingual meta-embeddings for code-switching named entity recognition. In *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2019, Florence, Italy, August 2, 2019*, pages 181–186. Association for Computational Linguistics, 2019a.

Genta Indra Winata, Zhaojiang Lin, Jamin Shin, Zihan Liu, and Pascale Fung. Hierarchical meta-embeddings for code-switching named entity recognition. 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, Hong Kong, China, November 3-7, 2019*, pages 3539–3545. Association for Computational Linguistics, 2019b.

Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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, Hong Kong, China, November 3-7, 2019*, pages 833–844. Association for Computational Linguistics, 2019.

Shijie Wu and Mark Dredze. Are all languages created equal in multilingual bert? In *Proceedings of the 5th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2020, Online, July 9, 2020*, pages 120–130. Association for Computational Linguistics, 2020.

Xin Wu, Yi Cai, Kai Yang, Tao Wang, and Qing Li. Task-oriented domain-specific meta-embedding for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3508–3513. Association for Computational Linguistics, 2020.

Wei Xia, Wen Zhu, Bo Liao, Min Chen, Lijun Cai, and Lei Huang. Novel architecture for long short-term memory used in question classification. *Neurocomputing*, 299:20–31, 2018.

Pengtao Xie, Haoran Shi, Ming Zhang, and Eric P. Xing. A neural architecture for automated ICD coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1066–1076. Association for Computational Linguistics, 2018.

Ying Xiong, Yedan Shen, Yuanhang Huang, Shuai Chen, Buzhou Tang, Xiaolong Wang, Qingcai Chen, Jun Yan, and Yi Zhou. A deep learning-based system for pharmaconer. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, BioNLP-OST@EMNLP-IJNCLP 2019, Hong Kong, China, November 4, 2019*, pages 33–37. Association for Computational Linguistics, 2019.

Ying Xiong, Yuanhang Huang, Qingcai Chen, Xiaolong Wang, Yuan Nic, and Buzhou Tang. A joint model for medical named entity recognition and normalization. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020*, volume 2664 of *CEUR Workshop Proceedings*, pages 499–504. CEUR-WS.org, 2020.

Dongfang Xu, Egoitz Laparra, and Steven Bethard. Pre-trained contextualized character embeddings lead to major improvements in time normalization: a detailed analysis. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics, *SEM@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, pages 68–74. Association for Computational Linguistics, 2019.

Dongfang Xu, Peter A. Jansen, Jaycie Martin, Zhengnan Xie, Vikas Yadav, Harish Tayyar Madabushi, Oyvind Tafjord, and Peter Clark. Multi-class hierarchical question classification for multiple choice science exams. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 5370–5382. European Language Resources Association, 2020.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org, 2015.

Yan Xu, Kai Hong, Junichi Tsujii, and Eric I-Chao Chang. Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries. *J. Am. Medical Informatics Assoc.*, 19(5):824–832, 2012.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 483–498. Association for Computational Linguistics, 2021.

Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2145–2158. Association for Computational Linguistics, 2018.

Hui Yang and Jonathan M. Garibaldi. Automatic detection of protected health information from clinic narratives. *J. Biomed. Informatics*, 58:S30–S38, 2015.

Jie Yang, Shuailong Liang, and Yue Zhang. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3879–3889. Association for Computational Linguistics, 2018.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1480–1489. Association for Computer Linguistics, 2016.

Michihiro Yasunaga, Jungo Kasai, and Dragomir R. Radev. Robust multilingual part-of-speech tagging via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 976–986. Association for Computational Linguistics, 2018.

Wenpeng Yin and Hinrich Schütze. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. Association for Computer Linguistics, 2016.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6470–6476. Association for Computational Linguistics, 2020.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1206–1215. Association for Computational Linguistics, 2018.

Amir Zeldes. The GUM corpus: creating multilayer resources in the classroom. *Lang. Resour. Evaluation*, 51(3):581–612, 2017.

Jiehang Zeng, Lu Liu, and Xiaoqing Zheng. Learning structured embeddings of knowledge graphs with adversarial learning framework. *CoRR*, abs/2004.07265, 2020.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02-03, 2018*, pages 335–340. Association for Computing Machinery, 2018.

Meishan Zhang, Yue Zhang, and Guohong Fu. Cross-lingual dependency parsing using code-mixed treebank. 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, Hong Kong, China, November 3-7, 2019*, pages 997–1006. Association for Computational Linguistics, 2019.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Adversarial training for unsupervised bilingual lexicon induction. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1959–1970. Association for Computational Linguistics, 2017a.

Ye Zhang, Stephen Roller, and Byron C. Wallace. MGNC-CNN: A simple approach to exploiting multiple word embeddings for sentence classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1522–1527. Association for Computer Linguistics, 2016.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 35–45. Association for Computational Linguistics, 2017b.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. Semantics-aware BERT for language understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9628–9635. AAAI Press, 2020.

Sendong Zhao, Ting Liu, Sicheng Zhao, and Fei Wang. A neural multi-task learning framework to jointly model medical named entity recognition and normalization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 817–824. AAAI Press, 2019.

Yue-Shu Zhao, Kunli Zhang, Hongchao Ma, and Kun Li. Leveraging text skeleton for de-identification of electronic medical records. *BMC Medical Informatics Decis. Mak.*, 18 (S-1):65–72, 2018.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3461–3471. Association for Computational Linguistics, 2019.

Yi Zhu, Benjamin Heinzerling, Ivan Vulic, Michael Strube, Roi Reichart, and Anna Korhonen. On the importance of subword information for morphological tasks in truly low-resource languages. In *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pages 216–226. Association for Computational Linguistics, 2019.