Weak Supervision and Label Noise Handling for Natural Language Processing in Low-Resource Scenarios

Michael A. Hedderich

A dissertation submitted towards the degree of

Doctor of Engineering (Dr.-Ing.)

of the Faculty of Mathematics and Computer Science of Saarland University

Saarbrücken, 2022

Day of Colloquium	20.12.2022
Dean of the Faculty	Prof. Dr. Jürgen Steimle

EXAMINATION COMMITTEE

Chair:	Prof. Dr. Verena Wolf
First Reviewer, Advisor:	Prof. Dr. Dietrich Klakow
Second Reviewer:	Prof. Dr. Alan Akbik
Third Reviewer:	Prof. Dr. Katharina Kann
Committee Member:	Dr. Volha Petukhova

ABSTRACT

The lack of large amounts of labeled data is a significant factor blocking many low-resource languages and domains from catching up with recent advancements in natural language processing. To reduce this dependency on labeled instances, weak supervision (semi-)automatically annotates unlabeled data. These labels can be obtained more quickly and cheaply than manual, gold-standard annotations. They also, however, contain more errors. Handling these noisy labels is often required to leverage the weakly supervised data successfully.

In this dissertation, we study the whole weak supervision pipeline with a focus on the task of named entity recognition. We develop a tool for automatic annotation, and we propose an approach to model label noise when a small amount of clean data is available. We study the factors that influence the noise model's quality from a theoretic perspective, and we validate this approach empirically on several different tasks and languages. An important aspect is the aim for a realistic evaluation. We perform our analysis, among others, on several African low-resource languages. We show the performance benefits that can be achieved using weak supervision and label noise modeling. But we also highlight open issues that the field still has to overcome. For the low-resource settings, we expand the analysis to few-shot learning. For classification errors, we present a novel approach to obtain interpretable insights of where classifiers fail.

ZUSAMMENFASSUNG

Der Mangel an annotierten Daten ist ein wesentlicher Faktor, der viele Sprachen und Domänen mit geringen Ressourcen daran hindert, mit den jüngsten Fortschritten in der digitalen Textverarbeitung Schritt zu halten. Um diese Abhängigkeit von gelabelten Trainingsdaten zu verringern, werden bei Weak Supervision nicht gelabelte Daten (halb-)automatisch annotiert. Diese Annotationen sind schneller und günstiger zu erhalten. Sie enthalten jedoch auch mehr Fehler. Oft ist eine besondere Behandlung dieser Noisy Labels notwendig, um die Daten erfolgreich nutzen zu können.

In dieser Dissertation untersuchen wir die gesamte Weak Supervision Pipeline mit einem Schwerpunkt auf den Einsatz für die Erkennung von Entitäten. Wir entwickeln ein Tool zur automatischen Annotation und präsentieren einen neuen Ansatz zur Modellierung von Noisy Labels. Wir untersuchen die Faktoren, die die Qualität dieses Modells aus theoretischer Sicht beeinflussen, und wir validieren den Ansatz empirisch für verschiedene Aufgaben und Sprachen. Ein wichtiger Aspekt dieser Arbeit ist das Ziel einer realistischen Analyse. Die Untersuchung führen wir unter anderem an mehreren afrikanischen Sprachen durch und zeigen die Leistungsvorteile, die durch Weak Supervision und die Modellierung von Label Noise erreicht werden können. Auch erweitern wir die Analyse auf das Lernen mit wenigen Beispielen. In Bezug auf Klassifizierungsfehler, stellen wir zudem einen neuen Ansatz vor, um interpretierbare Erkenntnisse zu gewinnen.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Dietrich Klakow for his support and guidance, for the Friday-croissants and for giving me the liberty to explore the wide landscape of the academic world. I hope that the results fit your requirement of "doing something interesting."

With Jilles Vreeken and Benjamin Roth, I had the luck of finding two mentors that advised me on the PhD life and beyond. Thank you!

At ICLR'21, we organized a Workshop on Weak Supervision which allowed me to broaden my perspectives on this topic. I would like to thank my co-organizers, the speakers and the participants for all the interesting and inspiring discussions.

The departments of computer science and language technology at Saarland Informatics Campus are full of awesome people and I had the pleasure to collaborate with some of them, either on work that went into the text below or on different but not less interesting questions. I would like to thank Adwait, Gabi, David, Dawei, Jesujoba, Johannes, Jonas, Lukas, Marius and Mittul for being such great people to work (and hang-out) with. I'm also grateful to the members of LSV, past and present, for the regular feedback, proofreading and discussions. A special mention goes to our technician Nico for making sure that I can actually do my work, even if that involves resetting the servers on a Sunday morning.

My PhD time was so enjoyable in no small part due to my friends in Saarbrücken and - during that unusual Covid time - in Mainz. Thanks for all the energy I got from you: Alex (2x), Anu, Aravind, Badr, Baltha, Benni, Dana, Dushyant, Ernie, Gianna, Guillermo, Hannah, Iona, Jana, Janine, Janis, Jonas, Josh, Julia, Julian, Kathrin, Kira, Koel, Lena, Magdalena, Mara, Marius, Markus, Merle, Miaoran, Nam, Paskal, Peter, Pia, Sebastian, Simina, Sol, Thomas, Tom and Winfried.

Last but definitely not least, I want to thank my family for their unwavering support: My grandparents who have always encouraged my curiosity, my parents who already proofread my texts long before I imagined writing a PhD thesis, and my siblings who always kept me on my toes.

CONTENTS

1 INTRODUCTION

Advances in natural language processing (NLP) and deep learning have resulted in impressive performance improvements for machine learning models on many tasks. Additionally, more and more machine learning systems have transitioned from purely academic settings to be also used in industry environments and real-life applications. These changes, however, have been limited mainly to English and a selected few other languages. A main reason is that most modern machine learning methods require large amounts of labeled training data. For the majority of languages the expensive and time-consuming manual annotation of large training datasets is not possible. More than 310 languages exist with at least one million L1-speakers each (Eberhard et al., 2019). Similarly, Wikipedia exists for 300 languages.[1] These languages are widely spoken and have access to digital technology. But modern NLP technology only supports a small fraction of them. This leaves behind millions of speakers of so-called low-resource languages. The lack of data becomes even more prevalent if one moves away from standard domains and tasks to more specialized settings. There, even English might be a resource-lean language.

Weak supervision has been proposed as a way to overcome the lack of manually labeled resources. Instead of manually annotating each instance, methods are used that automatically or semi-automatically obtain labels for unlabeled data. A typical example is named entity recognition (NER), the task of recognizing entities such as persons, locations or organizations in text. To automatically annotate such text, words can be matched against a list of entities from a knowledge base (Mintz et al., 2009). If a token appears as name in the list of entities, it is assumed to refer to the entity and is given the corresponding label. This approach allows to label data quickly given the access to external resources such as a knowledge base. Label rules are another form of weak supervision. The expert no longer has to label several hundred instances, a chore that in most cases becomes monotonous and tedious very quickly. Instead, the knowledge and insights the human has about their area of expertise can be directly and efficiently cast into a set of rules that automatically labels the data.

While it is quick and cheap to obtain large amounts of labeled data with weak supervision, the quality of the labels is a major issue. Due to the automatic process, the labels tend to contain many more errors compared to manually annotated data. The performance of a model trained on such noisy labels can therefore be lower than a

1 https://en.wikipedia.org/wiki/List_of_Wikipedias

model trained on just a small amount of manually annotated, noise-free data. To overcome this drawback, different methods to handle label noise have been proposed. Noise-modeling approaches are a major direction. There, one assumes an underlying process of the noise introduced through the weak supervision. An estimated model of the noise process is then used to mitigate the negative effects and successfully leverage the additional, weakly supervised data.

In this thesis, we study methods for training machine learning models in low-resource NLP settings. A particular focus is given to weak supervision and label-noise handling for the task of named entity recognition. We investigate this setting from different angles. This includes presenting methods to quickly obtain weak supervision in realistic low-resource settings as well as theoretical studies of the noise model estimation. We also expand into other settings, experimenting with tasks such as text classification or image classification and examining other low-resource methods such as few-shot transfer learning. Last but not least, we also look at label errors from the view of interpretability, finding explanations for misclassification of black-box machine learning models.

1.1 STRUCTURE AND CONTRIBUTIONS

This thesis is structured with its contributions in the following way:

- In Chapter 3, we give a survey on low-resource methods for NLP. We propose to analyze low-resource settings according to different data dimensions and present the recent literature on low-resource methods in a structured way. We highlight open issues, some of which will also be addressed in this work.
- Chapter 4 introduces the reader to the weak supervision setting for NER as well as a pipeline for noise modeling with a confusion matrix approach. We also present a new method that successfully leverages both a small number of clean labels and a large corpus of noisily labeled text.
- Weak supervision can only be useful if it is easy and fast to obtain. In Chapter 5, we present a tool to obtain distant supervision for NER for many languages and entity types while giving domain experts the possibility to adapt the automatic annotations efficiently. Experiments on several languages and domains show its usefulness as a source of weak supervision.
- Pre-trained language models had a significant impact on the NLP community. In Chapter 6, we analyze how pre-trained, multilingual models like mBERT and RoBERTa can be combined with low-resource techniques on three African languages. For weak supervision, we show that it can be successfully leveraged

in this setting when combined with noise handling. For transfer learning, we highlight its effectiveness when used in a minimal few-shot setting. We also collect new datasets for languages where NER and text classification datasets did not exist yet.

- Chapter 7 approaches noise modeling for weak supervision from a more theoretic perspective identifying the factors that influence the quality of the confusion matrix estimation. The theoretical insights are verified both on synthetic and realistic noisy label data. To this end, we also present NoisyNER, a new dataset that allows to evaluate noise handling methods for complex and realistic noise across multiple noise levels.

- Understanding the reasons behind incorrect classifications is crucial to improve both weak supervision methods and machine learning classifiers in general. In Chapter 8, we present an approach to obtain global explanations for misclassifications of a black-box classifier. To this end, we propose a new mining approach for label-descriptive patterns based on the Minimum Description Length principle. We evaluate the method successfully on synthetic data as well as both visual question answering and NER classifiers.

- This thesis closes in Chapter 9 with a conclusion and ideas for future work.

To improve reproducibility, the code is made available for all works. We refer to each chapter for the specific links.

1.2 NOMENCLATURE

In the existing literature, terms like weak and distant supervision are unfortunately used in a variety of different variations. To avoid confusion, we define these terms here in the way that we see as most consistent with recent literature.

Weak supervision methods annotate unlabeled data in an automatic or semi-automatic way. Manually created, rule-based heuristics are an example of weak supervision as they can be applied automatically to large amounts of unlabeled text without further human effort. *Distant supervision* is a specific form of weak supervision where unlabeled instances are aligned to an external knowledge source in an automatic way. We distinguish weak supervision from semi-supervised learning in that the latter only uses labeled and unlabeled data but no external annotation process.

Noisy-labeled data is data where some of the labels are incorrect. Even manually annotated, gold-standard data usually contains some mistakes. Northcutt et al. (2021) estimate, e.g., an average of 3.4% incorrect instances across ten popular machine learning datasets. For

noisily-labeled data, the percentage of incorrect labels is higher than what is expected of gold-standard data, usually above 20%. In this work, we focus on weakly supervised methods as the origin of the noisy labels. Other data acquisition processes like crowd-sourcing could also be included.

2 PUBLICATIONS

This dissertation summarises several of my works on weak supervision and low-resource learning in collaboration with other authors. Each chapter specifies the overlap between individual parts of this text and publications at other venues. Overall, this thesis includes results from the following publications:

Hedderich & Klakow (2018):
Training a Neural Network in a Low-Resource Setting on Automatically Annotated Noisy Data.
In Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP @ ACL
https://www.aclweb.org/anthology/W18-3402/

Hedderich, Adelani, Zhu, Alabi, Markus & Klakow (2020):
Transfer Learning and Distant Supervision for Multilingual Transformer Models: A Study on African Languages.
In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)
https://www.aclweb.org/anthology/2020.emnlp-main.204/

Hedderich, Zhu & Klakow (2021):
Analysing the Noise Model Error for Realistic Noisy Label Data
In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)
https://ojs.aaai.org/index.php/AAAI/article/view/16938

Hedderich, Lange & Klakow (2021):
ANEA: Distant Supervision for Low-Resource Named Entity Recognition
Presented at Pratical Machine Learning for Developing Countries @ ICLR
https://arxiv.org/abs/2102.13129

Hedderich*, Lange*, Adel, Stötgen & Klakow (2021):
A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios
In Proceedings of the Conference of the North American Chapter of the ACL (NAACL-HLT)
https://www.aclweb.org/anthology/2021.naacl-main.201/

Hedderich*, Fischer*, Vreeken & Klakow (2022):
Label-Descriptive Patterns and their Application to Characterizing Classification Errors
In Proceedings of the 39th International Conference on Machine Learning (ICML)
https://proceedings.mlr.press/v162/hedderich22a.html

* The first authors contributed equally.

3

THE STATE OF LOW-RESOURCE NLP

This chapter[1] gives a broad and structured overview of current efforts on low-resource NLP. We start with a general discussion about low-resource settings and the different dimensions of data availability. Then, we group existing NLP approaches in two main concepts: the generation of additional labeled data (Section 3.5) and transfer learning (Section 3.6). We close with some further ideas from non-NLP communities. In all sections, we discuss promising future directions and open issues, some of which we also address in later chapters.

3.1 INTRODUCTION

Most of today's research in natural language processing (NLP) is concerned with the processing of 10 to 20 high-resource languages with a special focus on English, and thus, ignores thousands of languages with billions of speakers (Bender, 2019). The rise of data-hungry deep learning systems increased the performance of NLP for high resource-languages, but the shortage of large-scale data in less-resourced languages makes their processing a challenging problem. Therefore, Ruder (2019) named NLP for low-resource scenarios one of the four biggest open problems in NLP nowadays.

The umbrella term low-resource covers a spectrum of scenarios with varying resource conditions. It includes work on threatened languages, such as Yongning Na, a Sino-Tibetan language with 40k speakers and only 3k written, unlabeled sentences (Adams et al., 2017). Other languages are widely spoken but seldom addressed by NLP research. Supporting technological developments for low-resource languages can help to increase participation of the speakers' communities in a digital world. Note, however, that 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 domains and tasks, for which – even in English – only little training data is available. Thus, the term "language" in this chapter also includes domain-specific language.

This importance of low-resource scenarios and the significant changes in NLP in the last years have led to active research on resource-lean settings and a wide variety of techniques have been proposed. They all share the motivation of overcoming the lack of labeled data by leveraging further sources. However, these works differ greatly on the

1 This chapter is based on (Hedderich et al., 2021a) with Lukas Lange and Michael Hedderich contributing equally as first authors.

sources they rely on, e.g., unlabeled data, manual heuristics or cross-lingual alignments. Understanding the requirements of these methods is essential for choosing a technique suited for a specific low-resource setting. Thus, one key goal of this survey is to highlight the underlying assumptions these techniques take regarding the low-resource setup. Table 3.1 gives an overview of the surveyed techniques along with their requirements a practitioner needs to take into consideration.

3.2 RELATED SURVEYS

Recent surveys cover low-resource machine translation (Liu et al., 2019a) and unsupervised domain adaptation (Ramponi and Plank, 2020). We refer to these works and do not elaborate on these topics in this chapter, as these tasks are not part of this thesis. We focus instead on general methods for low-resource, supervised natural language processing including data augmentation, distant supervision and transfer learning. This is also in contrast to the task-specific survey by Magueresse et al. (2020) who review highly influential work for several extraction tasks, but only provide little overview of recent approaches. In Table 3.2, we list past surveys that discuss a specific method or low-resource language family for those readers who seek a more specialized follow-up.

3.3 RESOURCE AVAILABILITY FOR DIFFERENT LANGUAGES

To visualize the variety of resource-lean scenarios, Figure 3.1 shows exemplarily which NLP tasks were addressed in different languages from basic to higher-level tasks. While a large number of labeled resources for English are available for many popular NLP tasks, this is not the case for the majority of low-resource languages. To measure which applications are accessible to speakers of low-resource languages we examined resources for six different languages, ranging from high- to low-resource languages for a fixed set of tasks of varying complexity, ranging from basic tasks, such as tokenization, to higher-level tasks, such as question answering.

For this short study, we have chosen the following languages where the number of speakers are the combined L1 and L2 speakers according to Eberhard et al. (2019):

(1) English: The most high-resource language according to the common view and literature in the NLP community.

(2) Yoruba: An African language, which is spoken by about 40 million speakers and contained in the EXTREME benchmark (Hu et al., 2020a). Even with that many speakers, this language is often considered as a low-resource language and it is still

Method	Requirements	Outcome	For low-resource languages	For low-resource domains
Data Augmentation (§ 3.5.1)	labeled data, heuristics*	additional labeled data	✓	✓
Distant Supervision (§ 3.5.2)	unlabeled data, heuristics*	additional labeled data	✓	✓
Cross-lingual projections (§ 3.5.3)	unlabeled data, high-resource labeled data, cross-lingual alignment	additional labeled data	✓	✗
Embeddings & Pre-trained LMs (§ 3.6.1)	unlabeled data	better language representation	✓	✓
LM domain adaptation (§ 3.6.2)	existing LM, unlabeled domain data	domain-specific language representation	✗	✓
Multilingual LMs (§ 3.6.3)	multilingual unlabeled data	multilingual feature representation	✓	✗
Adversarial Discriminator (§ 3.7)	additional datasets	independent representations	✓	✓
Meta-Learning (§ 3.7)	multiple auxiliary tasks	better target task performance	✓	✓

Table 3.1: Overview of low-resource methods surveyed in this chapter. * Heuristics are typically gathered manually.

	Low-resource surveys	Cieri et al. (2016) , Magueresse et al. (2020)
Method-specific	Active learning	Olsson (2009), Settles (2009), Aggarwal et al. (2014)
	Distant supervision	Roth et al. (2013), Smirnova and Cudré-Mauroux (2018), Shi et al. (2019).
	Unsupervised domain adaptation	Wilson and Cook (2020), Ramponi and Plank (2020)
	Meta-Learning	Hospedales et al. (2020)
	Multilingual transfer	Steinberger (2012), Ruder et al. (2019)
	LM pre-training	Rogers et al. (2021), Qiu et al. (2020)
	Machine translation	Liu et al. (2019a)
	Label noise handling	Frenay and Verleysen (2014), Algan and Ulusoy (2021)
	Transfer learning	Pan and Yang (2009), Weiss et al. (2016), Tan et al. (2018)
Language-	African languages	Grover et al. (2010), De Pauw et al. (2011)
	Arabic languages	Al-Ayyoub et al. (2018), Guellil et al. (2019), Younes et al. (2020)
	American languages	Mager et al. (2018)
	South-Asian languages	Daud et al. (2017), Banik et al. (2019), Harish and Rangan (2020)
	East-Asian languages	Yude (2011)

Table 3.2: Overview of existing surveys on low-resource topics.

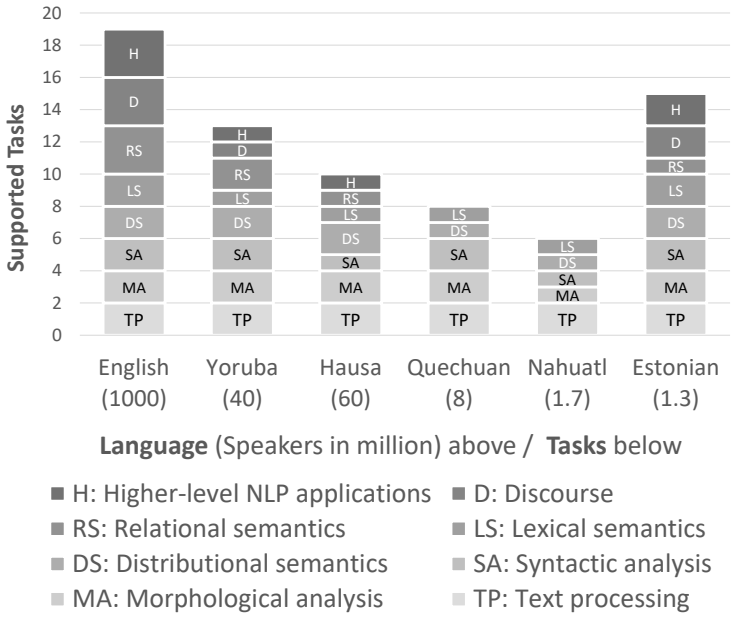


Figure 3.1: Supported NLP tasks in different languages. Note that the figure does not incorporate data quality or system performance.

discussed whether this language is also endangered (Fabuni and Salawu, 2005).

(3) Hausa: An African language with over 60 million speakers. It is not covered in EXTREME or the universal dependencies project (Nivre et al., 2020).

(4) Quechua: A language family encompassing about 8 million speakers, mostly in Peru.

(5) Nahuatl and (6) Estonian: Both have between 1 and 2 million speakers, but are spoken in very different regions (North America & Europe).

The tasks were chosen from a list of popular NLP tasks.[2] We selected two tasks for the lower-lever groups and three tasks for the higher-level groups, which reflects the application diversity with increasing complexity. Table 3.3 shows in detail which tasks were addressed for each language.

Word segmentation, lemmatization, part-of-speech tagging, sentence breaking and (semantic) parsing are covered for Yoruba and Estonian by treebanks from the universal dependencies project (Nivre et al., 2020). Cusco Quechua is listed as an upcoming language in the UD project, but no treebank is accessible at the time of writing. The WikiAnn corpus for named entity recognition (Pan et al., 2017) has resources and tools for NER and sentence breaking for all six languages. Lemmatization resources for Nahuatl were developed by Martinez-Gil et al. (2012) and Lozano et al. (2013) developed resources for part-of-speech tagging, tokenization and parsing of Quechuan. The CoNLL conference and SIGMORPHON organized two shared tasks for morphological reinflection which provided lemmatization resources for many languages, including Quechuan (Cotterell et al., 2018).

Basic resources for simple semantic role labeling and entity linking were developed during the LORELEI program for many low-resource languages (Strassel and Tracey, 2016; Tracey and Strassel, 2020), including resources for Yoruba and Hausa (even though the latter "fell short" according to the authors). Estonian coreference resolution is targeted by Kübler and Zhekova (2016), but the available resources are very limited. Estonian sentiment is done by Pajupuu et al. (2016). All languages are covered by the multilingual fasttext embeddings (Bojanowski et al., 2017) and byte-pair-encoding embeddings (Heinzerling and Strube, 2018). Yoruba, Hausa and Estonian are covered by mBERT or XLM-RoBERTa as well.

Text summarization is done for Estonian by Müürisep and Mutso (2005) and for Hausa by Bashir et al. (2017). The EXTREME benchmark

2 https://en.wikipedia.org/wiki/Natural_language_processing#Common_NLP_Tasks

Group	Task	Yoruba	Hausa	Quechuan	Nahuatl	Estonian
	Num-Speakers	40 mil.	60 mil.	8 mil.	1.7 mil.	1.3 mil.
Text processing	Word segmentation	✓	✓	✓	✓	✓
	Optical character recognition	Hakro et al. (2016)	Hakro et al. (2016)	Hakro et al. (2016)	Hakro et al. (2016)	Hakro et al. (2016)
Morphological analysis	Lemmatization / Stemming	Cotterell et al. (2018)	Cotterell et al. (2018)	Cotterell et al. (2018)	Martinez-Gil et al. (2012)	Cotterell et al. (2018)
	Part-of-Speech tagging	Nivre et al. (2020)	Tukur et al. (2019)	Lozano et al. (2013)	✗	Nivre et al. (2020)
Syntactic analysis	Sentence breaking	✓	✓	✓	✓	✓
	Parsing	Nivre et al. (2020)	✗	Nivre et al. (2020)	✗	Nivre et al. (2020)
Distributional semantics	Word embeddings	FT, BPEmb	FT, BPEmb	FT, BPEmb	FT, BPEmb	FT, BPEmb
	Transformer models	mBERT	XLM-R	✗	✗	mBERT, XLM-R
Lexical semantics	Named entity recognition	Adelani et al. (2020)	Adelani et al. (2020)	Pan et al. (2017)	Pan et al. (2017)	Tkachenko et al. (2013)
	Sentiment analysis	✗	✗	✗	✗	Pajupuu et al. (2016)
Relational semantics	Relationship extraction	✗	✗	✗	✗	✗
	Semantic Role Labelling	Tracey and Strassel (2020)	Tracey and Strassel (2020)	✗	✗	✗
	Semantic Parsing	Nivre et al. (2020)	✗	✗	✗	Nivre et al. (2020)
Discourse	Coreference resolution	✗	✗	✗	✗	Kübler and Zhekova (2016)
	Discourse analysis	✗	✗	✗	✗	
	Textual entailment	Hu et al. (2020a)	✗	✗	✗	Hu et al. (2020a)
Higher-level NLP	Text summarization	✗	Bashir et al. (2017)	✗	✗	Müürisep and Mutso (2005)
	Dialogue management	✗	✗	✗	✗	✗
	Question answering (QA)	Hu et al. (2020a)	✗	✗	✗	Hu et al. (2020a)
	SUM	13	10	8	6	15

Table 3.3: Overview of tasks covered by six different languages. Note that this list is non-exhaustive and due to space reasons we only give one reference per language and task.

(Hu et al., 2020a) covers question answering and natural language inference tasks for Yoruba and Estonian (besides NER, POS tagging and more). Publicly available systems for optical character recognition support all six languages (Hakro et al., 2016). All these tasks are supported for the English language as well, and most often, the English datasets are many times larger and of much higher quality. Some of the previously mentioned datasets were automatically translated, as in the EXTREME benchmark for several languages. Note that we do not claim that all tasks marked in the table yield high-performance model, but we instead indicate if any resources or models can be found for a language.

To summarize, while it is possible to build English NLP systems for many higher-level applications, low-resource languages lack the data foundation for this. Additionally, even if it is possible to create basic systems for tasks, such as tokenization and named entity recognition, for all tested low-resource languages, the training data is typical of lower quality compared to the English datasets, or very limited in size. It also shows that the four American and African languages with between 1.5 and 60 million speakers have been addressed less than the Estonian language, with 1 million speakers. This indicates the unused potential to reach millions of speakers who currently have no access to higher-level NLP applications. Joshi et al. (2020) study further the availability of resources for languages around the world.

3.4 DIMENSIONS OF RESOURCE AVAILABILITY

Many techniques presented in the literature depend on certain assumptions about the low-resource scenario. These have to be adequately defined to evaluate their applicability for a specific setting and to avoid confusion when comparing different approaches. We propose to categorize low-resource settings along the following three dimensions:

(i) The **availability of task-specific labels** in the target language (or target domain) is the most prominent dimension in the context of supervised learning. Labels are usually created through manual annotation, which can be both time- and cost-intensive. Not having access to adequate experts to perform the annotation can also be an issue for some languages and domains.

(ii) The **availability of unlabeled language- or domain-specific text** is another factor, especially as most modern NLP approaches are based on some form of input embeddings trained on unlabeled texts.

(iii) Most of the ideas surveyed in the next sections assume the **availability of auxiliary data** which can have many forms. Transfer learning might leverage task-specific labels in a different language or domain. Distant supervision utilizes external sources of information, such as knowledge bases or gazetteers. Some approaches require

other NLP tools in the target language like machine translation to generate training data. It is essential to consider this as results from one low-resource scenario might not be transferable to another one if the assumptions on the auxiliary data are broken.

3.4.1 *How Low is Low-Resource?*

On the dimension of task-specific labels, different thresholds are used to define low-resource. For part-of-speech (POS) tagging, Garrette and Baldridge (2013) limit the time of the annotators to 2 hours resulting in up to 1-2k tokens. Kann et al. (2020a) study languages that have less than 10k labeled tokens in the Universal Dependency project (Nivre et al., 2020) and Loubser and Puttkammer (2020) report that most available datasets for South African languages have 40-60k labeled tokens.

The threshold is also task-dependent and more complex tasks might also increase the resource requirements. For text generation, Yang et al. (2019) frame their work as low-resource with 350k labeled training instances. Similar to the task, the resource requirements can also depend on the language. Plank et al. (2016) find that task performance varies between language families given the same amount of limited training data.

Given the lack of a hard threshold for low-resource settings, we see it as a spectrum of resource availability. We, therefore, also argue that more work should evaluate low-resource techniques across different levels of data availability for better comparison between approaches. For instance, Plank et al. (2016) and Melamud et al. (2019) show that for very small datasets non-neural methods outperform more modern approaches while the latter obtain better performance in resource-lean scenarios once a few hundred labeled instances are available.

3.5 GENERATING ADDITIONAL LABELED DATA

Faced with the lack of task-specific labels, a variety of approaches have been developed to find alternative forms of labeled data as substitutes for gold-standard supervision. This is usually done through some form of expert insights in combination with automation. We group the ideas into two main categories: data augmentation which uses task-specific instances to create more of them (§ 3.5.1) and distant supervision which labels unlabeled data (§ 3.5.2) including cross-lingual projections (§ 3.5.3). Additional sections cover learning with noisy labels (§ 3.5.4) and involving non-experts (§ 3.5.5).

3.5.1 *Data Augmentation*

New instances can be obtained based on existing ones by modifying the features with transformations that do not change the label. In the computer vision community, this is a popular approach where, e.g., rotating an image is invariant to the classification of an image's content. For text, on the token level, this can be done by replacing words with equivalents, such as synonyms (Wei and Zou, 2019), entities of the same type (Dai and Adel, 2020; Raiman and Miller, 2017) or words that share the same morphology (Gulordava et al., 2018; Vania et al., 2019). Such replacements can also be guided by a language model that takes context into consideration (Fadaee et al., 2017; Kobayashi, 2018).

To go beyond the token level and add more diversity to the augmented sentences, data augmentation can also be performed on sentence parts. Operations that (depending on the task) do not change the label include manipulation of parts of the dependency tree (Dehouck and Gómez-Rodriguez, 2020; Şahin and Steedman, 2018; Vania et al., 2019), simplification of sentences by removal of sentence parts (Şahin and Steedman, 2018) and inversion of the subject-object relation (Min et al., 2020). For whole sentences, paraphrasing through back-translation can be used. This is a popular approach in machine translation where target sentences are back-translated into source sentences (Bojar and Tamchyna, 2011; Hoang et al., 2018). An important aspect here is that errors in the source side/features do not seem to have a large negative effect on the generated target text the model needs to predict. It is therefore also used in other text generation tasks like abstract summarization (Parida and Motlicek, 2019) and table-to-text generation (Ma et al., 2019). Back-translation has also been leveraged for text classification (Hegde and Patil, 2020; Xie et al., 2020). This setting assumes, however, the availability of a translation system. Instead, a language model can also be used for augmenting text classification datasets (Anaby-Tavor et al., 2020; Kumar et al., 2020). It is trained conditioned on a label, i.e., on the subset of the task-specific data with this label. It then generates additional sentences that fit this label. Ding et al. (2020) extend this idea for token level tasks.

Adversarial methods are often used to find weaknesses in machine learning models (Garg and Ramakrishnan, 2020; Jin et al., 2020). They can, however, also be utilized to augment NLP datasets (Morris et al., 2020; Yasunaga et al., 2018). Instead of manually crafted transformation rules, these methods learn how to apply small perturbations to the input data that do not change the meaning of the text (according to a specific score). This approach is often applied on the level of vector representations. For instance, Grundkiewicz et al. (2019) reverse the augmentation setting by applying transformations that flip the (binary)

label. In their case, they introduce errors in correct sentences to obtain new training data for a grammar correction task.

Open Issues: While data augmentation is ubiquitous in the computer vision community and while most of the above-presented approaches are task-independent, it has not found such widespread use in natural language processing. A reason might be that several of the approaches require an in-depth understanding of the language. There is not yet a unified framework that allows applying data augmentation across tasks and languages. Recently, Longpre et al. (2020) hypothesised that data augmentation provides the same benefits as pre-training in transformer models. However, we argue that data augmentation might be better suited to leverage the insights of linguistic or domain experts in low-resource settings when unlabeled data or hardware resources are limited.

3.5.2 *Distant & Weak Supervision*

In contrast to data augmentation, weak supervision uses unlabeled text and keeps it unmodified. The corresponding labels are obtained through a (semi-)automatic process. For named entity recognition (NER), a list of location names might be obtained from a dictionary and matches of tokens in the text with entities in the list are automatically labeled as locations. Distant supervision was introduced by Mintz et al. (2009) for relation extraction (RE) with extensions on multi-instance (Riedel et al., 2010) and multi-label learning (Surdeanu et al., 2012). It is still a popular approach for information extraction tasks like NER and RE where the external information can be obtained from knowledge bases, gazetteers, dictionaries and other forms of structured knowledge sources (Alt et al., 2019; Cao et al., 2019; Deng and Sun, 2019; Lange et al., 2019a; Le and Titov, 2019; Lison et al., 2020; Luo et al., 2017; Nooralahzadeh et al., 2019; Ye et al., 2019). The automatic annotation ranges from simple string matching (Yang et al., 2018) to complex pipelines including classifiers and manual steps (Norman et al., 2019). Weak supervision also encompasses other ideas like reg-ex labeling rules or simple programming functions (Adelani et al., 2020; Karamanolakis et al., 2021; Lison et al., 2020; Ratner et al., 2020; Ren et al., 2020; Zheng et al., 2019).

While distant and weak supervision are popular for information extraction tasks like NER and RE, it is less prevalent in other areas of NLP. Nevertheless, weak supervision has also been successfully employed for other tasks by proposing new ways for automatic annotation. Li et al. (2012) leverage a dictionary of POS tags for classifying unseen text with POS. For aspect classification, Karamanolakis et al. (2019) create a simple bag-of-words classifier on a list of seed words and train a deep neural network on its weak supervision. Wang et al. (2019) use context by transferring a document-level sentiment label

to all its sentence-level instances. Mekala et al. (2020) leverage metadata for text classification and Huber and Carenini (2020) build a discourse-structure dataset using guidance from sentiment annotations. For topic classification, heuristics can be used in combination with inputs from other classifiers like NER. For some classification tasks, the labels can be rephrased with simple rules into sentences. A pre-trained language model then judges the label sentence that most likely follows the unlabeled input (Opitz, 2019; Schick et al., 2020; Schick and Schütze, 2021a). An unlabeled review, for instance, might be continued with "It was great/bad" for obtaining binary sentiment labels.

Open Issues: The popularity of weak and distant supervision for NER and RE might be due to these tasks being particularly suited. There, auxiliary data like entity lists is readily available and distant supervision often achieves reasonable results with simple surface form rules. It is an open question whether a task needs to have specific properties to be suitable for this approach. The existing work on other tasks and the popularity in other fields like image classification (Lee et al., 2018; Li et al., 2020; Li et al., 2017; Mahajan et al., 2018; Xiao et al., 2015) suggest, however, that distant supervision could be leveraged for more NLP tasks in the future.

Distant supervision methods heavily rely on auxiliary data. In a low-resource setting, it might be difficult to obtain not only labeled data but also such auxiliary data. Kann et al. (2020a) find a large gap between the performance on high-resource and low-resource languages for POS tagging pointing to the lack of high-coverage and error-free dictionaries for the weak supervision in low-resource languages. This emphasizes the need for evaluating such methods in a realistic setting and avoiding to just simulate restricted access to labeled data in a high-resource language.

While distant supervision allows obtaining labeled data more quickly than manually annotating every instance of a dataset, it still requires human interaction to create automatic annotation techniques or to provide labeling rules. This time and effort could also be spent on annotating more gold label data, either naively or through an active learning scheme. Unfortunately, distant supervision papers rarely provide information on how long the creation took, making it difficult to compare these approaches. Taking the human expert into the focus connects this research direction with human-computer-interaction and human-in-the-loop setups (see e.g. Klie et al. (2020) and Qian et al. (2020)).

3.5.3 *Cross-Lingual Annotation Projections*

For cross-lingual projections, a task-specific classifier is trained in a high-resource language. Using parallel corpora, the unlabeled low-

resource data is then aligned to its equivalent in the high-resource language where labels can be obtained using the aforementioned classifier. These labels (on the high-resource text) can then be projected back to the text in the low-resource language based on the alignment between tokens in the parallel texts (Yarowsky et al., 2001). This approach can, therefore, be seen as a form of distant supervision specific for obtaining labeled data for low-resource languages. Cross-lingual projections have been applied in low-resource settings for tasks, such as POS tagging and parsing (Akbik et al., 2016; Eskander et al., 2020; Plank and Agić, 2018; Täckström et al., 2013; Wisniewski et al., 2014). Sources for parallel text can be the OPUS project (Tiedemann, 2012a), Bible corpora (Christodoulopoulos and Steedman, 2015; Mayer and Cysouw, 2014) or the recent JW300 corpus (Agić and Vulić, 2019). Instead of using parallel corpora, existing high-resource labeled datasets can also be machine-translated into the low-resource language (Amjad et al., 2020; Fei et al., 2020; Khalil et al., 2019; Zhang et al., 2019a). Cross-lingual projections have even been used with English as a target language for detecting linguistic phenomena like modal sense and telicity that are easier to identify in a different language (Friedrich and Gateva, 2017; Marasović et al., 2016; Zhou et al., 2015).

Open issues: Cross-lingual projections set high requirements on the auxiliary data needing both labels in a high-resource language and means to project them into a low-resource language. Especially the latter can be an issue as machine translation by itself might be problematic for a specific low-resource language. A limitation of the parallel corpora is their domains like political proceedings or religious texts. Mayhew et al. (2017), Fang and Cohn (2017) and Karamanolakis et al. (2020) propose systems with fewer requirements based on word translations, bilingual dictionaries and task-specific seed words, respectively.

3.5.4 *Learning with Noisy Labels*

The above-presented methods allow obtaining labeled data quicker and cheaper than manual annotations. These labels tend, however, to contain more errors. Even though more training data is available, training directly on this noisily-labeled data can actually hurt the performance. Therefore, many recent approaches for distant supervision use a noise handling method to diminish the negative effects of distant supervision. We categorize these into two ideas: noise filtering and noise modeling.

Noise filtering methods remove instances from the training data that have a high probability of being incorrectly labeled. This often includes training a classifier to make the filtering decision. The filtering can remove the instances completely from the training data, e.g., through a probability threshold (Jia et al., 2019), a binary classifier (Adel and

Schütze, 2015; Huang and Du, 2019; Onoe and Durrett, 2019), or the use of a reinforcement-based agent (Nooralahzadeh et al., 2019; Yang et al., 2018). Alternatively, a soft filtering might be applied that re-weights instances according to their probability of being correctly labeled (Le and Titov, 2019) or an attention measure (Hu et al., 2019).

The noise in the labels can also be modeled. A common model is a confusion matrix estimating the relationship between clean and noisy labels (Chen et al., 2019a; Fang and Cohn, 2016; Lange et al., 2019a,c; Luo et al., 2017; Paul et al., 2019; Wang et al., 2019). The classifier is no longer trained directly on the noisily-labeled data. Instead, a noise model is appended which shifts the noisy to the (unseen) clean label distribution. This can be interpreted as the original classifier being trained on a "cleaned" version of the noisy labels. In Ye et al. (2019), the prediction is shifted from the noisy to the clean distribution during testing. In Chen et al. (2020a), a group of reinforcement agents relabels noisy instances. Rehbein and Ruppenhofer (2017), Lison et al. (2020), Simpson et al. (2020) and Ren et al. (2020) leverage several sources of distant or weak supervision and learn how to combine them.

In NER, the noise in distantly supervised labels tends to be false negatives, i.e., mentions of entities that have been missed by the automatic method. Partial annotation learning (Cao et al., 2019; Nooralahzadeh et al., 2019; Yang et al., 2018) takes this into account explicitly. Related approaches learn latent variables (Jie et al., 2019), use constrained binary learning (Mayhew et al., 2019) or construct a loss assuming that only unlabeled positive instances exist (Peng et al., 2019). Zhang et al. (2021) recently presented a benchmark collection to evaluate noise handling methods for NLP and non-NLP tasks.

3.5.5 *Non-Expert Support*

As an alternative to an automatic annotation process, annotations might also be provided by non-experts. Similar to distant supervision, this results in a trade-off between label quality and availability. For instance, Garrette and Baldridge (2013) obtain labeled data from non-native-speakers and without a quality control on the manual annotations. This can be taken even further by employing annotators who do not speak the low-resource language (Mayhew et al., 2019; Mayhew and Roth, 2018; Tsygankova et al., 2021). Lee et al. (2021) propose sorting the instances to train non-experts during the annotation process.

Nekoto et al. (2020) take the opposite direction, integrating speakers of low-resource languages without formal training into the model development process in an approach of participatory research. This is part of recent work on how to strengthen low-resource language communities and grassroot approaches (Adelani et al., 2021; Alnajjar et al., 2020).

3.6 TRANSFER LEARNING

While distant supervision and data augmentation generate and extend task-specific training data, transfer learning reduces the need for labeled target data by transferring learned representations and models. A strong focus in recent works on transfer learning in NLP lies in the use of pre-trained language representations that are trained on unlabeled data like BERT (Devlin et al., 2019). Thus, this section starts with an overview of these methods (§ 3.6.1) and then discusses how they can be utilized in low-resource scenarios, in particular, regarding the usage in domain-specific (§ 3.6.2) or multilingual low-resource settings (§ 3.6.3).

3.6.1 *Pre-Trained Language Representations*

Feature vectors are the core input component of many neural network-based models for NLP tasks. They are numerical representations of words or sentences, as neural architectures do not allow the processing of strings and characters as such. Collobert et al. (2011) showed that training these models for the task of language-modeling on a large-scale corpus results in high-quality word representations, which can be reused for other downstream tasks as well. Subword-based embeddings such as fastText n-gram embeddings (Bojanowski et al., 2017) and byte-pair-encoding embeddings (Heinzerling and Strube, 2018) addressed out-of-vocabulary issues by splitting words into multiple subwords, which in combination represent the original word. Zhu et al. (2019) showed that these embeddings leveraging subword information are beneficial for low-resource sequence labeling tasks, such as named entity recognition and typing, and outperform word-level embeddings. Jungmaier et al. (2020) added smoothing to word2vec models to correct its bias towards rare words and achieved improvements in particular for low-resource settings. 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. 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. This includes pre-trained transformer models (Vaswani et al., 2017), such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019c). 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).

Open Issues: While pre-trained language models achieve significant performance increases compared to standard word embeddings, it is still questionable if these methods are suited for real-world low-

resource scenarios. For example, all of these models require large hardware requirements, in particular, considering that the transformer model size keeps increasing to boost performance (Raffel et al., 2020). Therefore, these large-scale methods might not be suited for low-resource scenarios where hardware is also low-resource.
Biljon et al. (2020) showed that low- to medium-depth transformer sizes perform better than larger models for low-resource languages and Schick and Schütze (2021b) 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 task 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, even for unlabeled data, might not be comparable to data from high-resource languages. Alabi et al. (2020c) found that word embeddings trained on larger amounts of unlabeled data from low-resource languages are not competitive to embeddings trained on smaller, but curated data sources.

3.6.2 *Domain-Specific Pre-Training*

The language of a specialized domain can differ tremendously from what is considered the standard language, thus, many text domains are often less-resourced as well. For example, scientific articles can 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.

One solution to overcome this gap is the adaptation to the target domain by finetuning the language model. Gururangan et al. (2020) showed that continuing the training of 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, i.a. by Adhikari et al. (2019), Alsentzer et al. (2019), Huang et al. (2019), and Lee and Hsiang (2020) and Jain and Ganesamoorty (2020). Most notably are 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 general-domain BERT model performs well in the materials science domain, but the

domain-adapted SciBERT performs best. Xu et al. (2020) used in- and out-of-domain data to pre-train a domain-specific model and adapt it to low-resource domains. Aharoni and Goldberg (2020) found domain-specific clusters in pre-trained language models and showed how these could be exploited for data selection in domain-sensitive training.

Powerful representations can be achieved by combining high-resource embeddings from the general domain with low-resource embeddings from the target domain (Akbik et al., 2018; Lange et al., 2019b). Kiela et al. (2018) showed that embeddings from different domains can be combined using attention-based meta-embeddings, which create a weighted sum of all embeddings. Lange et al. (2020b) further improved on this by aligning embeddings trained on diverse domains using an adversarial discriminator that distinguishes between the embedding spaces to generate domain-invariant representations.

3.6.3 *Multilingual Language Models*

Analogously to low-resource domains, low-resource languages can also benefit from labeled resources available in other high-resource languages. This usually requires the training of multilingual language representations by combining monolingual representations (Lange et al., 2020a) or training a single model for many languages, such as multilingual BERT (Devlin et al., 2019) or XLM-RoBERTa (Conneau et al., 2020). These models are trained using unlabeled, monolingual corpora from different languages and can be used in cross- and multilingual settings, due to many languages seen during pre-training.

In cross-lingual zero-shot learning, no task-specific labeled data is available in the low-resource target language. Instead, labeled data from a high-resource language is leveraged. A multilingual model can be trained on the target task in a high-resource language and afterwards, applied to the unseen target languages, such as for named entity recognition (Hvingelby et al., 2020; Lin et al., 2019), reading comprehension (Hsu et al., 2019), temporal expression extraction (Lange et al., 2020c), or POS tagging and dependency parsing (Müller et al., 2020). Pfeiffer et al. (2020) only adapt parts of the network for a more parameter-efficient training.

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 et al., 2019) as well as pre-trained language models. For example, by aligning the languages inside a single multilingual model, i.a., in cross-lingual (Liu et al., 2019b; Schuster et al., 2019) or multilingual settings (Cao et al., 2020).

This alignment is typically done by computing a mapping between two different embedding spaces, such that the words in both embeddings share similar feature vectors after the mapping (Joulin et al., 2018; Mikolov et al., 2013). This allows to use different embeddings

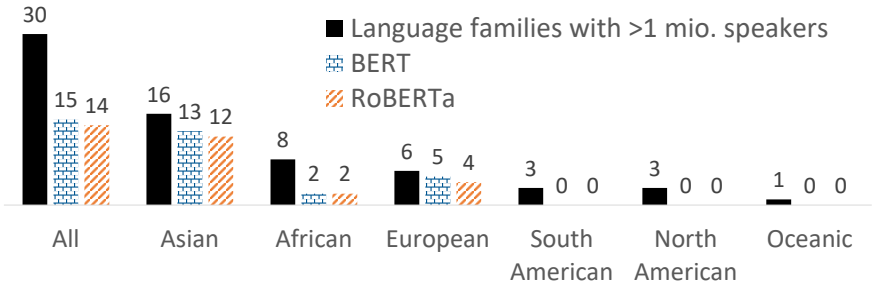


Figure 3.2: Language families with more than 1 million speakers covered by multilingual transformer models.

inside the same model and helps when two languages do not share the same space inside a single model (Cao et al., 2020). For example, Zhang et al. (2019b) 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.

Open Issues: Hu et al. (2020a) showed that for zero-shot learning, there is still a large gap between low and high-resource setting. Also, 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. For example, mBERT covers 104 and XLM-R 100 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 3.2 shows which language families with at least 1 million speakers are covered by mBERT and XLM-RoBERTa?[3] 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. (2020a) showed.

3.7 IDEAS FROM LOW-RESOURCE MACHINE LEARNING IN NON-NLP COMMUNITIES

Training on a limited amount of data is not unique to natural language processing. Other areas, like general machine learning and computer vision, can be a useful source for insights and new ideas. We already presented data augmentation and pre-training. Another example is Meta-Learning (Finn et al., 2017), which is based on multi-task learning. Given a set of auxiliary high-resource tasks and a low-resource

3 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.

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 (Chen et al., 2020b; Yu et al., 2018), natural language understanding (Dou et al., 2019), text classification (Bansal et al., 2020) and dialogue generation (Huang et al., 2020). Instead of having a set of tasks, Rahimi et al. (2019) built an ensemble of language-specific NER models which are then weighted depending on the zero- or few-shot target language.

Differences in the features between the pre-training 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., 2014) can prevent the model from learning a feature-representation that is specific to a data source. Gui et al. (2017), Liu et al. (2017), Kasai et al. (2019), Grießhaber et al. (2020) and Zhou et al. (2019) learned domain-independent representations using adversarial training. Kim et al. (2017), Chen et al. (2018) and Lange et al. (2020c) worked with language-independent representations for cross-lingual transfer. These examples show the beneficial exchange of ideas between NLP and the machine learning community.

3.8 DISCUSSION

Guidelines are necessary to support practitioners in choosing the right tool for their task. In this chapter, we highlighted that it is essential to analyze resource-lean scenarios across the different dimensions of data-availability. This can reveal which techniques are expected to be applicable in a specific low-resource setting. More theoretic and experimental work is necessary to understand how approaches compare to each other and on which factors their effectiveness depends. Longpre et al. (2020), for instance, hypothesized that data augmentation and pre-trained language models yield similar kind of benefits. Often, however, new techniques are just compared to similar methods and not across the range of low-resource approaches. While a fair comparison is non-trivial given the different requirements on auxiliary data, we see this endeavour as essential to improve the field of low-resource learning in the future. This could also help to understand where the different approaches complement each other and how they can be combined effectively.

3.9 CONCLUSION

In this chapter, we gave a structured overview of recent work in the field of low-resource natural language processing. In the following chapters, we will evaluate methods from the two main directions,

obtaining labeled data and transfer learning, with a focus on the former. We will also start addressing some of the identified issues, such as realistic evaluations and testing across different levels of resource availability.

4 THE WEAK SUPERVISION PIPELINE

This chapter[1] demonstrates the complete pipeline of obtaining labeled data via distant supervision and handling incorrect labels via noise-handling. We argue for combining a small amount of clean with a large amount of noisy labels and propose a noise handling method that leverages both. In the experiments, we show the usefulness of modeling the noise for distantly supervised data.

4.1 INTRODUCTION

For training statistical models in a supervised way, labeled datasets are required. For many natural language processing tasks like named entity recognition (NER), every word in a corpus needs to be annotated. While the large effort of manual annotation is regularly done for English, for other languages this is often not the case. And even for English, the corpora are usually limited to certain domains like newspaper articles. For tasks in low-resource areas there tend to be no or only few labeled words available.

In the previous chapter, weak supervision has been presented as an alternative to manually creating labels. These exploit the fact that frequently large amounts of unannotated texts do exist in the targeted domain, e.g. from web crawls. The labels are then assigned using a (semi-)automatic technique like a simple look-ups in knowledge bases or gazetteers. Once such an automatic labeling system is set up, the amount of text to annotate becomes nearly irrelevant, especially in comparison to manual annotation. Also, it is often rather easy to apply the system to different settings, e.g. by using a knowledge base in a different language.

However, while easily obtainable in large amounts, the automatically annotated data usually contains more errors than the manually annotated. When training a machine learning algorithm on such noisy training data, this can result in a low performance. Furthermore, the combination of noisy and clean training instances can perform even worse than just using clean data, as we will see below.

To overcome the negative effects of the noisy training data, we model the noise explicitly using a noise layer that is added to the network architecture. This allows us to directly optimize the network weights using standard techniques. After training, the noise layer is not needed anymore, removing any added complexity.

1 This chapter is based on (Hedderich and Klakow, 2018).

In the noisy label literature, it is common to assume that all training data is noisy (cf. e.g. Berg (2016) and Goldberger and Ben-Reuven (2016)). In our practical experience, it is, however, usually rather easy to also obtain a small clean training set since some data needs to be anyways manually annotated for testing. An additional motivation is the recent trend in few-shot learning. Specific works like (Lauscher et al., 2020a) have shown that it is both realistic and beneficial to assume a small amount of manually labeled instances. This motivates us to also study noise handling in the scenarios where a small amount of clean, gold-standard data, as well as a large amount of noisily labeled data, are available.

This technique is applicable to different classification scenarios. In this chapter, we apply it to Named Entity Recognition (NER). NER is the task of assigning entities in text with their corresponding type (like Person, Organisation or Location). It is a core NLP task and the basis for various applications, from information retrieval to virtual assistants. While there exist some large, hand-annotated corpora like CoNLL03 (Tjong Kim Sang and De Meulder, 2003) or OntoNotes (Weischedel et al., 2011), these are limited to a selected set of languages and domains. If such large corpora are not available, weakly supervised but noisy supervision can be an option to obtain labeled data.

To evaluate the use of noisily-labeled data on a non-synthetic, realistic source of noise, we use distant supervision via look-ups from gazetteers for automatically annotating the data. In the low-resource setting, we show the performance boost obtained from training with both clean and noisy instances and from handling the noise in the data. We also compare to another neural network noise handling approach and we give some more insight into the impact of using additional noisy data and into the learned noise model.

4.2 RELATED WORK

In Chapter 3, we divided noise handling techniques for NLP into noise filtering and noise modeling approaches. Here, we explore a noise modeling technique. Specifically, the idea of a noise channel, as proposed by Bekker and Goldberger (2016). They assume that all clean labels pass through a noisy channel that transforms the clean into a noisy label. One does only observe the noisy labels. The model of the noise channel, as well as the clean labels, are estimated using an EM algorithm. A neural network is then trained on the estimated labels. Berg (2016) applied this model to different tasks, obtaining small improvements on NER with automatically annotated data. A disadvantage of this approach is that the neural network needs to be retrained in every iteration of the EM algorithm, making the model difficult to scale to complex neural architectures.

Goldberger and Ben-Reuven (2016) transformed this model into an end-to-end trainable neural network by replacing the EM component with a noise adaptation layer. They experimented with simple image classification data and Dgani et al. (2018) applied it on the medical image domain. Both limit their approach to only using noisy data. Also, they just evaluate the effectiveness of their noise handling method on simple synthetic noise (uniform and permutation). When applied to real-life scenarios, the noise might have a more complex structure.

Veit et al. (2017) presented an alternative noise modeling approach which also leverages clean labels. It consists of two components. A cleaning network learns to map noisy labels to clean ones. The second network is used to learn the actual task from clean and cleaned labels. We adapt their image classification model to the NER setting and compare our approach to this idea in the experiments.

4.3 NOISE LAYER

Given a clean dataset C consisting of feature and label tuples (x, y), we can construct a multi-label neural network softmax classifier

$$p(y = i|x; w) = \frac{\exp(u_i^T h(x))}{\sum_{j=1}^{k} \exp(u_j^T h(x))} , \tag{4.1}$$

where k is the number of classes, h is a non-linear function or a more complex neural network and w are the network weights including the softmax weights u.

The noisy dataset N is a set of additional training instances. Following the approach of Goldberger and Ben-Reuven (2016), we assume that each originally clean (but unseen) label y went through a noise channel or process transforming it into the noisy label z. We only observe the noisy label, i.e. N consists of tuples (x, z).

The noise transformation from a clean label y with class i to a noisy label z with class j is modeled using a stochastic matrix

$$\theta(i, j) = p(z = j|y = i) = \frac{\exp(b_{ij})}{\sum_{l=1}^{k} \exp(b_{il})} , \tag{4.2}$$

for $i, j \in \{1, ..., k\}$ and where b are learned weights. We call this the noise layer here. The probability for an observed, noisy label then becomes

$$p(z = j|x; w; \theta) = \sum_{i=1}^{k} p(z = j|y = i; \theta) p(y = i|x; w) , \tag{4.3}$$

for $(x, z) \in N$.

In contrast to the work by Goldberger and Ben-Reuven (2016), we also have access to clean data C. From this, we create two models,

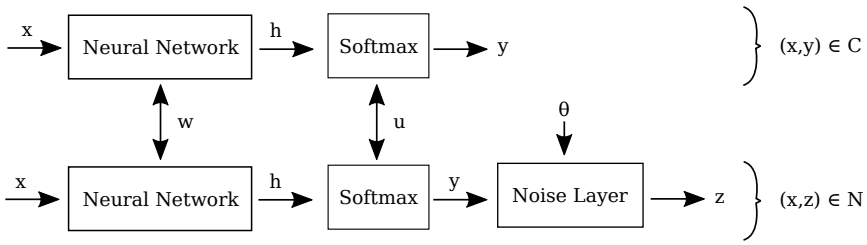


Figure 4.1: General architecture of the approach. Above is the base model trained on clean data C and predicting clean labels y. Below is the noise layer model trained on noisy label data N. The predicted labels y are transformed into the seen, noisy labels z using the noise layer.

as illustrated in Figure 4.1. The base model without noise layer is trained on C and the noise model with the noise layer is trained on N. Both models share the same network weights. The models are trained alternatingly, each for one epoch of its corresponding clean or noisy data. For prediction, the noise layer is removed and just the base model is used.

As stated by Goldberger and Ben-Reuven (2016), the initialization of θ is important. Since we have access to a small amount of clean data C, we use it for initializing the stochastic matrix. We assume that we can create noisy labels for the clean instances using the same process as for the noisy data N. We then initialize the weights of θ as

$$b_{ij} = \log\left(\frac{\sum_{t=1}^{|C|} 1_{\{y_t=i\}} 1_{\{z_t=j\}}}{\sum_{t=1}^{|C|} 1_{\{y_t=i\}}}\right), \tag{4.4}$$

where z_t is obtained by creating a noisy label for $(x_t, y_t) \in C$. In Chapter 7, we will revisit the noise process and the estimation of the stochastic matrix more in detail.

4.4 DATASET AND AUTOMATIC ANNOTATION

Named Entity Recognition (NER) is the task of assigning phrases in a text an entity label. In the sentence

Only France backed Fischler's proposal.

the country *France* is of the entity class location and *Fischler* refers to a person. Creating training data for this task requires that each word in the text is labeled with its corresponding class. The effort to create a sufficiently large dataset might be too large for a low-resource language.

To tackle this problem, Dembowski et al. (2017) proposed to use external lists and gazetteers of entities to automatically label words in a

Class	Precision	Recall	F1
PER	48.09%	25.90%	33.67%
ORG	52.45%	10.02%	16.83%
LOC	56.76%	65.42%	60.78%
MISC	0.00%	0.00%	0.00%
Overall	53.31%	27.36%	36.16%

Table 4.1: Evaluation of the automatic labeling on the full English CoNLL-2003 training set (which we use as noisy dataset N).

training corpus. A list of person names can e.g be extracted from all of the entries appearing in Wikipedia's person category. Equipped with such lists for all entity classes, one can then label a text automatically. A word gets assigned a specific class if it appears in the corresponding entity list. A word or token that does not appear in any list gets assigned the null class "O". Additionally, simple heuristics help to resolve conflicts between lists and to remove some sources of errors. One might e.g. not label the day of the weeks as names, although *"Friday"* might be in the list of person names.

For this work, we use the English CoNLL03 NER corpus (Tjong Kim Sang and De Meulder, 2003). The dataset is labeled with the classes person (PER), location (LOC), organization (ORG), miscellaneous name (MISC) and the null class (O). It consists of a training, a development and a test set. To obtain a low-resource setting, we randomly sample a subset of the training set as clean data C. In the experiments, we vary this size between ca. 400 and 20000 words. The rest of the labels are removed from the training set.

We then label the whole training set using the method by Dembowski et al. (2017) in the version with heuristics. This approach of automatically labeling words allows to quickly obtain large amounts of labeled text. However, both precision and recall tend to be lower than for manually labeled corpora (cf. Table 4.1). It should be noted that the MISC class is not covered with this technique which is an additional source of noise in the automatically annotated data. We use this as our noisy data N.

4.5 MODEL ARCHITECTURES AND TRAINING

In this section, we present the different model architectures we evaluated in our experiments and we give details on the training procedure.

For each instance, the input x is a sequence of words with the target word in the middle surrounded by 3 words from the left and from the right of the original sentence, e.g. x = *"countries other than Britain until the scientific"* where *"Britain"* is the target word with label y = LOC.

Sentence boundaries are padded. We encode the words using the 300-dimensional GloVe vectors trained on cased text from Common Crawl (Pennington et al., 2014).

The **base-model** uses a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) with state size 300 to encode the input. Then a dense layer is applied with size 100 and ReLU activation (Glorot et al., 2011). Afterwards, the softmax layer is used for classification. This model is only trained on the clean data C.

The **noise-model** is built upon the base model and uses the noise layer architecture explained in section 4.3. First, the model is trained without noise layer for one epoch on the clean data. Then, we alternate between training with the noise layer on the noisy data and without the noise layer on the clean, each for one epoch. Instead of training on the full noisy corpus, we use a subsample $\tilde{N}$, randomly picked in each epoch. This allows the model to see many different noisy samples while preventing the noise from being too dominant. In section 4.6.2, we evaluate this effect.

For the **base-model-with-noise** we use the same clean and noisy data but the noise layer is left out, using only the base model architecture without an explicit noise handling technique.

To evaluate the importance of the initialization of the stochastic matrix θ, the **noise-model-with-identity-init** uses the same training approach and data as the *noise-model*. However, θ is initialized with the identity matrix instead of using formula 4.4.

The **noise-adaptation-model** uses the original model of Goldberger and Ben-Reuven (2016). It consists of the base model with the noise layer and is trained on the whole noisy dataset in each epoch. It does not use the clean data. For initializing θ, the base model is pretrained on the noisy data and its predictions are used as an approximation to the clean labels.

We also compare to the recent work by Veit et al. (2017). They train a noise cleaning component which learns to map from a noisy label and a feature representation to a clean label. These cleaned labels are then used for training of what we call the base model. The authors did not report specific layer sizes and their architecture is developed for an image classification task, which differs structurally from our NER dataset (e.g. their label vector is much sparser). We, therefore, adapt their concept to our setting. As feature representation, we use the output of the BiLSTM which is projected to a 30-dimensional space with a linear layer. This is concatenated with the noisy label and used as input to the noise cleaning component. It is passed through a dense layer with the same dimension as the label vector. The skip-connection and clipping are used as in their publication. We use the same training approach and data as with the *noise-model*, replacing the step where the noise layer is trained. Instead, in each epoch the noise cleaning component is trained on C and the corresponding noisy labels. The

base model is then trained on a cleaned version of $\tilde{N}$ and C. We call this the **noise-cleaning-model**.

All models are trained using cross-entropy loss, except for the noise cleaning component of the *noise-cleaning-model* which is trained with the absolute error loss like in the original paper. All models are trained for 40 epochs and the weights of the best performing epoch are selected according to the F1 score on the development set. Adam (Kingma and Ba, 2015) is used for stochastic optimization.

4.6 EXPERIMENTS AND EVALUATION

In this section, we report on our experiments and their results. The training on noisy data as well as the randomness in training neural networks in general lead to a certain amount of variance in the evaluation scores. Therefore, we repeat all experiments five times and report the average as well as the standard error. To obtain meaningful results, no noise is added to the test data.

4.6.1 *Model Comparison*

To simulate different degrees of low-resource settings, we trained the models on different amounts of clean data. We vary the size between 407 labeled words (0.2% of the CoNLL-2003 training data) and 20362 labeled words (10%) in six steps. Since the noisy labels are easy to obtain, we use the whole corpus N. The size of the random subsample $\tilde{N}$ in each epoch is set to the same size as the clean data.

The results of this experiment are given in Figure 4.2. There is a general trend that the larger the amount of clean data is, the lower the differences between the models are. It seems that once we have obtained enough clean training data, the additional noisy data cannot add much more information, even when cleaned. This is reminiscent of results from semi-supervised learning (e.g. in Nigam et al. (2006)).

For the two settings with the lowest amount of data, the *base-model-with-noise* (which is trained on clean and noisy data without a noise channel) performs worst. For the four settings with more data, it is better than *base-model* (which is only trained on C). This could indicate that noisy labels do hurt the performance in low-resource settings. However, once a certain amount of clean training data is obtained, this is enough to cope with the noise to a certain degree and obtain improvements, even when the noise is not explicitly handled.

The models that do handle noise, outperform these baselines. When comparing *noise-model* and *noise-model-with-identity-init*, we see a large gap in performance. This shows the importance of a good initialization of the noise model θ in the low-resource setting.

The original *noise-adaptation-model* model by Goldberger and Ben-Reuven (2016) obtains an average F1 score of 38.8. This shows that a

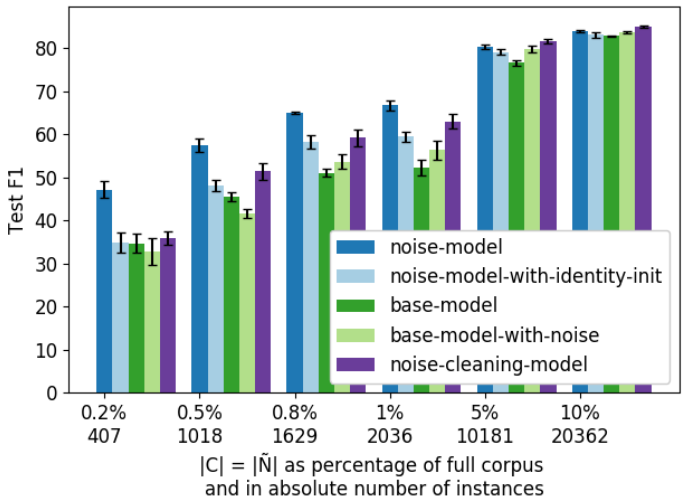


Figure 4.2: Evaluation results of the models. Experiments were run for different sizes of the clean data C and the per epoch randomly subsampled data $\tilde{N}$. The average F1 score on the test set is given over five runs. The error bars show two-times standard error in both directions.

model purely trained on a large amount of automatically annotated data can be an alternative to a model trained on very few clean instances. However, the effect of cleaning noisy labels without access to any clean data seems limited, as the model cannot even reach the performance of either the *base-model* trained on 1018 instances nor our *noise-model* on the smaller set of 407 instances.

Our proposed *noise-model* outperforms the *cleaning-model* in the four lower-resource settings while the latter performs slightly better in the two scenarios with more data. With its access to the features in the noise cleaning component, the *cleaning-model* might be able to model more complex noise transformations. However, it does not seem to be able to leverage this capability in a low-resource setting. In the low-resource settings, our *noise-model* is able to handle the noise well and it gains over ten points in F1 score over not using a noise handling mechanism or only training on clean data.

4.6.2 *Amount of Noisy Data*

In this experiment, we evaluate the effect of using different amounts of noisy data during each epoch, i.e. we vary the size of the subsampled, noisy data $\tilde{N}$. We experiment with the *noise-model* and fix the amount of clean data C to 2036 labeled words (1% of the CoNLL-2003 training data). We choose $|\tilde{N}|$ as multiples of $|C|$ using factors 0.5, 1, 2, 10, 20, 30 and 50.

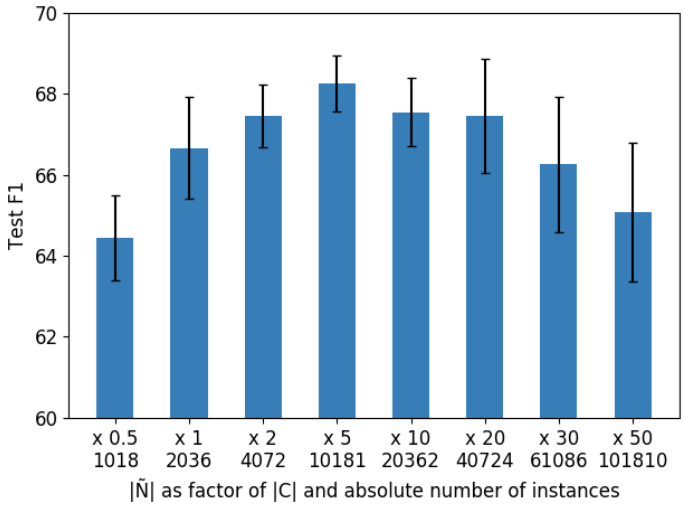


Figure 4.3: Evaluation results for varying the size of the per epoch randomly subsampled noisy data $\tilde{N}$. The *noise-model* was used and the amount of clean data C fixed to 2036 labeled words. The average F1 score on the test set is given over five runs. The error bars show two-times standard error in both directions.

The results are given in Figure 4.3. One can see a trend that increasing the size of $\tilde{N}$ results in an improvement in F1 score. This holds until factor 5. Afterwards, the performance degrades again. This might indicate that the noisy data becomes too dominant and the cleaning effect of the noise layer is not able to mitigate it.

4.6.3 *Learned Weights*

Figure 4.4 shows the stochastic matrix θ that was learned in one run of training the *noise-model* with $|C| = |\tilde{N}| = 2036$ labeled words (1% of the CoNLL-2003 training data).

One can see that the learned weight matrix represents a reasonable model of the noise. For the classes PER, ORG and MISC, the recall is very low in the noisy data and therefore the corresponding weights in the first column of the matrix are high: Instances (or a certain percentage of the probability mass) which the base model correctly classifies as PER/ORG/MISC, are mapped to the class O because this is the most common noisy label for these classes (indicated by the low recall we can see in Table 4.1). For the LOC class, the recall in the noisy labels is much higher and we see this reflected in the learned weights. The highest weight for this class is $\theta_{\text{LOC, LOC}}$, i.e. a prediction of the label LOC is mostly left unchanged because it tends to be correctly labeled in the noisy data.

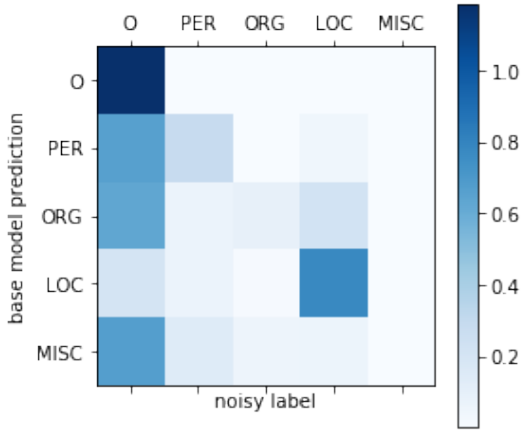


Figure 4.4: Representation of the noise transition weights θ learned in the noise layer. Each square is a value $\exp(\theta_{ij})$ where i is the vertical and j the horizontal index in the visualization.

4.7 CONCLUSIONS AND OPEN QUESTIONS

In this chapter, we presented the pipeline of distant supervision and noise handling for low-resource classification. We proposed a technique to train a neural network on a combination of clean and noisy annotations and to model the noise explicitly using a noise layer. We evaluated our approach on an NER task using real noise in the form of automatically annotated labels. We found that the probabilistic noise matrix learned is a useful model of the noise. In the low-resource setting where only a few manually annotated instances are available, we showed the improvements of up to 35% obtained from using additional, noisy data and handling the noise.

In the experiments, we saw that the ranking of the best performing baselines changed depending on the amount of clean data available. The base-model e.g. developed a certain robustness against noise given enough clean labels. This highlights the importance of evaluating across different levels of resource availability as argued in the previous chapter.

While this first study indicates the usefulness of the approach, several questions are still open. While we used a realistic source of noisy labels, the setting itself - NER for English news text - was only a simulated low-resource scenario. It is necessary to evaluate how this idea performs for true resource-lean settings.

Also, the base model used an LSTM with pre-trained word embeddings. While such a model architecture is comparatively light-weight and easy to deploy in a low-resource scenario, recent advances like BERT (Devlin et al., 2019) have shown much improved performance in the high-resource case. One advantage of the noise handling method we proposed is its independence of the underlying base model. We are,

therefore, interested to see how well this noisy label idea combines with more modern architectures.

Last but not least, this study only evaluated one task and one form of weak supervision. We, therefore, want to test how it performs for different tasks and different sources of noisy labels. These questions are addressed in the following chapters.

5 A TOOL FOR DISTANT SUPERVISION

A limitation of the evaluation from the previous chapter was the use of a simulated low-resource setting with English data. In Chapter 3, we argued that simulations might miss assumptions like the dependency on auxiliary data. This chapter[1] presents a tool that facilitates distant supervision for Named Entity Recognition. It aims to support many languages and entity types including those from realistic low-resource scenarios such as developing countries. It allows to retrieve entity lists and to automatically annotate unlabeled text in an easy to use and fast way while also empowering the domain expert to adapt the automatic annotation.

5.1 INTRODUCTION

Named Entity Recognition is a basic task which is essential for a variety of NLP applications like advanced search methods, personal assistants or information extraction systems. For many low-resource languages and domains, it is, however, not possible to manually label every token of large corpora due to time and resource constraints. As seen in Chapter 3, the absence of labeled data is prevalent for languages from developing countries. We see this as a significant factor limiting the development of NLP technologies in these regions with respect to the ongoing tendency towards data-driven models.

In the previous chapter, we have seen that distant supervision can be a useful training resource in the absence of expensive, high-quality labels. Even in low-resource settings, unlabeled text is often available, and for NER, a widespread approach is to use lists, dictionaries or gazetteers of named entities (e.g. a list of person names or cities). Each word in the corpus is assigned the corresponding named entity label if it appears in this list of entities. Introduced by Mintz et al. (2009), this is still a popular technique and used e.g. by Peng et al. (2019), Adelani et al. (2020) and Lison et al. (2020). For an extensive list of recent works using distant supervision for low-resource NER, we refer to Chapter 3 on related work.

While distant supervision performs very well on high-resource languages and on simulated low-resource settings (like we have seen in Chapter 4), it has been shown to be more difficult to leverage in real low-resource scenarios due to the lack of external information (Kann et al., 2020a). Additionally, several difficulties arise when applying it in a practical way, such as obtaining these dictionaries (e.g. a list of city

1 This chapter is based on (Hedderich et al., 2021b).

names in Yorùbá) or adapting the matching procedure to the specific language and domain (e.g. deciding for or against lemmatization and, thus, trading off recall and precision). Distant supervision can only be beneficial and save resources if it is easy to use and fast to deploy.

The ANEA tool we present provides the functionality to actually use distant supervision approaches in practice for many languages and named entity types while minimizing the amount of manual effort and labeling cost. A process is provided to automatically extract entity names from Wikidata, a free and open knowledge base. The information is used to annotate named entities for large amounts of unlabeled text automatically. The tool also supports the user in tuning the automatic annotation process. It enables language experts to efficiently include their knowledge without having to annotate many tokens manually. Both a library and a graphical user interface are provided to assist users of varying technical backgrounds and different use-cases. In an experimental study on six different scenarios, we show that ANEA outperforms two baselines in nearly all cases regarding the quality of the automatic annotation. When used to provide distantly supervised training data for a neural network model, it creates on average a boost of 18 F1 points with less than 30 minutes of manual interaction. The tool, further information and technical documentation and the additional model code and evaluation data are made publicly available online.[2]

5.2 RELATED WORK

A variety of open-source tools exist to annotate text manually. While their focus is on the manual annotation of data, some support the user with certain degrees of automation. A token can be labeled automatically if it has been labeled before by the user in WebAnno (Yimam et al., 2014) and TALEN (Mayhew and Roth, 2018). In TALEN, a bilingual lexicon can be integrated but just to support annotators that do not speak the text's language. WebAnno and brat (Stenetorp et al., 2012) allow importing the annotations of external tools as suggestions for the user. The focus is, however, still on the user manually checking all tokens. Also, the annotator cannot use their insight to directly influence and improve the external tool like in the tuning process of ANEA.

In the area of information extraction, the tools by Gupta and Manning (2014), Li et al. (2015) and Dalvi et al. (2016) allow the user to create rules or patterns, e.g. "[Material] conducts [Energy]". This can, however, require a large amount of manual rule creation effort to obtain good coverage for NER. With Snorkel (Ratner et al., 2020), a user can define similar and more general labeling functions. Oiwa et al. (2017) presented a tool to create entity lists manually. These

2 https://github.com/uds-lsv/anea

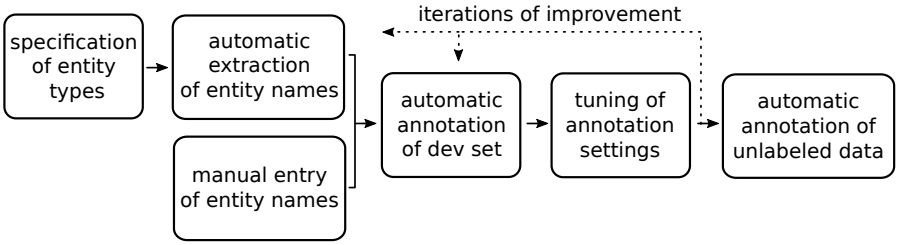


Figure 5.1: Overall workflow of ANEA.

lists could be imported into ANEA. NER is closely related to entity linking. Zhang et al. (2018) presented a system to link entities in many languages automatically but they focus on disaster monitoring and, therefore, only consider persons, geopolitical entities, organizations, and locations.

5.3 WORKFLOW

The workflow is visualized in Figure 5.1 and we provide an online video that shows an exemplary walkthrough.[3] The process is split into four parts:

5.3.1 *Extraction*

The user starts by searching for the category names of the entity types that should be extracted (e.g. *person* or *film*). The tool will then automatically extract the names of all the corresponding entities (e.g. for *person*: "Alan Turing", "Edward Sapir", ...). As the source for the extractions, we use a dump of Wikidata. It is a free and open knowledge base that is created both by manual edits and automatic processes. At the time of writing, it contains over 100 million items.

For most items, the names are available in multiple languages (e.g. 32k person names for Yorùbá or 26k movie names for Spanish). The user searches for and specifies the entity types they want to extract and which language should be used for the names (Figure 5.2). The tool will then extract all items that have the "is an instance of" property of the given entity types. The results are the lists of entity names. Additionally, the user can also provide existing lists of entity names in case of a very specific domain.

5.3.2 *Automatic Annotation*

The automatic annotation is performed by checking each word against the list of extracted entities. A word (or token) is assigned the label of the entity name it matches. If matches of several entity names overlap,

3 https://www.youtube.com/watch?v=eXwho2Pq6Eg

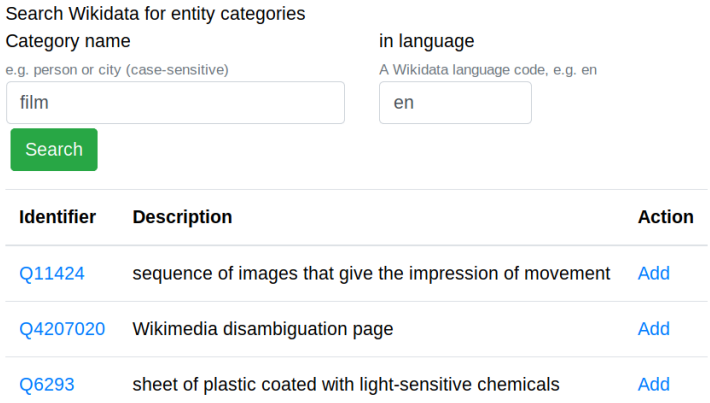


Figure 5.2: Interface in ANEA to search for Wikidata categories from which to extract entity names.

Token	Autom Label	Matches	Other Matches (not picked)
United	B-LOC	United Arab Emirates (en-LOC-Q6256-1)	United (ORG, en-ORG-Q4830453-1)
Arab	I-LOC	United Arab Emirates (en-LOC-Q6256-1)	
Emirates	I-LOC	United Arab Emirates (en-LOC-Q6256-1)	
was	O		

Figure 5.3: Interface in ANEA to manually inspect the automatic labeling.

the longest match is used. I.e. for the string "United Arab Emirates" the entity name of the country is preferred over the substring "United" (the airline) if both are in lists of entities.

5.3.3 *Evaluation*

If a small set of labeled data exists, it can be used to evaluate the automatic annotation. The tool can calculate precision, recall and F1-score directly. It also reports the tokens that were most often labeled incorrectly or not labeled. For a more in-depth analysis, for each token, one can check which label was assigned, which alternative labels could have been assigned and to which entities they correspond (Figure 5.3). These forms of feedback allow a user to understand issues of the automatic annotation . Specific labels can also be changed manually.

5.3.4 *Tuning*

ANEA provides multiple options with which the automatic annotation can be improved. Guided by the evaluation from the previous step, this allows the user to easily insert language expertise into the annotation process and prevent common mistakes while still avoiding

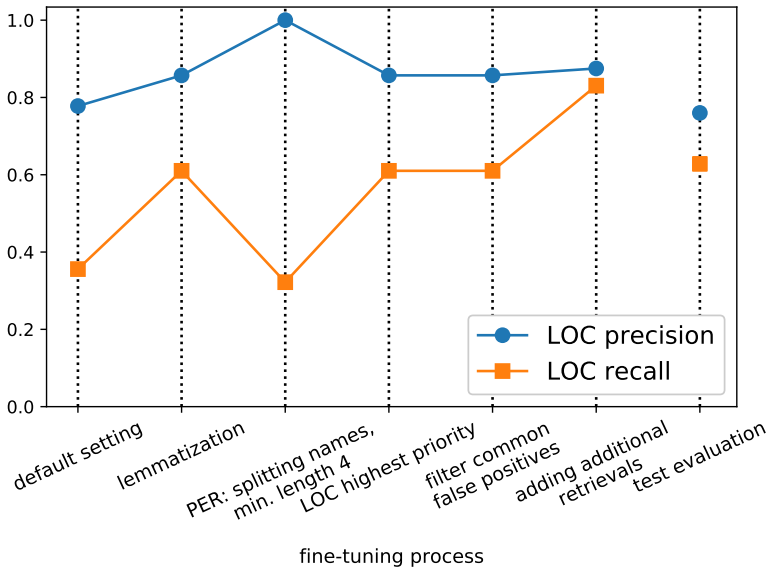


Figure 5.4: Development of precision and recall during the tuning process on the Estonian data. On the x-axis, the setting changes over time are reported.

to annotate or post-edit many tokens manually. The options include lemmatization, filtering common false positives, stopword removal, adding alias names (like "ICLR" for the "International Conference on Learning Representations"), splitting entity names, removing diacritics, requiring a minimum length for the entities, prioritization of lists for resolution of conflicts or fuzzy matching of entities.

The effects of such a tuning process are visualized in Figure 5.4 for an Estonian dataset and the *location* label. Adding lemmatization in tuning-step 1 increases recall due to the language's rich morphological structure that can hinder the matching. In step 3, location entities are given a higher priority if they conflict with person entities on the same token. In the last tuning-step, another gain can be obtained by extracting additional entity lists for Estonian locations based on the evaluation feedback. After the (optional) tuning process, unlabeled text can be automatically annotated for use as distant supervision.

5.4 EXPERIMENTAL EVALUATION

5.4.1 *Datasets*

We selected a variety of datasets that reflect different languages and entity granularities. The first 1500 tokens of each dataset are used as labeled training instances. Garrette and Baldridge (2013) reported this as the number of tokens that can be annotated within two hours for a low-resource POS task. We think that this is a reasonable amount

of labeled data that one can expect even in a low-resource setting, and it is also necessary for training the baselines we compare to. For **English (En)**, the CoNLL03 dataset is probably the most popular NER dataset. It was created for the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003). To obtain a more specialized domain, we manually annotated the *location* labels from the CoNLL03 dataset with more specific labels. For **Spanish (Es)**, we manually annotated news articles with the label *movie* to resemble a Latin-American setting where e.g. a start-up requires a fine-grained and less common label. For **Yorùbá (Yo)**, a language spoken predominantly in West Africa, we evaluate on the dataset by Alabi et al. (2020a). We also evaluate on two European low-resource languages, namely **Estonian (Et)** (Tkachenko et al., 2013) and **West Frisian (Fy)** (Pan et al., 2017). All results are reported on held-out test sets. The manually labeled data created for this evaluation is made publicly available online.

5.4.2 *Machine Learning Models*

We evaluate against two baselines that should, like ANEA, be easy and quick to use, do not require extensive development of hand-engineered features and do not have large hardware requirements. The Stanford NER tagger (Finkel et al., 2005) is a popular tool based on Conditional-Random-Fields (**CRF**) which we use in their suggested configuration.[4] For the second baseline, a neural network (**NN**), we performed preliminary experiments on held-out, English data in a low-resource setting and chose a combination of a bidirectional Gated Recurrent Unit (Cho et al., 2014a) and a ReLU with Dropout (Srivastava et al., 2014) between the layers. To easily apply the model to many different languages, we used pretrained fastText embeddings (Grave et al., 2018a) which are available in 157 languages. Model details are given in the code. In the high-resource setting on the full CoNLL03 dataset (>250k labeled tokens), both baselines achieve an F1 test score of 87.

5.4.3 *Experimental Setup*

Experiment A: Here, the quality of the automatic annotation is evaluated. The CRF is trained on the 1500 labeled training tokens of each dataset. Similarly, for the neural network, the first 1000 tokens are used for the training. The remaining 500 tokens are held-out as the development set to select the best performing epoch and avoid overfitting. For ANEA, we report the scores with and without the tuning phase. *ANEA No Tuning* just uses the default settings without any labeled supervision and no manual interaction. For *ANEA + Tuning*, the 1500 labeled training token are used for the manual tuning. The manual

4 https://nlp.stanford.edu/software/crf-faq.html#a

	CRF			NN			ANEA No Tuning			ANEA + Tuning		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
En PER	**75**	14	23	54	40	46	36	**51**	42	67	49	**57**
En LOC	66	22	33	54	52	52	**70**	45	55	56	**74**	**64**
En ORG	**24**	08	12	23	**13**	**16**	17	07	10	21	09	13
En CITY	**100**	14	25	27	43	33	16	30	21	29	**51**	**37**
En COUN.	**94**	05	10	63	51	56	93	80	86	84	**90**	**87**
En CONTI.	00	00	00	00	00	00	**75**	**94**	**83**	**75**	**94**	**83**
Es MOVIE	**75**	02	05	08	07	08	32	35	33	40	**40**	**40**
Et PER	66	24	35	61	30	40	**75**	17	27	41	**51**	**45**
Et LOC	59	27	37	44	25	32	71	36	48	**76**	**63**	**69**
Et ORG	00	00	00	17	09	12	75	12	21	**81**	**17**	**29**
Fy PER	07	06	07	04	03	04	**55**	**42**	**48**	**55**	**42**	**48**
Fy LOC	32	55	41	33	42	37	**68**	24	37	61	**34**	**43**
Fy ORG	00	00	00	00	00	00	89	07	13	**90**	**08**	**14**
Yo PER	33	05	10	15	22	18	11	13	12	**49**	**43**	**46**
Yo LOC	**100**	07	12	48	27	35	64	72	68	65	**74**	**69**
Yo ORG	00	00	00	07	08	08	16	28	20	**46**	**52**	**49**

Table 5.1: Results of Experiment A comparing the automatic annotation approaches. We report precision/recall/F1-score in percentage (higher is better).

steps were performed by a subject with experience in NLP and fluency in English and Spanish but not in the other languages. Interaction was limited to no more than 10 manual steps and 30 minutes of user interaction per dataset.

Experiment B: For evaluating the effect of the distant supervision, unlabeled tokens are automatically annotated by the CRF, the NN and ANEA with Tuning. The NN model is then retrained on both the manually labeled and the distantly-supervised instances. 200k tokens from each of the datasets are used as unlabeled data. For Spanish, West Frisian and Yorùbá, ca. 15k and 70k and 18k tokens are used, respectively, due to the smaller dataset sizes. These texts are disjoint of the labeled training and test data.

5.4.3.1 *Results*

The results of Experiment A are given in Table 5.1. The CRF approach can provide a high precision but often has a very low recall due to the limited amount of training data. The NN can leverage the pre-training of the embeddings on large amounts of unlabeled text. However, the training data seems not enough to reach a competitive performance. Our tool struggles most with organizations as these are stored as several different entity types in Wikidata. Another issue is the

	NN + Distant Supervision by ...		
	CRF	NN	ANEA
En PER	-35	+5	**+15**
En LOC	-20	+1	**+13**
En ORG	-6	**0**	-5
En CITY	-13	+1	**+6**
En COUN.	-45	-6	**+30**
En CONTI.	0	0	**+88**
Es Movie	-7	+2	**+14**
Et PER	-7	-7	**+14**
Et LOC	+10	-1	**+39**
Et ORG	-2	0	**+17**
Fy PER	+1	0	**+26**
Fy LOC	**+4**	+1	**+4**
Fy ORG	+1	+1	**+7**
Yo PER	-4	**+6**	-5
Yo LOC	-25	+4	**+5**
Yo ORG	-1	+1	**+20**

Table 5.2: Results of Experiment B comparing the use of the automatically annotated data for distant supervision. We report change in precision/recall/F1-score in percentage points compared to the *NN* baseline in Table 5.1 (higher is better).

existence of false positives of words that have other meanings beyond entity names, e.g. the Turkish city "Of". Nevertheless, reasonable results are obtained even if the amount of labeled tokens is too low for the baselines to learn anything meaningful (cf. *En CONTINENT* or *Et ORG*). Even without any labeled data, we are often able to reach competitive performance. Using the tuning process is helpful to boost the performance further. The possibility for the user to trade-off precision and recall can be seen in several cases (e.g. *En LOC* or *Et PER*). Overall, ANEA outperforms the other baselines in all metrics in a majority of the settings. It achieves the best F1-score in all but one case.

The higher quality of the automatic annotation is also reflected in Experiment B (Table 5.2). For 14 out of 16 evaluated entity types, the distant supervision provided by ANEA achieves the largest improvements. On average, it increases the classifier's performance by 18 points F1-score.

5.5 TECHNICAL ASPECTS

The tool consists of both a library for the core functionalities as well as a graphical user interface. The user can control the interface in the browser with the back end running on the local system. Alternatively, the back end can run on a different, more powerful machine and is then accessed remotely. All the code is published as open-source under the Apache 2 license, and we welcome contributions from other authors. The tool is implemented in Python 3 using Flask[5] for the webserver's back end and Bootstrap 4[6] for the front end. To overcome the rate limitations of the Wikidata Web API, a database dump of Wikidata is used. To reduce hardware requirements, care was taken during the implementation to limit the RAM footprint.

The user can upload text files or insert them directly into a text field. For labeled data, the CoNLL column format is supported. Annotated text can be downloaded in the same format. Tokenization and lemmatization are provided for a variety of languages via SpaCy (Honnibal et al., 2020) and EstNLTK (Laur et al., 2020). For other languages, the text can be preprocessed with an external system before inputting it, or the external tool can be easily integrated into ANEA. Stopword lists for 58 languages are included.

5.6 CONCLUSION

We presented an open-source tool to obtain large amounts of distantly supervised training data for NER in a quick way and with few manual efforts and costs. While the annotation itself is automatic, the user can tune it to add their expertise. To support users of varying technical backgrounds, both a library and a graphical user interface are provided. The experiments showed its usefulness in six different language and domain settings.

ANEA has already been used in several projects including (Lange et al., 2019c), (Lange et al., 2019a) and (Adelani et al., 2020). We will leverage the tool to generate the distant supervision in Chapters 6 and 7 to be able to evaluate our noise handling methods on realistic low-resource languages.

5 http://flask.pocoo.org
6 https://getbootstrap.com

6

LOW-RESOURCE TECHNIQUES MEET PRE-TRAINED LANGUAGE MODELS

In the previous chapters, we presented a system to obtain distant supervision in an efficient way and we have seen that weak supervision and noise handling can be effectively used to boost performance. This chapter[1] addresses previous limitations, such as the simulated low-resource scenario. Here, we study two NLP tasks on three African languages. For this aim, we also collected and published new datasets for settings where data was previously lacking.

While the work above used LSTM-based models, in this chapter we focus on the recent multilingual transformer models and evaluate how they can be combined with low-resource techniques. Following the categorization in Chapter 3, we study both an approach to obtain labeled data - the previously introduced weak supervision - as well as a transfer learning method.

6.1 INTRODUCTION

Deep learning techniques, including contextualized word embeddings based on transformers and pretrained on language modelling, have resulted in considerable improvements for many NLP tasks. However, they often require large amounts of labeled training data, and there is also growing evidence that transferring approaches from high to low-resource settings is not straightforward. In (Loubser and Puttkammer, 2020), rule-based or linguistically motivated CRFs still outperform RNN-based methods on several tasks for South African languages. For pretraining approaches where labeled data exists in a high-resource language, and the information is transferred to a low-resource language, Hu et al. (2020b) find a significant gap between performance on English and the cross-lingually transferred models. Concurrent to the first publication of our results, Lauscher et al. (2020b) found that the transfer for multilingual transformer models is less effective for resource-lean settings and distant languages but also that a small amount of target language supervision helps to boost performance. A popular technique to obtain labeled data quickly and cheaply is distant and weak supervision. Kann et al. (2020b) recently inspected POS classifiers trained on weak supervision. They found that in contrast to scenarios with simulated low-resource settings of high-resource languages, in truly low-resource settings this is still a difficult problem.

1 This chapter is based on (Hedderich et al., 2020).

These findings also highlight the importance of aiming for realistic experiments when studying low-resource scenarios.

In this chapter, we analyze multilingual transformer models, namely mBERT (Devlin, 2019; Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020). We evaluate both sequence and token classification tasks in the form of news title topic classification and named entity recognition (NER). A variety of approaches have been proposed to improve performance in low-resource settings. In this work, we study (i) transfer learning from a high-resource language and (ii) weak supervision. We selected these as they are two of the most popular techniques in the recent literature and are rather independent of a specific model architecture. Both need auxiliary data. For transfer learning, this is labeled data in a high-resource language, and for weak supervision, this is expert insight and a mechanism to (semi-)automatically generate labels. We see them, therefore, as orthogonal and depending on the scenario and the data availability, either one or the other approach might be more applicable.

Our study is performed on three linguistically different, African languages: Hausa, isiXhosa and Yorùbá. These represent languages with millions of speakers and active use of digital infrastructure, but with only very limited support for NLP technologies. For this aim, we also collected three new datasets that are made publicly available alongside the code and additional material.[2]

We show both challenges and opportunities when working with multilingual transformer models evaluating trends for different levels of resource scarcity. We start with an overview about the languages we study (Section 6.2) and present the datasets we evaluate on (Section 6.3). The paper is then structured into the following questions we are interested in:

- How do more complex transformer models compare to established RNNs? (Section 6.5)
- How can transfer-learning be used effectively? (Section 6.6)
- Is distant and weak supervision helpful? (Section 6.7)
- What assumptions do we have to consider when targeting a realistic treatment of low-resource scenarios? (Section 6.8)

All experimental details are given at the end of the chapter in Section 6.9.

6.2 LANGUAGES

In this work, we evaluate on three African languages, namely Hausa, isiXhosa and Yorùbá. Hausa is from the Afro-Asiatic family while

2 https://github.com/uds-lsv/transfer-distant-transformer-african

isiXhosa and Yorùbá belong to different branches of the large Niger-Congo family. Hausa and Yorùbá are the second and third most spoken languages in Africa, and isiXhosa is recognized as one of the official languages in South Africa and Zimbabwe.

The Hausa language is native to the northern part of Nigeria and the southern part of the Republic of Niger with more than 45 million native speakers (Eberhard et al., 2019). It is the second most spoken language in Africa after Swahili. Hausa is a tonal language, but this is not marked in written text. The language is written in a modified Latin alphabet.

Yorùbá, on the other hand, is native to south-western Nigeria and the Republic of Benin. It has over 35 million native speakers (Eberhard et al., 2019) and is the third most spoken language in Africa. Yorùbá is a tonal language with three tones: low, middle and high. These tones are represented by the grave ("\"), optional macron ("−") and acute ("/") accents respectively. The tones are represented in written texts along with a modified Latin alphabet.

Lastly, we consider isiXhosa, a Bantu language that is native to South Africa and also recognized as one of the official languages in South Africa and Zimbabwe. It is spoken by over 8 million native speakers (Eberhard et al., 2019). isiXhosa is a tonal language, but the tones are not marked in written text. The text is written with the Latin alphabet.

Kann et al. (2020b) used as an indicator for a low-resource language the availability of data in the Universal Dependency project (Nivre et al., 2020). The languages we study suit their indicator having less than 10k (Yorùbá) or no data (Hausa, isiXhosa) at the time of writing. Yorùbá has been part of the unlabeled training data for the mBERT multilingual, contextual word embeddings. Texts in Hausa and isiXhosa have been part of the XLM-RoBERTa training.

6.3 DATASETS

The three languages have few or no labeled datasets online for popular NLP tasks like named entity recognition (NER) and topic classification. We use the NER dataset by Eiselen (2016) for isiXhosa and the one by Alabi et al. (2020b) for Yorùbá. We collected and manually annotated a NER dataset for Hausa and news title topic classification datasets for Hausa and Yorùbá. Table 6.1 gives a summary of the datasets.

6.3.1 *Existing Datasets*

The WikiAnn corpus (Pan et al., 2017) provides NER datasets for 282 languages available on Wikipedia. These are, however, only silver-standard annotations and for Hausa and isiXhosa less than 4k and 1k tokens respectively are provided. The LORELEI project announced

Dataset Name	Data Source	Train/Val/Test Sentences
Hausa NER*	VOA Hausa	1,014 / 145 / 291
Hausa Topic Classification*	VOA Hausa	2,045 / 290 /582
isiXhosa NER (Eiselen, 2016)	SADiLaR	5,138 / 608 / 537
Yorùbá NER (Alabi et al., 2020b)	GlobalVoices	816 / 116 / 236
Yorùbá Topic Classification*	BBC Yoruba	1,340 / 189 / 379

Table 6.1: Datasets Summary. *Created for this work.

the release of NER datasets for several African languages via LDC (Strassel and Tracey, 2016; Tracey et al., 2019) but have not yet done so for Hausa and Yorùbá at the time of writing.

Eiselen and Puttkammer (2014) and Eiselen (2016) created NLP datasets for South African languages. We use the latter's NER dataset for isiXhosa. For the Yorùbá NER dataset (Alabi et al., 2020b), we use the authors' split into training, dev and test set of the cased version of their data.[3] For the isiXhosa dataset,[4] we use an 80%/10%/10% split following the instructions in (Loubser and Puttkammer, 2020). The split is based on token-count, splitting only after the end of the sentence (information obtained through personal conversation with the authors). For the fine-tuning of the zero- and few-shot models, the standard CoNLL03 NER (Tjong Kim Sang and De Meulder, 2003) and AG News (Zhang et al., 2015) datasets are used with their existing splits.

6.3.2 *New Datasets*

6.3.2.1 *Hausa NER*

For the Hausa NER annotation, we collected 250 articles from VOA Hausa,[5] 50 articles each from the five pre-defined categories of the news website. The categories are Najeriya (Nigeria), Afirka (Africa), Amurka (USA), Sauran Duniya (the rest of the world) and Kiwon Lafiya (Health). We removed articles with less than 50 tokens which results in 188 news articles (over 37K tokens). We asked two volunteers who are native Hausa speakers to annotate the corpus separately. Each volunteer was supervised by someone with experience in NER annotation. Following the named entity annotation in Yorùbá by Alabi et al. (2020b), we annotated PER, ORG, LOC and DATE (dates and times) for Hausa. The annotation was based on the MUC-6 Named

3 https://github.com/ajesujoba/YorubaTwi-Embedding/tree/master/Yoruba/Yor%C3%B9b%C3%A1-NER
4 https://repo.sadilar.org/handle/20.500.12185/312
5 https://www.voahausa.com

Entity Task Definition guide.[6] Comparing the annotations of the volunteers, we observed a conflict for 1302 tokens (out of 4838 tokens) excluding the non-entity words (i.e. words with 'O' labels). One of the annotators was better in annotating DATE, while the other was better in annotating ORG especially for multi-word expressions of entities. We resolved all the conflicts after discussion with one of the volunteers. The split of annotated data of the Yoruba and Hausa NER data is 70%/10%/20% for training, validation and test sentences.

6.3.2.2 *Hausa and Yorùbá Text classification*

For the topic classification datasets, news titles were collected from VOA Hausa and the BBC Yoruba news website.[7] Two native speakers of the language annotated each dataset. We categorized the Yorùbá news headlines into 7 categories, namely "Nigeria", "Africa", "World", "Entertainment", "Health", "Sport", "Politics". Similarly, we annotated 5 (of the 7) categories for Hausa news headlines, excluding "Sport" and "Entertainment" as there was only a limited number of examples. The "Politics" category in the annotation is only for Nigerian political news headlines. Comparing the two annotators, there was a conflict rate of 7.5% for Hausa and 5.8% for Yorùbá. The total number of news titles after resolving conflicts was 2,917 for Hausa and 1,908 for Yorùbá.

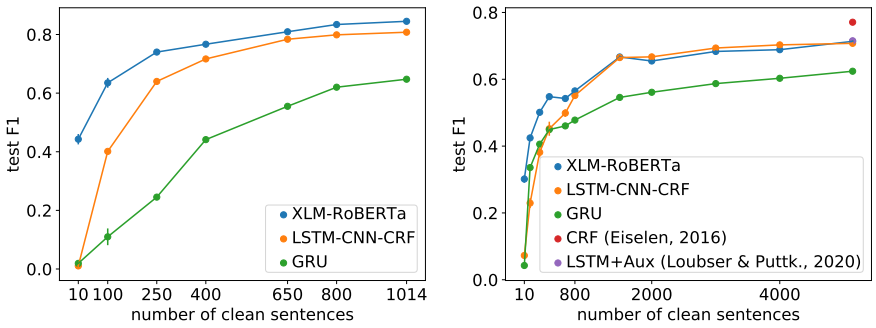
6.4 EXPERIMENTS

To evaluate different amounts of resource-availability, we use subsets of the training data with increasing sizes from ten to the maximally available number of sentences. All the models are trained on their corresponding language-model pretraining. Except if specified otherwise, the models are not fine-tuned on any other task-specific, labeled data from other languages. We report mean F1-score on the test sets over ten repetitions with standard error on the error bars. Additional experimental details are given in the following sections and at the end of the chapter. We made the code as well as a table with the scores for all the runs available online.

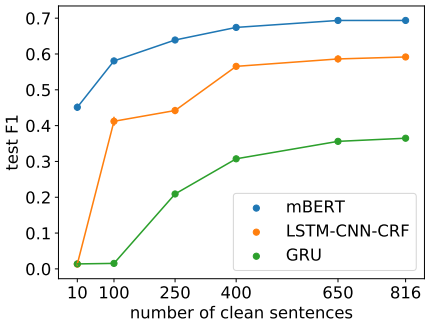
6.5 COMPARING TO RNNS

Loubser and Puttkammer (2020) showed that models with comparatively few parameters, like CRFs, can still outperform more complex, neural RNNs models for several task and low-resource language combinations. This motivates the question whether model complexity is an issue for these low-resource NLP models. We compare to simple

6 https://cs.nyu.edu/faculty/grishman/NEtask20.book_1.html
7 https://www.bbc.com/yoruba



(a) Hausa (b) isiXhosa



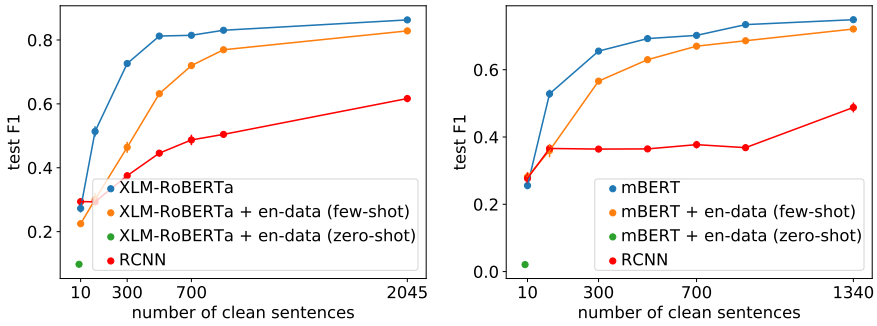
(c) Yorùbá

Figure 6.1: Comparing to RNNs on NER datasets.

GRU based (Cho et al., 2014b) models as well as the popular (non-transformer) combination of LSTM-CNN-CRF (Ma and Hovy, 2016a) for NER and to the RCNN architecture (Lai et al., 2015) for topic classification. For these models, we use pre-trained, non-contextual word embeddings trained for the specific language. Figures 6.1 and 6.2 show that an increase in model complexity is not an issue in these experiments. For Hausa and Yorùbá and for the low resource settings for isiXhosa, BERT and XLM-RoBERTa actually outperform the other baselines, possibly due to the larger amounts of background knowledge through the language model pre-training. For larger amounts of task-specific training data, the LSTM-CNN-CRF and the transformer models perform similarly. One should note that for isiXhosa, the linguistically motivated CRF (Eiselen, 2016) still outperforms all approaches on the full dataset.

6.6 TRANSFER LEARNING

The mBERT and XLM-RoBERTa models are trained with tasks that can be obtained from unlabeled text, like masked language modelling. Additionally, the multilingual models can be fine-tuned on task-specific, supervised data but from a different, high-resource language. There is evidence that the multilingual transformer models can learn parallel concepts across languages (Hu et al., 2020b; Pires et al., 2019; Wu



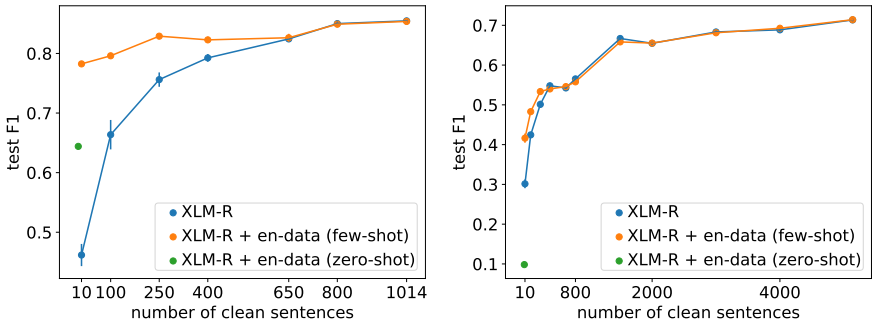
(a) Hausa (b) Yorùbá

Figure 6.2: Comparing to RCNN and using transfer learning on topic classification datasets.

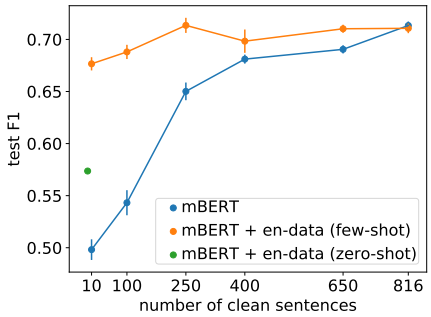
and Dredze, 2019). This allows to then apply or evaluate the model directly without having been fine-tuned on any labeled data in the target language (zero-shot) or on only a small amount of labeled data in the target language (few-shot).

For NER, we pre-train on the English CoNLL03 NER dataset (Tjong Kim Sang and De Meulder, 2003). For topic classification, the models are pre-trained on the English AG News corpus (Zhang et al., 2015). The texts in the high-resource English and the low-resource Hausa and Yorùbá target datasets share the same domain (news texts). One issue that is visible in these experiments is the discrepancy between classes. While some classes like "Politics" are shared, the topic classification datasets also have language- and location-specific classes like "Nigeria" and "Africa" which are not part of the English fine-tuning dataset. In our experiments, we use the intersection of labels for NER (excluding DATE and MISC for Hausa and Yorùbá) and the union of labels for topic classification.

The results for NER in Figure 6.3 confirm the benefits of fine-tuning on high-resource languages already shown in past research. They show, however, also the large gains in performance that can be obtained by training on a minimal number of target instances. While the zero-shot setting in (Hu et al., 2020b) is interesting from a methodological perspective, using a small training set for the target language seems much more beneficial for a practical application. In our experiments, we get - with only ten labeled sentences - an improvement of at least 10 points in the F1-score for a shared label set on NER. The difference between training with and without transfer learning disappear once enough target language training data is available - in our settings with around 600 to 800 labeled sentences. For topic classification (Figure 6.2), the transfer learning is not beneficial, which might be due to the mismatch in the label sets. Taking the label space into account, as done e.g. by Halder et al. (2020) for single-language data, might help the transfer to unseen, cross-lingual labels.



(a) Hausa (b) isiXhosa



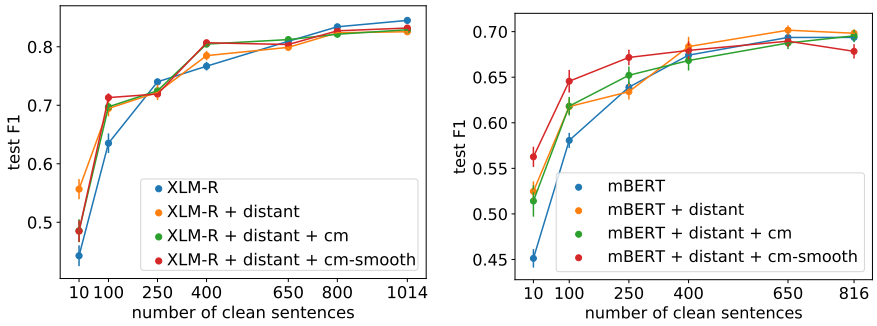
(c) Yorùbá

Figure 6.3: Using transfer learning for NER.

6.7 WEAK SUPERVISION

Distant and weak supervision are popular techniques when labeled data is lacking. It allows a domain expert to insert their knowledge without having to label every instance manually. This kind of (semi-) automatic supervision tends to contain more errors which can hurt the performance of classifiers (see e.g. Fang and Cohn (2016)). To avoid this, it can be combined with label noise handling techniques. We introduced this pipeline in Chapter 4 and similar approaches have also been successfully applied for several other NLP tasks (see Chapter 3), however, mostly for RNN based architectures. In Section 6.5 we saw that the RNN models have a lower baseline performance. We are, therefore, interested in whether weak supervision is still useful for the better performing transformer models. Several of the past works evaluated their approach only on high-resource languages or simulated low-resource scenarios. We are, thus, also interested in how the weak supervision performs for the actual resource-lean African languages we study.

To create the weak supervision, native speakers with a background in NLP were asked to write labeling rules. For the NER labels PER, ORG and LOC, we match the tokens against lists of entity names. These were extracted from the corresponding categories from Wikidata. For the DATE label, the insight is used that date expressions are



(a) Hausa (b) Yorùbá

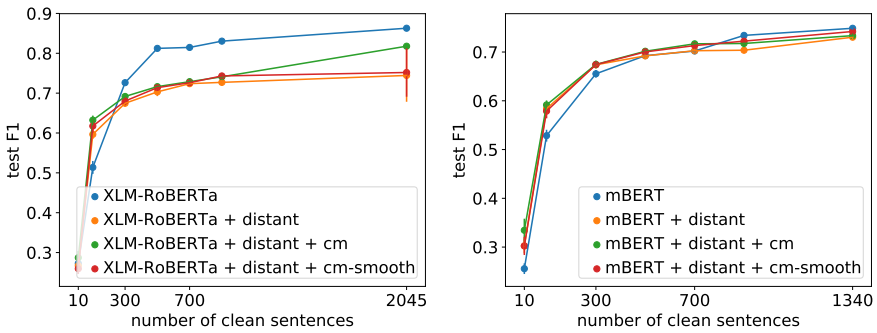
Figure 6.4: Using weak and distant supervision for NER.

usually preceded by date keywords in Yorùbá, as reported by Adelani et al. (2020). We find similar patterns in Hausa like "*ranar*"(day), "*watan*" (month), and "*shekarar*"(year). For example, *"18th of May, 2019"* in Hausa translates to *"ranar 18 ga watan Mayu, shekarar 2019"*. The annotation rules are based on these keywords and further heuristics. Directly applying this weak supervision on the NER test sets results in an F1-score of 54% and 62% on Hausa and Yorùbá, respectively.

For the topic classification task, the weak supervision rules are based on a dictionary of words relating to each of the classes. To induce the dictionaries, we collected terms related to different classes from web sources. For example, for the "Sport" label, names of sportspeople and sport-related organizations were collected and similarly for the "Africa" label, names of countries, their capitals and major cities and their politicians. To label a news headline, the intersection between each class-dictionary and the text was computed, and a class was selected with a majority voting scheme. We obtain an F1-score of 49% and 55% on the Hausa and Yorùbá test set respectively when applying the weak supervision directly to the topic classification test sets. Additional details on the weak supervision are given in Section 6.9.

For label noise handling, we use the confusion matrix approach for NER introduced in Chapter 4, marked as *cm* in the plots. Additionally, we propose to combine it with the smoothing concept by Lv et al. (2020), marked as *cm-smooth*.

The Figures 6.4 and 6.5 show that when only a small amount of manually labeled data is available, weak supervision can be a helpful addition. E.g. for the NER task in Yorùbá, combining weak supervision and noise handling with 100 labeled sentences achieves similar performance to using 400 manually labeled sentences. For label noise handling, combining the confusion matrix with the smoothing approach might be beneficial because the estimated confusion matrix is flawed when only small amounts of labeled data are given. When more manually labeled data is available, the noisy annotations lose



(a) Hausa (b) Yorùbá

Figure 6.5: Using weak and distant supervision for topic classification.

their benefit and can become harmful to performance. Improved noise-handling techniques might be able to mitigate this.

6.8 QUESTIONING ASSUMPTIONS

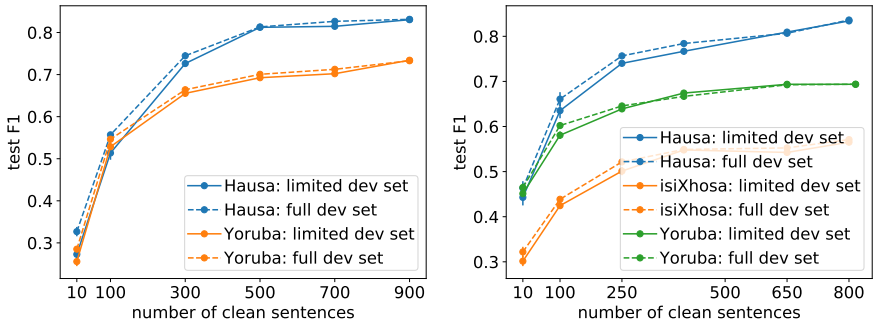
In this section, we want to discuss certain assumptions taken by us and previous work in low-resource learning to see if these hold and what challenges and opportunities they could bring for future work.

6.8.1 *Development Set*

Kann et al. (2019) criticized that research on low-resource often assumes the existence of a development set. Addressing this, we perform hyperparameter optimization on high-resource English data. For early-stopping (to avoid overfitting), Kann and her colleagues experiment with obtaining an early-stop-epoch from the average of several other languages. To avoid this multi-language set-up and the need to obtain labeled data for multiple languages, we suggest using instead a development set downsized by the same factor as the training data. This approach keeps the ratio between training and development set giving the development set a reasonable size to obtain in a low-resource setting. For the setting with ten labeled sentences for training, also ten sentences are used for the dev set. The results in Figure 6.6 show that a similar performance can be reached with a limited development set compared to the full development set. All other experiments in this chapter use, therefore, the limited development set to reduce the assumed availability of labeled data.

6.8.2 *Hardware Resources*

While the multilingual transformer models show impressive improvements over the RNN baselines, they also require more hardware



(a) Topic Classification (b) NER

Figure 6.6: Studying the use of the development set for Hausa and Yorùbá.

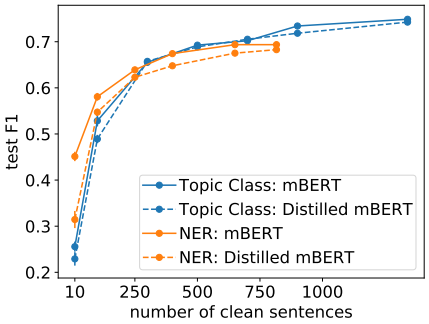


Figure 6.7: Comparing mBERT to its distilled version for Yorùbá.

resources. The LSTM-CNN-CRF model, e.g. has ca. 5M parameters compared to mBERT's over 150M parameters. The computing capabilities for training and deploying such models might not always be given in low-resource scenarios. Through personal conversations with researchers from African countries, we found that this can be an issue. There are approaches to reduce model size while keeping a similar performance quality, e.g. the 25% smaller DistilBERT (Sanh et al., 2019). Figure 6.7 shows that this performs indeed similar in many cases but that there is a significant drop in performance for NER when only few training sentences are available.

6.8.3 *Annotation Time*

In (Hu et al., 2020b) and (Kann et al., 2020b), it is assumed that no labeled training data is available for the target language. In the previous sections, we showed that even with ten labeled target sentences, reasonable model quality can be achieved. For our annotation efforts, we measured on average 1 minute per annotator per sentence for NER and 6 seconds per sentence for topic classification. We, therefore, think that it is reasonable to assume the availability of small amounts of labeled data. Especially, as we would argue that it is beneficial to

have a native speaker or language expert involved when developing a model for a specific language.

For weak supervision, these annotation times pose a challenge. Extracting named entities from knowledge bases requires minimal manual effort assuming a set-up system (as introduced in Chapter 5) and manual crafting rules took 30 minutes for the DATE label. Given our annotators' speed, they would be able to label 30 sentences in the same time. This makes the weak supervision an interesting competitor to manual annotation. In contrast, 2.5 hours were needed to create the rules for each topic classification dataset. In that time, 1500 sentences could be manually labeled. There, manual annotation easily outperforms weak supervision, if annotation time is the only criterium. When reporting results for weak supervision, the performance benefits should therefore also be compared against manual annotation in the same time frame.

6.9 EXPERIMENTAL DETAILS

In this section, we give additional details about the experimental set-up, the weak and distant supervision and the model hyperparameters.

6.9.1 *Word Embeddings*

For the RNN models, we make use of word features obtained from Word2Vec embeddings for the Hausa language and FastText embeddings for Yorùbá and isiXhosa languages. We utilize the better quality embeddings recently released by Abdulmumin and Galadanci (2019) and Alabi et al. (2020b) for Hausa and Yorùbá respectively instead of the pre-trained embeddings by Facebook that were trained on a smaller and lower quality dataset from Wikipedia. For isiXhosa, we did not find any existing word embeddings, therefore, we trained FastText embeddings from data collected from the I'solezwe news website[8] and the *OPUS* corpus (Tiedemann, 2012b). The corpus size for isiXhosa is 1.4M sentences (around 15M tokens). We trained FastText embeddings for isiXhosa using *Gensim*[9] with the following hyper-parameters: embedding size of 300, context window size of 5, minimum word count of 3, number of negative samples ten and number of iterations 10.

6.9.2 *Weak and Distant Supervision*

6.9.2.1 *Distant Supervision for Person Names, Organisations and Locations*

We make use of lists of entities to annotate PER, ORG and LOC automatically. We use ANEA (see Chapter 5) to extract personal names,

8 https://www.isolezwelesixhosa.co.za/
9 https://radimrehurek.com/gensim/

organizations and locations from Wikidata as entity lists and assign a corresponding named entity label if tokens from an unlabeled text match an entry in an entity list. The option to tune the automatic annotation is used to improve matching. For Yorùbá, a minimum token length of 3 was set for extraction of LOC and PER, while the minimum length for ORG was set to 2. This reduces the false positive rate, e.g. preventing matches with function words like "of".

Applying this on the test set, we obtained a precision of 80%, 38% and 28% for LOC, ORG and PER respectively; a recall of 73%, 52% and 14% for LOC, ORG and PER respectively; and an F1-score of 76%, 44% and 19% for LOC, ORG and PER respectively.

For Hausa NER, a minimum token length of 4 was set for extraction of LOC, ORG and PER. Based on these manual heuristics, on the test set, we obtained a precision of 67%, 12% and 47% for LOC, ORG and PER respectively; a recall of 63%, 37% and 56% for LOC, ORG and PER respectively; and an F1-score of 65%, 18% and 51% for LOC, ORG and PER respectively.

6.9.2.2 *DATE Rules for NER*

Rules allow us to apply the knowledge of domain experts without the manual effort of labeling each instance. We asked native speakers with knowledge of NLP to write DATE rules for Hausa and Yorùbá. In both languages, date expressions are preceded by date keywords, like "*ranar*" / "*ọjọ́*" (day), "*watan*" / "*oṣù*" (month), and "*shekarar*" / "*ọdún*" (year) in Hausa/Yorùbá. For example, *"18th of December, 2019"* in Hausa / Yorùbá translates to " *ranar 18 ga watan Disamba, shekarar 2019*" / "*ọjọ́ 18 oṣù Ọpẹ̀, ọdún 2019*". The annotation rules are based on the following three criteria: (1) A token is a date keyword or follows a date keyword in a sequence. (2) A token is a digit, and (3) other heuristics to capture relative dates and date periods connected by conjunctions e.g between July 2019 and March 2020. Applying these rules results in a precision of 49.30%/51.35%, a recall of 60.61%/79.17% and an F1-score of 54.42%/62.30% on the Hausa/Yorùbá test sets respectively.

6.9.2.3 *Rules for Topic Classification*

For the Yorùbá topic classification task, we collected terms that correspond to the different classes into sets. For example, the set for the class Nigeria contains names of agencies and organizations, states and cities in Nigeria. The set for the World class is made up of the name of countries of the world, their capitals and major cities and world affairs related organizations. Names of sporting clubs and sportspeople across the world were used for the Sports class and list of artists and actresses and entertainment-related terms for the Entertainment class. Given a news headline to be annotated, we get the union set of 1- and 2-grams and obtain the intersection with the class dictionaries we have.

The class with the highest number of intersecting elements is selected. In the case of a tie, we randomly pick a class out the classes with a tie. Just as we did for Yorùbá, we collected the class-related tokens for the Hausa text classification. We, however, split the classification into two steps, checking some important tokens first and then using the same approach as we used for Yorùbá. If a headline contains the word *cutar* (disease), it is classified as Health, if it contains tokens such as *inec*, *zaben*, *pdp* or *apc* (which are all politics related tokens) it is classified as Politics. Furthermore, sentences with any of the tokens *buhari*, *legas*, *kano*, *kaduna* or *sokoto* are classified as Nigeria while sentences with *afurka*, *kamaru* or *nijar* are classified as Africa. If none of the tokens highlighted above is found, we apply the same approach as we did for the Yorùbá setting, which is a majority voting of the intersection set of the news headline with a keyword set for each class. Applying these rules results in a precision of 59.54%/60.05%, a recall of 46.04%/53.66% and an F1-score of 48.52%/54.93% on the Hausa /Yorùbá test set respectively.

6.9.3 *Model Settings*

6.9.3.1 *General*

All experiments were repeated ten times with varying random seeds but with the same data (subsets). We report mean F1 test score and standard error ($\sigma/\sqrt{10}$). For NER, the score was computed following the standard CoNLL approach (Tjong Kim Sang and De Meulder, 2003) using the *seqeval* implementation.[10] Labels were in the BIO2-scheme. For evaluating topic classification, the implementation by *scikit-learn* was used.[11] All models were trained for 50 epochs, and the epoch that performed best on the (possibly size-reduced) development set was used for evaluation.

6.9.3.2 *BERT and XLM-RoBERTa*

As multilingual transformer models, mBert and XLM-RoBERTa were used, both in the implementation by Wolf et al. (2019). The specific model IDs were *bert-base-multilingual-cased* and *xlm-roberta-base*.[12] For the DistilBERT experiment it was *distilbert-base-multilingual-cased*. As is standard, the last layer (language model head) was replaced with a classification layer (either for sequence or token classification). Models were trained with the Adam optimizer and a learning rate of $5e^{-5}$. Gradient clipping of value 1 was applied. The batch size was 32 for NER and 128 for topic classification. For weak supervision and XLM-

10 https://github.com/chakki-works/seqeval
11 https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html
12 https://huggingface.co/transformers/pretrained_models.html

RoBERTa on the Hausa topic classification data with 100 or more labeled sentences, we observed convergence issues where the trained model would just predict the majority classes. We, therefore, excluded for this task runs where *on the development set* the class-specific F1 score was 0.0 for two or more classes. The experiments were then repeated with a different seed.

6.9.3.3 *Other Architectures*

For the GRU and LSTM-CNN-CRF model, we used the implementation by Chernodub et al. (2019) with modifications to support FastText embeddings and the *seqeval* evaluation library. Both model architectures were bidirectional. Dropout of 0.5 was applied. The batch-size was 10 and SGD with a learning rate of 0.01, and a decay of 0.05 and momentum of 0.9 was used. Gradients were clipped with a value of 5. The RNN dimension was 300. For the CNN, the character embedding dimension was 25 with 30 filters and a window-size of 3.

For the topic classification task, we experimented with the RCNN model proposed by Lai et al. (2015). The hidden size in the Bi-LSTM was 100 for each direction. The linear layer after the Bi-LSTM reduced the dimension to 64. The model was trained for 50 epochs.

6.9.3.4 *Transfer Learning*

For transfer learning, the model was first fine-tuned on labeled data from a high-resource language. Following Hu et al. (2020b), we used the English CoNLL03 NER dataset (Tjong Kim Sang and De Meulder, 2003) for NER. It consists of ca. 8k training sentences. The model was trained for 50 epochs and the weights of the best epoch according to the development set were taken. The training parameters were the same as before. On the English CoNLL03 test set, the model achieved a performance of 0.90 F1-score. As the Hausa and Yorùbá datasets have slightly different label sets, we only used their intersection, resulting in the labels PER, LOC and ORG and excluding MISC from CoNLL03 and the DATE label from Hausa/Yorùbá. For isiXhosa, the label sets were identical (i.e. also including MISC). After fine-tuning the model on the high-resource data, the model was directly evaluated on the African test set (for zero-shot) or fine-tuned and then evaluated on the African data (for few-shot).

For topic classification, the AG News corpus was used (Zhang et al., 2015). It consists of 120k training sentences. The model was trained for 20 epochs and the weights of the best epoch according to the test set were used. On this set, an F1-score of 0.93 was achieved. The training procedure was the same as above. For the labels, we used the union of the labels of the AG News corpus (Sports, World, Business and Sci/Tech) and the African datasets.

6.9.3.5 *Label Noise Handling*

We used the specific approach presented in Chapter 4 that was developed to work with small amounts of manually labeled, clean data and a large amount of automatically annotated, noisy labels obtained through weak supervision. To get the confusion matrix of the noise, the weak supervision was applied on the small set of clean training instances. From the resulting pairs of clean and noisy labels, the confusion matrix was estimated.

In a setting where only a few instances were available, the estimated confusion matrix might not be close to the actual noise distribution. We, therefore, combined it with the smoothing approach by Lv et al. (2020). Each entry of the probabilistic confusion matrix was raised to the power of β and then row-wise normalized.

Following the insights from Chapter 4, we did not use the full amount of available, weakly supervised instances in each epoch. Instead, in each epoch, only a randomly selected subset of the size of the clean, manually labeled training data was used to lessen the negative effects of the noisy labels additionally. For smoothing, $\beta = 0.8$ was used as this performed best for Lv et al. (2020).

6.10 CONCLUSIONS

In this chapter, we evaluated transfer learning and weak supervision on multilingual transformer models, studying realistic low-resource settings for African languages. We showed that even with a small amount of labeled data, reasonable performance can be achieved. With few-shot transfer learning, just 10 labeled target language sentences are needed to reach a similar performance as a model trained on several hundred labeled sentences of only the target language. While transfer learning performed well for NER, the study also showed that the method struggled on the topic classification datasets, possibly due to the label mismatch. Here, weak supervision could shine with its higher flexibility for language specific labels. With our new datasets, we hope that we can foster future research in this area.

Realistic evaluations are important to ensure that methods are applicable in the real world. In the second part of the analysis, we reflected on assumptions taken by us and previous low-resource work and how realistic they are. Assuming access to a large development set seems unrealistic in a low-resource scenario. We showed that, instead, a development set of limited size can be an alternative obtaining similar test performance. Another assumption is the availability of expensive hardware such as GPUs with large memory. As this might not be the case in low-resource scenarios, distilled versions of these models could be an option to reduce the memory requirements. While these models achieve similar performance as their original counterparts in the higher resourced settings, we saw that the performance difference

is more noticeable in some of the low-resource cases. Last but not least, we argue that one should compare to the strong baseline of spending the same time manually labeling data as is needed to set-up the weak supervision by the domain expert. This provides a challenge for weak supervision as well as motivating the development of better and faster weak supervision systems in the future.

Above, we saw the benefit that label noise handling can bring to improve weak supervision. In the next chapter, we will take a closer look at noise modeling and identify from a theoretic perspective the factors that influence the estimation of label noise models.

7 NOISE MODEL ESTIMATION ON REALISTIC NOISE

In Chapters 4 and 6, we showed that label noise modeling can be used to better leverage weakly labeled data in applied low-resource NLP settings. In this chapter,[1] we will revisit label noise modeling from a more task-independent and theoretic perspective. We identify the factors that influence the estimation of the noise model. Bridging back to the applied evaluation, we also compare the types of synthetic noise often assumed in theoretic work to realistic noise that might be found "in the wild."

7.1 INTRODUCTION

One of the factors in the success of deep neural networks is the availability of large, labeled datasets. Where such labeled data is not available, weak and distant supervision have become popular. Related but different to semi-supervised learning, in distant supervision, the unlabeled data is automatically annotated by a separate process using e.g. rules and heuristics created by expert (Ratner et al., 2016) or exploiting the context of images (Mahajan et al., 2018). For information extraction from text (Mintz et al., 2009), this has become one of the dominant techniques to overcome the lack of labeled data.

While distant supervision allows generating labels in a cheap and fast way, these labels tend to contain more errors than gold standard ones. We have seen in previous chapters that training with this additional data might even deteriorate the performance of a classifier.

Effectively leveraging this noisily-labeled data for machine learning algorithms has become a very active field of research. One of the major approaches is the explicit modeling of the noise. This general concept is task-independent and can be added to existing deep learning architectures. It is visualized in Figure 7.1. The *base model* is the model that was originally developed for a specific classification task. It is directly used during testing and when dealing with other clean data. When working with noisily-labeled data during training, a *noise model* is added after the base model's output. The noise model is an estimate of the underlying noise process. The training process of the base model can benefit from it as the noise model can be seen as changing the distribution of the labels from the clean to the noisy. This will be properly defined below.

Many works on noise modeling assume that no manually annotated, clean data is available. In Chapter 6, we have shown, however, that it

1 This chapter is based on (Hedderich et al., 2021c).

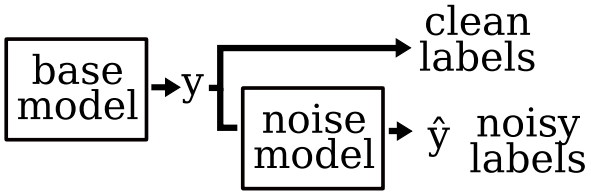


Figure 7.1: Visualization of the general noise model architecture. The *base model* works directly on the clean data and predicts the clean label y. For noisily-labeled data, a *noise model* is added after the base model's predictions.

is both realistic and beneficial to assume a small amount of manually labeled instances. This motivates us to study in this work scenarios where a small amount of clean, gold-standard data, as well as a large amount of noisily labeled data, are available.

In this chapter, we focus on the quality of the noise model. The better such a model is estimated, the better it can model the relationship between clean and noisy labels. We are interested in the factors on which the quality of the noise model depends. We show, both theoretically and empirically, the influence of the clean data and the noise properties. We also propose to adapt the sampling technique to obtain more accurate estimates. Apart from helping to improve the understanding of these noise models in general, we hope that the insights can also be useful guidance for practitioners.

In the noisy labels literature, theoretical insights are often evaluated only on simulated noise processes, e.g. on MNIST with added single-label-flip noise (e.g. in Reed et al. (2015), Goldberger and Ben-Reuven (2016), Bekker and Goldberger (2016), Patrini et al. (2017) and Han et al. (2018)). This synthetic noise has the advantage that it can be controlled completely and allows to rigorously and continuously evaluate aspects like the noise level. However, certain assumptions about the noise have to be taken. And these are usually the same assumptions that are chosen for the noise model itself. It might, therefore, not be too surprising that such a model is suited for such a noise.

Recently, efforts have been taken to also evaluate on more realistic scenarios, mostly in the vision domain, e.g. the Clothing1M dataset by Xiao et al. (2015). We want to add to this by making available a noisy label dataset from the natural language processing (NLP) domain based on an existing named entity recognition (NER) corpus. It provides parallel clean and noisy labels for the full data allowing to evaluate different scenarios of resource availability of both clean and noisy data. This new dataset also contains properties that can make learning with noisy labels more challenging such as skewed label distributions and a noise level higher than the true label probability in some settings. In contrast to existing work, we provide seven different sets of noisy labels, each obtained by a realistic noise process via different heuristics in the distant supervision. This makes it possible

to experiment with different noise levels for the same instances. The dataset along with the code for the experiments is made publicly available.[2]

Our key contributions:

- A derivation of the expected error of the noise model estimated from pairs of clean and noisy labels with empirical verification of the derived results on both simulated and realistic noisy labels.
- A set of experiments analyzing how the noise model estimation influences the test performance of the base model.
- NoisyNER, an NLP dataset with noisy labels obtained through non-synthetic, realistic distant supervision that also provides different levels of noise and parallel clean labels.

7.2 BACKGROUND

We are given a dataset D consisting of instances $(x, \hat{y})$ where $\hat{y}$ is a noisy label that can differ from the unknown, clean/true label y. Both clean and noisy label have one of k possible label values/classes. We assume that the change from the clean to the noisy label was performed by a probabilistic noise process described as $p(\hat{y} = j|y = i)$. This describes the probability of a true label y being changed from value i to the noisy label $\hat{y}$ with label value j. With probability $p(\hat{y} = j|y = j)$ the label value is kept unchanged. This is a common approach to describe noisy label settings. Under this process, a uniform noise (Larsen et al., 1998) with noise level ϵ is obtained with

$$p_{\text{uni}}(\hat{y} = j|y = i) = \begin{cases} 1 - \epsilon, & \text{for } i = j \\ \frac{\epsilon}{k-1}, & \text{for } i \neq j \end{cases}, \tag{7.1}$$

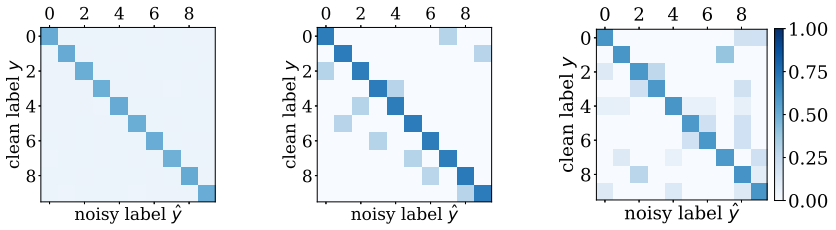
and a single label-flip noise (Reed et al., 2015) via

$$p_{\text{flip}}(\hat{y} = j|y = i) = \begin{cases} 1 - \epsilon, & \text{for } i = j \\ \epsilon, & \text{for one } i \neq j \\ 0, & else \end{cases}. \tag{7.2}$$

In this work, we will use the more general form that just requires that a noise process can be described with a valid noise transition probability $p(\hat{y} = j|y = i)$ (Bekker and Goldberger, 2016), i.e.

$$\sum_{j=1}^{k} p(\hat{y} = j|y = i) = 1 \text{ and } p(\hat{y} = j|y = i) \geq 0 \; \forall i, j. \tag{7.3}$$

2 https://github.com/uds-lsv/noise-estimation



(a) uniform (b) single-flip (c) multi-flip

Figure 7.2: Different noise processes visualized as noise matrices M: uniform noise ($\epsilon = 0.5$), single-flip noise ($\epsilon = 0.3$) and multi-flip noise ($\epsilon = 0.4$).

This allows to model more complex noise processes where a true label can be confused with multiple other labels at different noise rates/probabilities. We call this multi-flip noise. This definition generalizes to multi-class classification what in the binary case Natarajan et al. (2013) and Scott et al. (2013) describe as class-conditional and asymmetric noise. It is called Markov label corruption by Rooyen and Williamson (2017).

The noise process can also be described as a matrix $M \in \mathbb{R}^{k \times k}$ where

$$M_{i,j} = p(\hat{y} = j | y = i) \; . \tag{7.4}$$

The matrix M is called confusion or noise transition matrix. See Figure 7.2 for example matrices with uniform, single-flip and a more complex multi-flip noise.

For a multi-class classification setting, this noise process results in the following relation:

$$p(\hat{y} = j | x) = \sum_{i=1}^{k} p(\hat{y} = j | y = i) p(y = i | x) \; . \tag{7.5}$$

This is used to adapt the predictions from the clean label distribution $p(y = i|x)$ (learned by the base model) to the noisy label distribution $p(\hat{y} = j|x)$ for the noisily labeled data via the noise process or model $p(\hat{y} = j|y = i)$.

Several ways have been proposed to estimate the noise model when only noisy data is available. This includes the use of expert insight (Mnih and Hinton, 2012), EM-algorithm (Bekker and Goldberger, 2016; Chen et al., 2019a; Paul et al., 2019), backpropagation with regularizers (Luo et al., 2017; Sukhbaatar et al., 2015) and the estimates of a pretrained neural network (Dgani et al., 2018; Goldberger and Ben-Reuven, 2016; Patrini et al., 2017; Wang et al., 2019). In Chapters 4 and 6, we studied a setting where a small portion of clean data can be used. Noise model estimation methods have also been proposed for such a scenario by Fang and Cohn (2016), Hendrycks et al. (2018), and Xiao et al. (2015) and Lange et al. (2019c).

Out of the different approaches to estimate a noise model given a small amount of clean labels, we will follow the specific definitions from Chapter 4 which are based on (Goldberger and Ben-Reuven, 2016) and are also used in (Lange et al., 2019c). We assume that a small dataset D_C with clean labels exists, i.e. $(x, y) \in D_C$ is known. The size of this clean dataset is much smaller than that of the noisy dataset D. We then relabel the instances in D_C with the same mechanism that was used for D (e.g. distant supervision) to obtain $\hat{y}$. This results in a set S_{NC} of pairs $(y, \hat{y})$ with $|S_{NC}| = |D_C|$. The noise matrix M is then estimated as $\widetilde{M}$ using the transitions from y to $\hat{y}$ (or equivalently the confusion between y and $\hat{y}$):

$$\widetilde{M}_{ij} = \frac{m_{ij}}{n_i} = \frac{\sum_{(y,\hat{y}) \in S_{NC}} 1_{\{y=i, \hat{y}=j\}}}{\sum_{(y,\hat{y}) \in S_{NC}} 1_{\{y=i\}}}, \tag{7.6}$$

where m_{ij} is the number of times that the label was observed to change from i to j due to the noise process and n_i is the number of instances in D_C with label $y = i$. $\widetilde{M}$ is the estimated model of the noise process. This noise model is then integrated into the training process using Equation 7.4 and 7.5 and as visualized in Figure 7.1.

7.3 EXPECTED ERROR OF THE NOISE MODEL

The noise model obtained in Equation 7.6 is an approximation of the underlying true noise process estimated on a small number of instances D_C. In this section, we derive a formula for the expected error of the estimated noise model $\widetilde{M}$. This gives us insights into the factors that influence the noise model's quality as well as their effect.

Assumptions: In the following proofs, we assume that M describes a noise process following Equations 7.3 and 7.4. $\widetilde{M}_{ij}$ is estimated using Equation 7.6.

We study two sampling techniques on how to obtain the set of clean and noisy label pairs $|S_{NC}|$. Commonly, a fixed number of unlabeled instances n is obtained and then manually annotated with gold labels. The value of n_i then follows the distribution of classes in the data. We call this **Variable Sampling** as the value of n_i varies.

In contrast to that, for **Fixed Sampling**, for each label value i, we sample n_i instances with $y = i$. This could be conducted e.g. by asking annotators to provide a specific number of labeled instances per class. In this case, n_i is fixed. For readability, we write $\mathbb{E}$ for $\mathbb{E}_{\sim S_{NC}}$ and analogously for Var and Cov. We assume that the instances are sampled independently.

As quality metric for evaluating the noise model, we use squared error which is in this matrix case the square of the Frobenius norm

$$SE = ||M - \widetilde{M}||_F^2 = \sum_{i=1}^{k}\sum_{j=1}^{k}(M_{ij} - \widetilde{M}_{ij})^2 \,. \tag{7.7}$$

Theorem 1
The expected squared error of the noise model is

$$\mathbb{E}[SE] = \sum_{i=1}^{k}\sum_{j=1}^{k}\mathrm{Var}[\widetilde{M}_{ij}] \,.$$

Sketch of the proof: We show that $\widetilde{M}$ is an unbiased estimator, i.e. $\mathbb{E}[\widetilde{M}_{ij}] = M_{ij}$. From that, Theorem 1 can be followed. For readability, the full proofs are given at the end of the chapter.

Theorem 2a Assuming *Variable Sampling*, it holds $\mathrm{Var}[\widetilde{M}_{ij}] = M_{ij}(1 - M_{ij}) \sum_{n_i=1}^{n} P(n_i)\frac{1}{n_i}$ where $P(n_i)$ is the probability of sampling n_i instances with label $y = i$ from the data.

Theorem 2b Assuming *Fixed Sampling*, it holds $\mathrm{Var}[\widetilde{M}_{ij}] = \frac{M_{ij}(1-M_{ij})}{n_i}$.

Sketch of the proof: The proofs for both variants of the theorem work on the main insight that given n_i, the value of m_{ij} follows a multinomial distribution defined by M_{ij}.

Combining Theorem 1 and 2, we obtain a closed-form solution for the expected error of the estimated noise model for both Fixed and Variable Sampling. From this, we can see that

- the error changes with the amount of sampled instances by factor $\frac{1}{n_i}$.
- the error depends on the noise distribution as well as the level of noise M_{ij}. In the single-flip scenario, it reaches its maximum when the noise is as dominant as the true label value.
- Fixed Sampling obtains lower error than Variable Sampling in most cases.

These results are visualized and experimentally verified below.

7.4 DATA WITH SYNTHETIC NOISE

Experiments with synthetic or simulated noise allow fine-grained control of the noise level and type of noise. An existing dataset is taken and the labels are assumed to be all correct and clean. Then, to obtain a noisy label dataset, for each instance, the label is flipped according to the noise process. Reed et al. (2015) and Goldberger and Ben-Reuven (2016) use the MNIST dataset (LeCun et al., 1998) and apply single-flip noise (Equation 7.2) to obtain the noisy labels. We follow their approach and label-flip pattern. Additionally, we also generate noisy labels with uniform noise (Equation 7.1) and a more

complex, multi-flip noise where one label can be changed into one of several incorrect labels. All three noise types are visualized in Figure 7.2. We see the multi-flip noise as the most realistic of these synthetic noises, as it resembles most the two realistic datasets presented in the next section.

7.5 DATA WITH REALISTIC NOISE

While evaluating on synthetically generated noise is popular and allows for an easy evaluation in a controlled environment, it is limited by the assumptions on the noise. Certain assumptions are taken when building a model of the noise and the same assumptions are used to generate the noisy labels.

In real-world scenarios, some of these assumptions might not apply. Inspecting realistic noise matrices (Figures 7.3 and 7.4), it is already quite obvious that these do not resemble the popular uniform or single-flip noise. We think it is therefore important to also evaluate on more realistic data that does not rely on the noise being simulated. Nevertheless, having parallel clean and noisy labels for the same instances is very useful as it allows e.g. to compute the upper bounds of training on clean data compared to training with noisy labels. In this specific work, it is required to obtain an approximation of the true noise pattern and to flexibly vary the number of clean labels. The Clothing1M dataset by Xiao et al. (2015) and our newly proposed NoisyNER dataset offer these possibilities.

7.5.1 *Clothing1M*

The Clothing1M dataset is part of a classification task to label clothing items present in an image. The noisy labels were obtained through a distant supervision process that used the text context of the images appearing on a shopping website. For 37k images, both clean and noisy labels are available. The percentage of correct labels in the noisy data is 38% and a visualization of the noise is given in Figure 7.3. One can see that the noise distributes neither uniformly nor is there a single label flip. Rather a label tends to be confused with several other related labels, e.g. a "Jacket" with a "Hoodie" and a "Downcoat".

7.5.2 *NoisyNER*

In this work, we propose another noisy label dataset. It is from the text classification domain with word-level labels for named entity recognition (NER). The labels are persons, locations and organizations. The language is Estonian, a typical low-resource language with a demand for natural language processing techniques. The text and the clean labels were collected by Laur (2013) through expert annotations

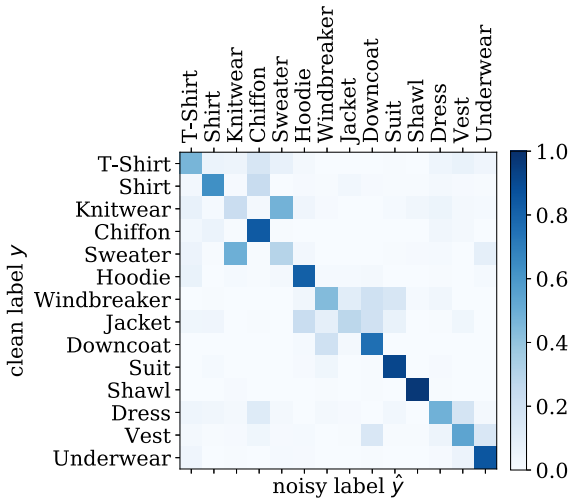


Figure 7.3: Noise matrix for Clothing1M computed over all pairs of clean and noisy labels.

(Tkachenko et al., 2013). The noisy labels are obtained through a distant supervision approach. Via the ANEA tool, lists of named entities were extracted from Wikidata and matched against the words in the text as presented in Chapter 5.

However, these labels are not error-free. Reasons include non-complete lists of entities, grammatical complexities of Estonian that do not allow simple string matching or entity lists in conflict with each other (e.g. "Tartu" is both the name of an Estonian city and a person name). The heuristic functions in ANEA allow to leverage insights from experts and they can be applied to correct some of these error sources, e.g. by normalizing (lemmatizing) the grammatical form of a word or by excluding certain high false-positive words. Specifically the following manual heuristics were chosen after inspecting the automatic annotation:

- **Label Set 1**: No heuristics.
- **Label Set 2**: Applying Estonian lemmatization to normalize the words.
- **Label Set 3**: Splitting person entity names in the list, i.e. both first and last names can be matched separately. Person names must have a minimum length of 4. Also, lemmatization.
- **Label Set 4**: If entity names from two different lists match the same word, location entities are preferred. Also, lemmatization.
- **Label Set 5**: Locations preferred, lemmatization, splitting names with minimum length 4.

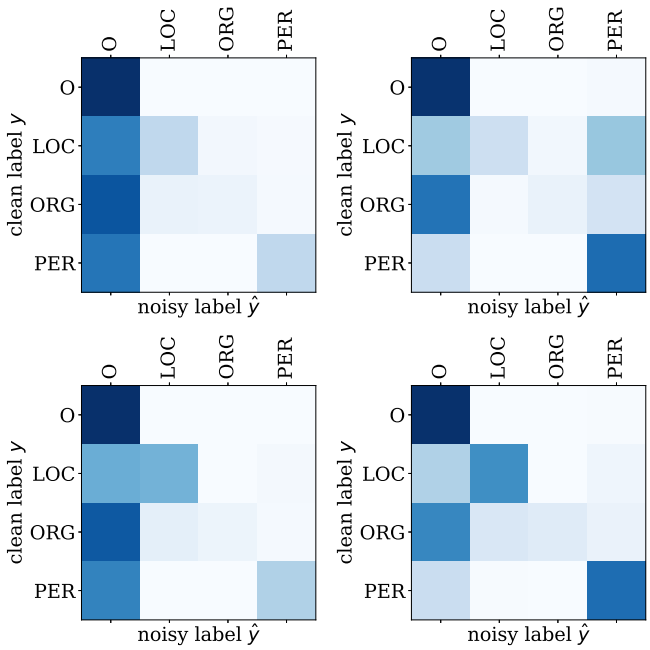


Figure 7.4: Noise matrices for NoisyNER (label sets 1, 3, 4 and 7) computed over all pairs of clean and noisy labels.

- **Label Set 6**: Removing the entity names "kohta", "teine", "naine" and "mees" from the list of person names (high false-positive rate). Also, all of label set 5.
- **Label Set 7**: Using alternative, alias names for organizations and retrieving additional entity lists for locations and organizations. Also, all of label set 6.

In contrast to Clothing1M, we provide seven sets of labels that differ in the noise process and therefore in the amount of noise and the noise pattern. Four of these are visualized in Figure 7.4. Table 7.1 lists an overview of the percentage of correct labels. The different amounts of heuristics reflect different levels of manual effort that one might be able to spend on the creation of distantly supervised data. Having these multiple sets of labels for the same instances allows to directly evaluate for different noise levels while excluding side effects that differing features might have. We want to highlight several properties of the dataset:

- Clean labels are available for all instances. This allows studying different splits of clean and noisy labels as well as computing upper bounds of performance on only clean data.
- The distribution of the labels is skewed. Out of the ca. 217k instances, ca. 8k are persons, 6k are locations and 6k are organizations. All other instances are labeled as non-entity O. Such a

Label Set	1	2	3	4	5	6	7
Precision	67	73	37	75	48	53	59
Recall	18	27	31	27	41	41	49
F1	28	39	34	40	44	46	54

Table 7.1: Percentages of correct labels in the NoisyNER dataset for the seven different label sets. For each subsequent label set, more manual effort in the labeling heuristics was invested. As the label distribution is highly skewed, precision, recall and F1 score are reported. Following the standard approach for named entity recognition (Tjong Kim Sang and De Meulder, 2003), the micro-average is computed excluding the non-entity label.

skewed label distribution is typical for a named entity recognition dataset.

- In past works, experiments were often only performed until the probability of a noisy class reaches the probability of the true class, i.e. it is assumed that $p(\hat{y} = j|y = i) < p(\hat{y} = i|y = i)\ \forall j \neq i$. This assumption does not hold for several of our label sets which can make learning on the data more challenging.
- While not studied in this work, the labels in the dataset also contain sequential dependencies. A clean or noisy label can span over several words/instances, e.g. for the mention of a person with a first and last name. These sequential dependencies could be leveraged in future work.

7.6 ANALYSIS OF THE NOISE MODEL ERROR

In this section, the theoretically expected squared error between the noise model estimate and the true noise matrix is compared to the empirically measured one. We vary the two parameters found in Theorem 2: the amount of sampled data n_i/n and the amount of noise M_{ij}. For Clothing1M only the data size can be varied while for NoisyNER the variation of the sample size can be compared across different noise levels and noise distributions.

7.6.1 *Experimental Setup*

From the full dataset, a small subset D_C and corresponding S_{NC} is sampled uniformly at random either using the Fixed or Variable Sampling approach. The noise model $\widetilde{M}$ is estimated on this sample and compared to the true noise process M. The process is repeated

500 times and the average empirical squared error is reported as well as its standard deviation on the error bars.

For the synthetic noisy labels, the true noise process is known by construction. For the realistic datasets, the true noise process is unknown. Instead, the noise matrix M is computed over the whole data as an approximation. For the distribution $P(y)$, that is part of the Variable Sampling formula, we assume a uniform distribution for MNIST and a multinomial distribution for Clothing1M and NoisyNER with the parameters of the distribution estimated over the whole dataset. In praxis, one might also rely on expert knowledge for this distribution (e.g. from Augenstein et al. (2017) for NER).

7.6.2 *Results & Analysis*

On the synthetic data (Figure 7.5), the theoretically expected error of the noise model follows closely the empirical measurements. This holds across different noise types, sample sizes and noise levels. Only for a very small set of clean labels in combination with a high noise level, there is a slight deviation.

As stated above, we can see the influence of the sample size and noise process. The error changes with the amount of sampled instances by factor $\frac{1}{n_i}$ and it depends on the noise distribution as well as the level of noise M_{ij}. For the evaluated scenarios, due to the additional dependency on the clean label distribution $P(y)$, the Variable Sampling technique has a higher expected error than the Fixed Sampling approach, especially for settings with large noise. From the empirical experiments, one can see that the variance of the noise model error mostly depends on the sample size.

The theoretical and experimental results also match on the realistic noisily labeled datasets Clothing1M and NoisyNER (Figure 7.6). Again, only for the very low sample size, one can observe a deviation. It is interesting to note that for NoisyNER the estimation error of the noise model is higher for the data with overall lower noise level (measured in F1 score in Table 7.1). This is due to how the noise distribution changes in the realistic setting. The difference between Fixed Sampling and Variable Sampling is most noticeable for NoisyNER increasing the error by a factor of around 3. This suggests that in practice, especially for such skewed distributions, a sampling technique is beneficial which focuses on each label separately.

7.7 ANALYSING THE BASE MODEL PERFORMANCE

In the previous sections, we studied the factors on which the quality of the noise model's estimate depends. The noise model is part of a larger classifier and it is combined with the actual base model that performs the task-specific classification (cf. Figure 7.1). In this section,

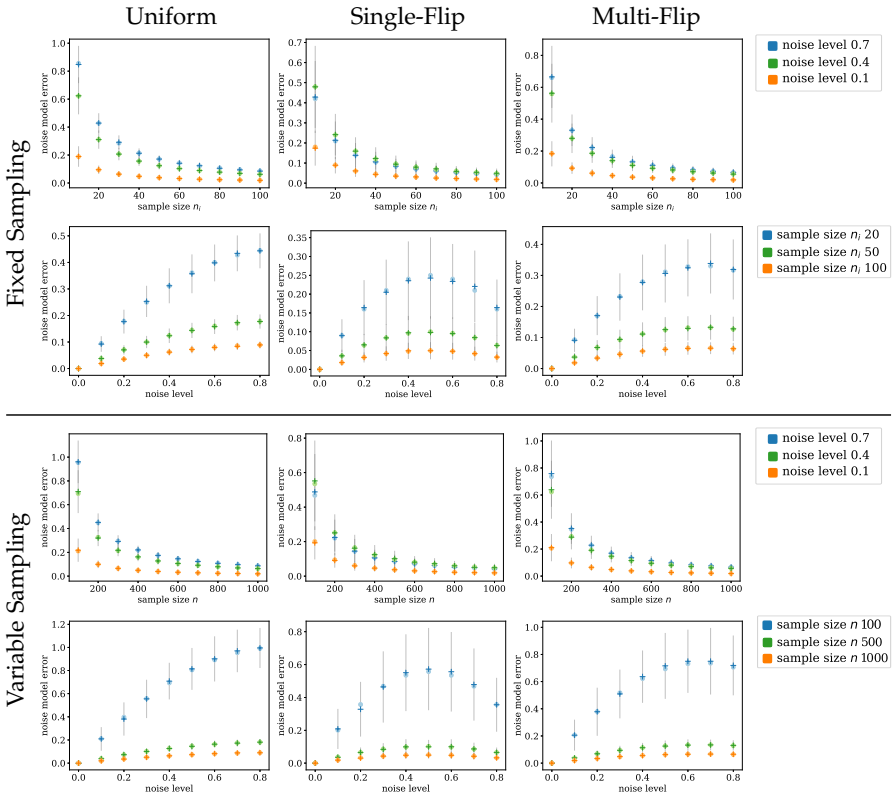
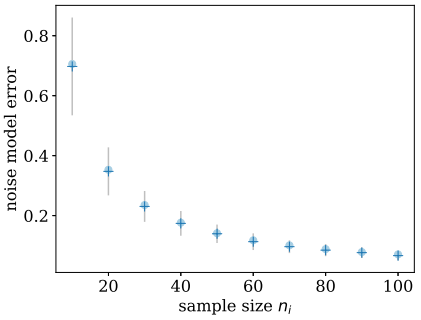
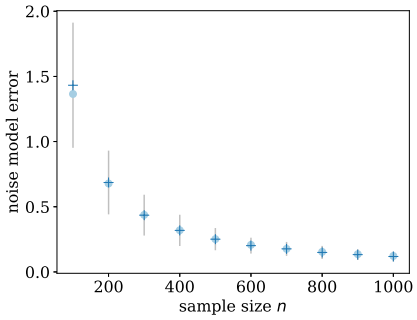


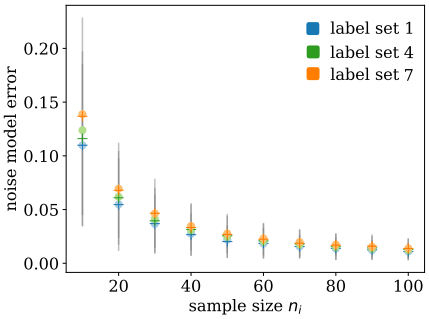
Figure 7.5: Comparison between the theoretically expected mean error (circle marker) and the empirically measured mean error (cross) of the noise model on the MNIST dataset. In the columns, the three different synthetic noise types "uniform", "single-flip" and "multi-flip" are given. Upper part uses Fixed Sampling, lower part uses Variable Sampling. On the x-axis, either the sample size (n_i or n respectively) or the noise level ϵ is varied. Error bars show the empirical standard deviation.



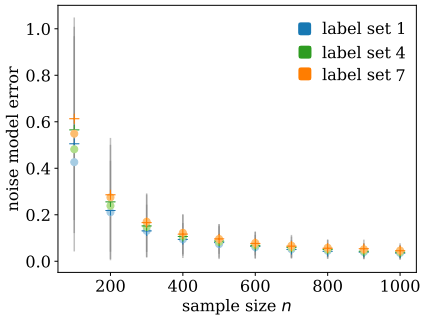
(a) Clothing1M - Fixed Sampling



(b) Clothing1M - Variable Sampling



(c) NoisyNER - Fixed Sampling



(d) NoisyNER - Variable Sampling

Figure 7.6: Comparison between the theoretically expected mean error (circle marker) and the empirically measured mean error (cross) of the noise model on the Clothing1M and NoisyNER datasets.

we evaluate in different experiments the base model on clean test data and analyze the effects of the noise modelling and the noise model estimation on the base model and its task performance.

7.7.1 *Experimental Details*

As in the experimental setup detailed in the previous section, a clean subset D_C of the data is sampled and the noise model estimated on it. The base model is then trained directly on the clean data D_C. The noise model is added to the base model for training on the noisy dataset D (cf. Figure 7.1). For evaluation, a test set is held-out from the data. We follow the training procedure presented in Chapter 4. Due to the longer runtimes, the experiments are repeated 50 times for Clothing1M and NoisyNER. The error bars show the empirical standard deviation. For additional plots, we refer to the Appendix of (Hedderich et al., 2021c).

ON NOISYNER For the NoisyNER dataset, we create an 80/10/10 train/dev/test split. Following the standard approach for named entity recognition (Tjong Kim Sang & De Meulder, 2003), we evaluate with the micro-average F1 score excluding the non-entity label. For each run of the experiment, a small subset D_C is sampled uniformly at random from the full dataset. Either Fixed or Variable Sampling are used. The noise model $\widetilde{M}$ is estimated on this sample.

As base model, we use the architecture for named entity recognition proposed in Chapter 4. It consists of a Bi-LSTM model (state size 300 for each direction) with an additional fully connected layer (size 100) and a softmax classification layer. The context size is 3 on each side. The tokens are embedded using fixed FastText embeddings (Grave et al., 2018). The weights of the noise matrix $\widetilde{M}$ are fixed during training. The model is optimized using Adam (Kingma & Ba, 2014) with a learning rate of 0.001. Training is performed for 80 epochs. We test using the learned weights of the epoch that performed best according to the clean development set. In each epoch, the model is alternately trained on the clean and the noisy instances (the latter with the noise model). Following the findings in the earlier chapter, the model is trained with a noisy subset of the full noisy data. This noisy subset is sampled uniformly at random for each epoch. We set the size to 15 times $|D_C|$.

In the experiments where the number of clean instances is fixed for the base model (to separate the effect on noise estimation from the effect of the base model just having access to more clean data), the base model is trained on 50 instances per class for Fixed Sampling and 50 instances in total for Variable Sampling. The other hyperparameters remain the same in both cases.

ON CLOTHING1M For the Clothing1M dataset, we extract the 33,747 images from the original dataset where both clean and noisy labels are available. Among these images, we use a 90/10 split for training/test sets. Again we take a small fraction out from the training set as our D_C and in training, the noise matrix is estimated by S_{NC}.
As base model, we employ a pre-trained ResNet18 (He et al, 2016) classifier obtained from the Pytorch library (Paszke et al, 2019). Specifically, we first switch the CNN header (i.e. the final linear classifier) to have the correct output dimension (14 in our case). Then we only train the new header and freeze all other layers. We fine-tune the network for 10 epochs. In training, an Adam optimizer with a learning rate of 0.005 is used for training. Again the noisy subset used in training is sampled uniformly at random for each epoch, with a size of 10 times the $|D_C|$.

In the experiments where the amount of clean labels is fixed for the base model, the base model is trained on 25 clean instances per class for Fixed Sampling and 250 instances in total for Variable Sampling. Again, the other hyperparameters remain the same in both settings.

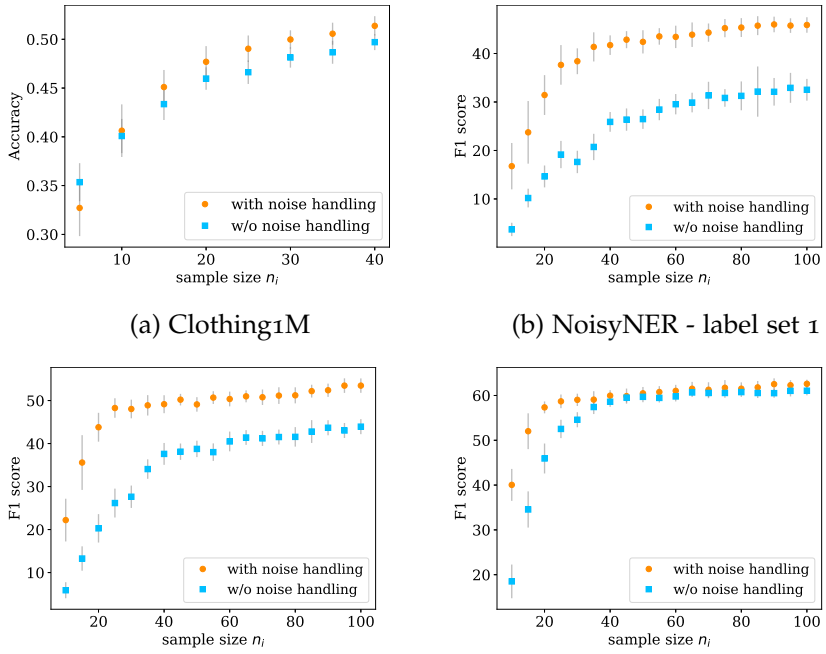
7.7.2 *Effect of Noise Handling*

Figure 7.7 shows the test performance of the base model for increasing size of D_C. It compares training the base model directly on the clean and the noisy label data to handling the noisy labels via a noise model. The latter improves the results in most cases. This confirms past findings that noise handling is an important technique to leverage distantly supervised training data. Comparing the different noise levels on NoisyNER, one can see that larger noise levels also result in larger improvements via noise handling. Only in a few cases with a very small amount of clean training samples does noise handling deteriorate the results, possibly due to a bad estimation of the noise model.

7.7.3 *Relationship between Noise Estimation and Base Model Performance*

There are several factors on which the base model's performance depends. These could include the amount of clean and noisy training data, the noise distribution and the quality of the estimated noise model. Here, we show experimentally on Clothing1M and NoisyNER that the noise model estimation error directly influences the performance of the base model.

In Figure 7.8, the expected noise model error is plotted against the test performance of the base model for Fixed Sampling. They show a clear negative correlation. The influence of the noise distribution is visible in the different slopes in the plots for the different noisy label sets of NoisyNER. For all settings, the Pearson Correlation between



(a) Clothing1M (b) NoisyNER - label set 1

(c) NoisyNER - label set 4 (d) NoisyNER - label set 7

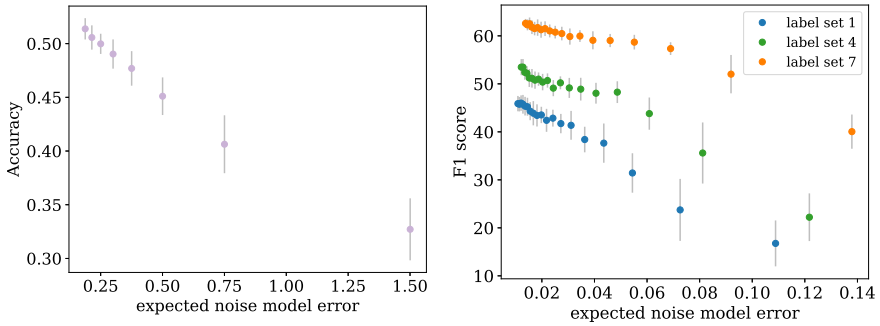
Figure 7.7: Mean test performance (F1 score) of the base model on Clothing1M and NoisyNER (label sets 1, 4 and 7) on clean and noisy data with and without noise handling using Fixed Sampling.

the mean test performance and the expected noise model error is at least -0.96.

Increasing the number of clean labels is not only beneficial to the estimation of the noise model but also to the base model itself. To remove this effect, the experiment is repeated with a fixed amount of training instances for the base model ($n_i = 50$ for NoisyNER and $n_i = 25$ for Clothing1M) and a varying amount of clean labels for the noise model estimation. The same linear relationship can be seen. The Pearson Correlation is again at least -0.96 for all settings.

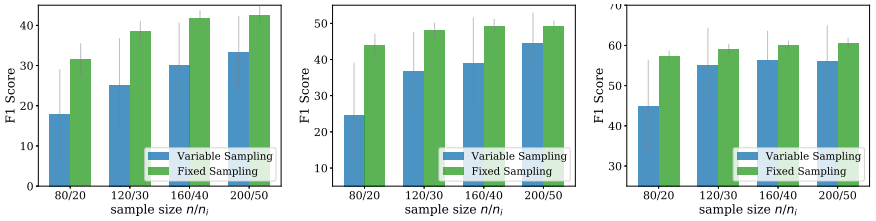
7.7.4 *Effect of Sampling during Estimation*

Above, we saw both from the theoretical analysis as well as in the experiments that Variable versus Fixed Sampling has a strong influence on the noise model estimation error. Figure 7.9 shows that this effect also transfers to the performance of the base model on the test set. Fixed Sampling consistently outperforms Variable Sampling on the average performance across different noise levels. It also reduces the variance of the reached test performance, another important issue models trained on noisy labels suffer from. The same holds again when a fixed amount of training instances for the base model is used to remove the effect of just having more clean data.



(a) Clothing1M (b) NoisyNER

Figure 7.8: Relationship between the theoretically expected noise model error and the test performance (Accuracy/F1 score) of the base model for Clothing1M and NoisyNER (label set 1, 4 and 7) with Fixed Sampling. Each point corresponds to one sample size n_i (cf. Figure 7.7).



(a) label set 1 (b) label set 4 (c) label set 7

Figure 7.9: Comparing Variable and Fixed Sampling by mean test performance (F1 score) of the base model on NoisyNER label set 1, 4 and 7. The two approaches were given the same amount of clean data with $n_i = n/k$ for Fixed Sampling.

As far as we are aware, Variable Sampling is more common in the literature, suggesting that Fixed Sampling could be a useful and model-independent way to improve noise handling.

7.8 OTHER RELATED WORK

Apart from the publications mentioned in the previous sections, we want to highlight the following related work. The survey by Frenay and Verleysen (2014) gives a comprehensive overview of the literature on learning with noisily labeled data. More recently, Algan and Ulusoy (2021) surveyed deep learning techniques for noisy labels in the image classification domain.

Other aspects of learning with noisy labels have been studied from a theoretical viewpoint. For the binary classification setting, Liu and Tao (2016) prove an upper bound for the noise rate. Natarajan et al. (2013) and Patrini et al. (2016) derive upper bounds on the risk when learning in a similar noise scenario. Assuming a known and correct noise model, Rooyen and Williamson (2017) derive upper and lower bounds on learning with noisy labeled data. The existence of anchor points is an important assumption for several recent works on estimating a noise matrix, i.a. by Liu and Tao (2016) and Patrini et al. (2017) and Xia et al. (2019). Yao et al. (2020) propose to learn an intermediate class to reduce the estimation error of the noise matrix. Chen et al. (2019b) propose a formula to calculate the test accuracy assuming, however, that the test set is noisy as well.

On the empirical side, Veit et al. (2017) model the relationship $p(y|\hat{y})$, i.e. the opposite of the noise models in this work. They also use pairs of clean and corresponding noisy labels. Rahimi et al. (2019) include the concept of noise transition matrices in a Bayesian framework to model several sources of noisy supervision. In their few-shot evaluation, they also leverage pairs of clean and noisy labels. Ortego et al. (2020) study the behaviour of the loss for different noise types. Some recent works have taken instance-dependent (instead of only label-dependent) noise into consideration both from a theoretic and an empiric viewpoint (Cannings et al., 2018; Cheng et al., 2020; Luo et al., 2017; Menon et al., 2018; Xia et al., 2020). Luo et al. (2017) successfully modeled instance-dependent noise for an information extraction task, making instance-dependence an interesting future work for the NoisyNER dataset.

In the image classification domain, further noisy label datasets exist. The distantly supervised WebVision (Li et al., 2017) was obtained by searching for images on Google and Flickr where the labels (and related words) are used as search keywords. It contains 2.4 million images but it lacks clean labels for the training data. The Food101N dataset (Lee et al., 2018) was created similarly, focusing, like Clothing1M, on a specific image domain. The authors provide ca. 300k

images, ca. 50k of which have additional, human-annotated labels. The noise rate is lower with around 20%. The older Tiny Images dataset (Torralba et al., 2008) consists of 80 million images at a very low resolution with 3526 training images having clean labels. Recently, Jiang et al. (2020) proposed artificially adding different amounts of incorrectly labeled images to existing datasets for evaluation. For the task of text sentiment analysis, Wang et al. (2019) proposed a dataset that obtained sentence level labels by exploiting document level ratings. Clean sentence level labels exist for two of their studied text domains.

7.9 PROOFS

We now give the full proofs of the previously introduced theorems.

LEMMA 1: Assuming a noise process following Equation 7.3 and 7.4 with $p(\hat{y} = j|y = i) = M_{ij}$ and assuming $(y, \hat{y}) \in S_{NC}$ is sampled independently at random with fixed $n = |S_{NC}|$[3] then $\mathbb{E}_{\sim S_{NC}}[\widetilde{M}_{ij}]$ is an unbiased estimator of M_{ij}.

PROOF: For readability, we write $\mathbb{E}$ for $\mathbb{E}_{\sim S_{NC}}$ and analogously for Var and Cov. Let N_i be the random variable describing how often $y = i$, i.e. $N_i = \sum_{(y,\hat{y})\in S_{NC}} 1_{y=i}$. Let S be the random variable describing how often $y = i$ and $\hat{y} = j$ on the same instance, i.e. $S = \sum_{(y,\hat{y})\in S_{NC}} 1_{\{y=i,\hat{y}=j\}}$. Note that, given $N_i = n_i$ and independent sampling, S is multinomially distributed with probabilities M_i and n_i trials. $\widetilde{M}$ is defined by Equation 6. For $n_i = 0$, we define $\widetilde{M}_{ij} = 0$.

3 This holds both for *Fixed* and *Variable Sampling*.

$$
\begin{aligned}
\mathbb{E}[\widetilde{M}_{ij}] &= \mathbb{E}[\frac{\sum\limits_{(y,\hat{y})\in S_{NC}} 1_{\{y=i,\hat{y}=j\}}}{\sum\limits_{y\in S_{NC}} 1_{\{y=i\}}}] \\
&= \sum_{s,n_i=1}^{n} P(S=s, N_i=n_i)\frac{s}{n_i} \\
&= \sum_{s,n_i=1}^{n} P(S=s|N_i=n_i)P(N_i=n_i)\frac{s}{n_i} \\
&= \sum_{n_i=1}^{n} P(N_i=n_i)\frac{1}{n_i}\sum_{s=1}^{n_i} P(S=s|N_i=n_i)s \\
&= \sum_{n_i=1}^{n} P(N_i=n_i)\frac{1}{n_i}\mathbb{E}[S|N_i=n_i] \\
&= \sum_{n_i=1}^{n} P(N_i=n_i)\frac{1}{n_i}n_i M_{ij} \\
&= M_{ij}\ .
\end{aligned}
$$

THEOREM 1: Let the assumptions be as in Lemma 1 and let SE be defined as in Formula 7.7. Then $\mathbb{E}[SE] = \sum_{i=1}^{k}\sum_{j=1}^{k} \mathrm{Var}[\widetilde{M}_{ij}]$.

PROOF: Using Lemma 1, the proof follows from the definition of SE:

$$
\begin{aligned}
\mathbb{E}[SE] &= \mathbb{E}\left[\sum_{i=1}^{k}\sum_{j=1}^{k}(M_{ij}-\widetilde{M}_{ij})^2\right] \\
&= \sum_{i=1}^{k}\sum_{j=1}^{k}\mathbb{E}[(\mathbb{E}[\widetilde{M}_{ij}]-\widetilde{M}_{ij})^2] \\
&= \sum_{i=1}^{k}\sum_{j=1}^{k}\mathrm{Var}[\widetilde{M}_{ij}]\ .
\end{aligned}
$$

THEOREM 2A: Let the assumptions be as in Theorem 1 and assuming *Variable Sampling*, it holds $\mathrm{Var}[\widetilde{M}_{ij}] = M_{ij}(1-M_{ij})\sum\limits_{n_i=1}^{n} P(n_i)\frac{1}{n_i}$ where $P(n_i)$ is the probability of sampling n_i instances with label $y=i$ from the data.

PROOF:

$$
\begin{aligned}
\mathrm{Var}[\widetilde{M}_{ij}] &= \mathbb{E}[\widetilde{M}_{ij}^2] - \mathbb{E}[\widetilde{M}_{ij}]^2 \\
&= \mathbb{E}[\widetilde{M}_{ij}^2] - M_{ij}^2 \\
&= \sum_{s,n_i=1}^{n} P(S=s, N_i=n_i)\frac{s^2}{n_i^2} - M_{ij}^2 \\
&= \sum_{s,n_i=1}^{n} P(S=s|N_i=n_i)P(N_i=n_i)\frac{s^2}{n_i^2} - M_{ij}^2 \\
&= \sum_{n_i=1}^{n} P(N_i=n_i)\frac{1}{n_i^2}\sum_{s=1}^{n_i} P(S=s|N_i=n_i)s^2 - M_{ij}^2 \\
&= \sum_{n_i=1}^{n} P(N_i=n_i)\frac{1}{n_i^2}\sum_{s=1}^{n_i} \mathbb{E}[S^2|N_i=n_i] - M_{ij}^2 \\
&= \sum_{n_i=1}^{n} P(N_i=n_i)\frac{1}{n_i^2}(n_i^2M_{ij}^2 + n_iM_{ij}(1-M_{ij})) - M_{ij}^2 \\
&= M_{ij}(1-M_{ij})\sum_{n_i} P(N_i=n_i)\frac{1}{n_i}\,,
\end{aligned}
$$

where we use $\mathbb{E}[S^2] = \mathrm{Var}[S] + \mathbb{E}[S]^2$ with $\mathrm{Var}[S] = n_iM_{ij}(1-M_{ij})$ and $\mathbb{E}[S]^2 = n_i^2M_{ij}^2$ as S is multinomial distributed.

THEOREM 2B: Let the assumptions be as in Theorem 1 and assuming *Fixed Sampling*, it holds $\mathrm{Var}[\widetilde{M}_{ij}] = \frac{M_{ij}(1-M_{ij})}{n_i}$.

PROOF: Given *Fixed Sampling* with $n_i = n_i' \; \forall i$, it follows that $P(N_i = n_i') = 1$ and 0 otherwise. The derivation of the variance from Theorem 2a then simplifies to

$$
\mathrm{Var}[\widetilde{M}_{ij}] = \frac{M_{ij}(1-M_{ij})}{n_i'}\,.
$$

7.10 CONCLUSION

In this chapter, we analyzed the factors that influence the estimation of the noise model in noisy label modeling for weak supervision. We derived the expected error of a noise model estimated from pairs of clean and noisy labels highlighting factors like noise distribution and sampling technique. Extensive experiments on synthetic and realistic noise showed that it matches the empirical error. We also analyzed how the noise model estimation affects the test performance of the base model. Our experiments show the strong influence of the noise model estimation and how theoretical insights on e.g. the sampling technique can be used to improve the task performance.

A major contribution of this chapter is our newly created NoisyNER dataset, a named entity recognition dataset consisting of seven sets of labels with differing noise levels and patterns. It allows a simple and extensive evaluation of noisy label approaches in a realistic setting and for different levels of resource availability. With its interesting properties, we hope that it is useful for the community to develop noise handling methods that are applicable to real scenarios.

8

UNDERSTANDING THE REASONS FOR LABEL ERRORS

The ANEA tool presented in Chapter 5 allowed the users to inspect the automatically annotated data. They then could use the insights they obtained about annotation errors to improve the automatic process. Understanding where a system labels data incorrectly is crucial to being able to improve it. This is not limited to weak supervision but also holds for machine learning classifiers and their errors in general. In this chapter,[1] we propose the PREMISE algorithm that finds statistically significant label-descriptive patterns. This method allows us to describe in human-interpretable form where misclassifications occur. Unlike existing solutions, it ably recovers ground truth patterns, even over many unique items or where patterns are only weakly associated to labels. Through two studies on visual question answering and named entity recognition, we confirm that PREMISE gives clear and actionable insight into the systematic errors made by NLP classifiers.

8.1 INTRODUCTION

State-of-the-art deep learning methods are known for their ability to achieve human-like performance on challenging tasks. As much as 'to err is human,' these classifiers make errors too. Some of these errors are due to noise that is inherent to the process we want to model, and therewith relatively benign. Systematic errors, on the other hand, e.g. those due to bias or misspecification, are much more serious as these lead to models that are inherently unreliable. If we know under what conditions a model performs poorly, we can actively intervene, e.g. by augmenting the training data, and so improve overall reliability and performance. Before we can do so, we first need to know whether a classifier makes systematic errors, and if so, how to characterize them in easily understandable terms.

Given a dataset with labels that specify which instances were classified correctly or incorrectly, we are interested in finding combinations of features that describe where the classifier's predictions are incorrect. For a Natural Language Processing (NLP) task, the input features are words. If, for example, the combination of words "*how, many*" strongly correlates with misclassified instances, this can indicate that our classifier struggles with the concept of counting. A toy example is visualized in Figure 8.1.

1 This chapter is based on (Hedderich et al., 2022) with Jonas Fischer and Michael Hedderich contributing equally as first authors.

Instances	Correct Prediction?
How many ducks are in the picture?	✗
What are the ducks eating?	✗
How many roosters are in the puddle?	✗
Do you see ducks in the puddle?	✓
Are there many ducks playing?	✓

Figure 8.1: Toy example with input instances and the label specifying if the classifier predicted correctly. The pattern $\bigwedge$(*how, many*) correlates with misclassification. The word *ducks* is also a frequent pattern but independent of the label and therefore not of relevance.

Local explanation methods like LIME (Ribeiro et al., 2016) describe the decision boundary of each instance. In contrast, we are interested in an efficient way to obtain a global and non-redundant description of our classifier's issues on the given input data. To this end, we turn to data mining. Here, a combination of features is a pattern, and we look for the set of patterns that best characterizes on which instances the classifier tends to perform well or poorly. This can be phrased as the more general problem of label description. For data with binary features, we are interested in the associations between the feature data and the labels. We formulate this problem in terms of the Minimum Description Length (MDL) principle, which identifies the best set of patterns as the one that best compresses the data without loss.

To capture phenomena of text input, e.g. synonyms, we consider a rich pattern language that allows us to express conjunctions, mutual exclusivity, and nested combinations thereof. As the search space is double exponential and does not exhibit any easy-to-exploit structure, we propose the efficient and hyper-parameter-free Premise algorithm to heuristically discover the *premises* under which we see the given predictions.

We evaluate Premise both on synthetic and real-world data. We show that, unlike the state of the art in data mining, Premise is robust to noise, scales to large numbers of features, and deals well with class imbalance, as well as varying association strength of patterns to labels. Through two case studies, we show that Premise discovers patterns that provide clear insight into the systematic errors of NLP classifiers. For Visual Question Answering (VQA), we elucidate the issues of two classifiers (Tan and Bansal, 2019; Zhu et al., 2016), including aspects like counting, spatial orientation and higher reasoning. For a neural Named Entity Recognition (NER) model (Ma and Hovy, 2016b), we show that Premise discovers patterns that are both interpretable and that can be acted upon.

8.2 RELATED WORK

Label Description in Data Mining

Describing labels in terms of features is obviously related to classification. Here, however, we are not so much interested in prediction, but rather description and therewith value interpretability of the results over accuracy. We share this notion with emerging pattern mining (Dong and Li, 1999) which aims to discover those conditions under which a target attribute has an exceptional distribution. The key difference is that we are not interested in discovering *all* patterns that are associated, which would be overly redundant and hard to interpret as a whole, but rather want a small and non-redundant set of patterns capturing relevant associations.

Subgroup discovery (García-Vico et al., 2018; Wrobel, 1997) returns the top-k patterns that correlate most strongly. This keeps the result sets of manageable size but does not solve the problem of redundancy (Leeuwen and Knobbe, 2012). Statistical pattern mining aims to discover patterns that correlate *significantly* to a class label (Llinares-López et al., 2015; Papaxanthos et al., 2016; Pellegrina and Vandin, 2018; Webb, 2007). In practice, these methods discover many hundreds of thousands of 'significant' patterns even for small data. For surveys we refer to (Atzmueller, 2015; García-Vico et al., 2018; Novak et al., 2009).

Rule mining aims to discover rules of the form $X \to Y$ (Agrawal et al., 1993; Hämäläinen, 2012), lending themselves to describe labels, too. Like above, most existing methods evaluate patterns individually, thereby discovering millions of rules even if the data is pure noise. GRAB (Fischer and Vreeken, 2019) instead mines small sets of rules that together summarize the data well, and CLASSY (Proença and van Leeuwen, 2020) discovers rule lists characterizing a given label. We show that both approaches do not scale well and are sensitive to label imbalance.

Explainable ML and Misclassification

Specifically to explain classifiers, several approaches aim to capture dependencies of features or attributes that a classifier uses to make a prediction, e.g. in terms of patterns or rules (Barakat and Diederich, 2005; Henelius et al., 2014), by model distillation (Frosst and Hinton, 2017; Lakkaraju et al., 2017), or to discover patterns of neurons within neural networks that drive a decision (Fischer et al., 2021). These, however, focus on the dependencies the classifier exploits for successful prediction as opposed to understanding where – or why – something goes wrong. Here, SliceFinder (Chung et al., 2020) explains where a classifier performs particularly poorly in terms of feature subspaces.

However, the approach was only evaluated on data with less than 50 features. Our experiments show that this method does not scale well to the feature spaces common in NLP data.

For specific applications in NLP, there exist manual approaches based on challenging test sets (Gardner et al., 2020; Ribeiro et al., 2020) or testing a hypothetical cause for misclassification (Lee et al., 2019; Rondeau and Hazen, 2018; Wu et al., 2019). Such manual approaches, however, require existing knowledge about the difficulties of the models. Local explanation methods like LIME (Ribeiro et al., 2016) provide insights into what changes in the input influence a classifier's decision on a specific instance. ANCHORS (Ribeiro et al., 2018) obtains such explanations in an interpretable form similar to our patterns. As they need to explore the local decision boundary, they require, however, multiple classifier evaluations per instance. For a survey focused on local methods for explainable NLP, we refer to Danilevsky et al. (2020).

Here, we propose to mine sets of patterns that provide concise, interpretable, and global descriptions of the given label, which we formulate in MDL terms. We further propose an efficient heuristic to discover such pattern sets in practice, which we test against state-of-the-art across all aforementioned fields on synthethic data with known ground truth, as well as real world case studies. We show that PREMISE is the only approach to be scalable and robust to noise and label imbalance while retrieving succinct pattern sets, all of which is crucial to tackle real world applications.

8.3 PRELIMINARIES

In this section, we introduce the notation we use throughout the paper and give a brief primer to MDL.

8.3.1 *Notation*

We consider binary data, such as a sequence of input words of an NLP task where each word of the vocabulary is a binary feature (bag-of-words, word is present or not present). In data mining terms, each instance of our dataset is a transaction and each word present in the instance is an item of the transaction. For each instance, we also have a label that expresses whether the instance is misclassified by our classifier. Our whole dataset can then be described as binary transaction data D over a set of items $\mathcal{I}$, where each transaction $t \in D$ is assigned a binary label $\ell(t) \in \{l_-, l_+\}$. For ease of notation, we define the partition of the database according to this binary label $D^- = \{t \in D \mid \ell(t) = l_-\}$ and $D^+ = \{t \in D \mid \ell(t) = l_+\}$. In general, $X \subseteq \mathcal{I}$ denotes an itemset, the set of transactions that contain X is

defined as $T_X = \{t \in D \mid X \subseteq t\}$. The projection of D on an itemset X is $\pi_X(D) = \{t \cap X \mid t \in D\}$.

We are looking for human-interpretable associations of items that best explain a given database partition. We describe these associations in terms of 'patterns', which we define by logical conditions over sets of items. For a logical condition c, we define a selection operator as $\sigma_c(D) = \{t \in D \mid c(t) \equiv \top\}$. For an item $I \in \mathcal{I}$, it holds that $[c_I(t) \equiv \top \leftrightarrow I \in t]$. The k-ary AND operator $\bigwedge(c_1, \ldots, c_k)$ describes patterns of co-occurrence and holds iff all its conditions hold. Similarly, the k-ary XOR operator $\bigotimes$ describes patterns of mutual exclusivity and holds if exactly one of its condition holds. We denote $it(c)$ for the items in the condition and define the projection on a condition as $\pi_c(D) = \pi_{it(c)}(D)$. Conditions can be nested; specifically we are interested in patterns of AND operator over XOR operations, i.e. $\bigwedge(\bigotimes_{c_1,\ldots,c_k}, \ldots, \bigotimes_{c'_1,\ldots,c'_k})(t)$. An XOR operation is called clause, $\gamma(c)$ lists all clauses in conjunctive condition c. To simplify notation, we drop t when clear from context, write I for conditions on a single item $c(I)$, and use condition and pattern interchangeably.

8.3.2 *Minimum Description Length*

The Minimum Description Length (MDL) principle (Rissanen, 1978) is a practical approximation of Kolmogorov complexity (Li and Vitányi, 1993) that is both statistically well-founded and computable. It identifies the best model M^* for data D out of a class of models $\mathcal{M}$ as the one that obtains the maximal lossless compression. For refined, or one-part, MDL, the length of the encoding in bits is obtained using the entire model class $L(D|\mathcal{M})$. While this variant of MDL provides strong optimiality guarantess (Grünwald, 2007), it is only attainable for certain model classes. In practice, crude two-part MDL is often used, which computes the length of the model encoding $L(M)$ and the length of the description of the data given the model $L(D|M)$ separately. The total length of the encoding is then given as $L(M) + L(D|M)$. We use one-part MDL where possible and two-part MDL otherwise. When applying MDL, we are only interested in the codelengths and not the actual codes. Codelengths are measured in bits, hence all log operations are base 2 and we define $0 \log 0 = 0$.

8.4 THEORY

To discover those patterns best describing the given labels, we here introduce the class of models $\mathcal{M}$ and corresponding codelength functions that yield the number of bits required to encode a model, respectively the number of bits needed to encode data given a model. Before we define these formally, we give the intuition.

8.4.1 *The Problem, informally*

Given a dataset of binary transaction data and a binary label for each transaction, we aim to find a set of patterns that together identify the partitioning of the data according to the labels. As an application, consider the input words of an NLP task as transactions, along with labels that express whether an instance is misclassified by a given model. We are now interested in patterns of words that describe these labels. In essence, we want to find word combinations such as $\bigwedge$(*how, many*), or mutual exclusive patterns, e.g. $\bigotimes$(*color, colour*), that capture synonyms or different writing styles, all occuring predominantly when a misclassifcation happens. The pattern language we use here is a combination of the two, namely conjunctions of mutual exclusive clauses such as $\bigwedge$(*what,* $\bigotimes$(*color, colour*)). We provide an example in Figure 8.2.

We thus define a model $M \in \mathcal{M}$ as a set of patterns $\mathcal{P}$ containing all patterns that help to describe given labels. Additionally, to ensure that we can always encode any data over the set of items using our model, M contains all singleton words $I \in \mathcal{I}$, describing the entire data D label unspecific. The model containing all singletons also serves as a baseline implementing the assumption that there are no associations that describe the label. Whenever there is a structure in the labels that can be explained by a pattern, we transmit the data corresponding to a label (D^+, D^-) separately. This allows us to more succinctly transmit where a pattern holds.

Let us consider the example in Figure 8.2, where we would first send $\bigwedge(A, \bigotimes(B,\ C))$ occurrences in D^+, and then its occurrences in D^-. Thus, we identify where A, C, and D hold at once, and we leverage the fact that $\bigwedge(A, \bigotimes(B, C))$ occurs predominantly in D^+, resulting in more efficient transmission. Intuitively, a bias of a pattern to occur in one label more than in the other corresponds to a large deviation between the conditional probability – the pattern occurrence conditioned on the label – and the unconditional probability – the pattern occurence in the whole database. We hence transmit more efficiently by sending the pattern separately for D^+ and D^- if there is a large deviation between these two probabilities. Coming back to the example, F however occurs similarly often in both labels – there is almost no deviation between conditional and unconditional probability – hence it is unlikely that it identifies a structural error. Here, the baseline encoding transmitting F as singleton in all of D will be most efficient. This approach allows us to identify patterns that occur predominantly for one of the labels as the patterns that yield better compression when conditioned on the labels, and thus characterise labels in easily understandable terms.

We are hence after the model $M^* \in \mathcal{M}$ that minimizes the cost of transmitting the data and model. In the following sections, we will

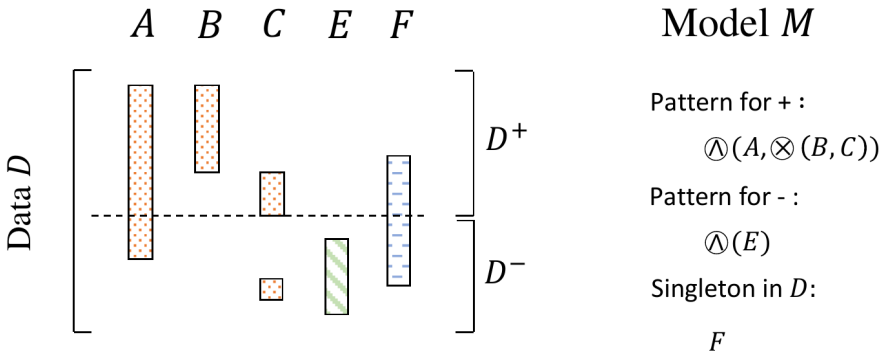


Figure 8.2: *Example database and model.* A toy database D over a set of items, separated by labels into D^+ and D^-, is given on the left. The corresponding model M containing patterns, which describe the data partitions D^- and D^+ induced by labels l_- and l_+, is given on the right.

formalize this intuition using an MDL score to identify that pattern set that best describes the data given the labels. We will first detail how to compute the encoding cost for the data given the model and then the cost for the model itself.

8.4.2 *Cost of Data Given Model*

Let us start by explaining how to encode a database D with singleton items I in the absence of any labels, which will later serve as the baseline encoding corresponding to independence between items and labels. To encode in which transaction an item I holds, optimal data-to-model codes are used, which are indices over canonically ordered enumerations (Li and Vitányi, 1993). For the data cost, we thus obtain

$$L\left(\pi_I\left(D\right) \mid I\right) = \log \binom{|D|}{|\sigma_I\left(D\right)|} . \tag{8.1}$$

Taking into account the partitioning of D along the label, yielding D^+ and D^-, we encode I separately:

$$L\left(\pi_I\left(D\right) \mid I\right) = \log \binom{|D^-|}{|\sigma_I\left(D^-\right)|} + \log \binom{|D^+|}{|\sigma_I\left(D^+\right)|} . \tag{8.2}$$

As such, we explicitly reward patterns (here, singletons) that have a different distribution between the unconditional probability, i.e. frequency in D of I and the conditional probability of I conditioned on the label – i.e. frequency in D^- respectively D^+. It models the property that we are interested in; a pattern that characterizes a certain label. It is straightforward to extend to patterns of co-occurring items $P = \text{Ⓐ}(X_1, \ldots, X_k)$ by selecting on transaction where the pattern holds

$$L\left(\pi_P\left(D\right) \mid P\right) = \log \binom{|D^-|}{|\sigma_P\left(D^-\right)|} + \log \binom{|D^+|}{|\sigma_P\left(D^+\right)|} . \tag{8.3}$$

There might be transactions where individual items of P are present, but not the full pattern holds. To ensure a lossless encoding, the singleton code $L(\pi_I(D) \mid I)$ is modified to cover all item occurrences left unexplained after transmitting all patterns. Hence, we get

$$L_s(\pi_I(D) \mid P) = \log \binom{|D|}{|\sigma_I(D) \setminus \left(\bigcup_{P \in \mathcal{P}, I \in P} \sigma_P(D)\right)|}.$$

For patterns expressing conjunctions over mutual exclusive items, e.g. $\owedge(\otimes(A,B),\otimes(C,D))$, we first send for both D^- and D^+ for which transactions the pattern holds, after which we specify which of the items is active where. We do that one by one, as we know that when the pattern holds and A is present, B cannot be present too. With each transmitted item of the clause, there are thus fewer transactions where the remaining items could occur, hence the codelength is reduced. More formally, the codelength for a pattern P of conjunctions of clauses is given as

$$L(\pi_P(D) \mid P) = \sum_{l \in \{-,+\}} \log \binom{|D^l|}{|\sigma_P(D^l)|} + \tag{8.4}$$

$$\sum_{cl \in \gamma(P)} \sum_{I \in cl} \log \binom{|\sigma_P(D^l)| - \sum_{I' \in cl, I' \leq I} |\sigma_{I'}(\sigma_P(D^l))|}{|\sigma_I(\sigma_P(D^l))|}, \tag{8.5}$$

assuming a canonical order on $\mathcal{I}$. With clauses of only length 1 we arrive at a simple conjunctive pattern, where the second term evaluates to 0 and thus resolves to the codelength function for conjunctive patterns discussed above. Note here that the codelength is the same regardless of the order assumed on the $\mathcal{I}$. This statement trivially holds for clauses of length 2. For readability, we provide the proof for the case of 3 items at the end in Section 8.8, the case for an arbitrary number of items follows the same reasoning.

This concludes the definition of codelength functions for transmitting the data, and we can define the overall cost of transmitting the data D given a model M as

$$L(D \mid M) = \Big(\sum_{P \in \mathcal{P}} L(\pi_P(D) \mid P)\Big) + \Big(\sum_{I \in \mathcal{I}} L_s(\pi_I(D) \mid P)\Big).$$

8.4.3 *Cost of the Model*

Let us now discuss how to transmit the model M for pattern set $\mathcal{P}$. First, we transmit the number of patterns $|\mathcal{P}|$ using the MDL-optimal code for integers $L_{\mathbb{N}}(|\mathcal{P}|)$. It is defined as $L_{\mathbb{N}}(n) = \log^* n + logc_0$ with $\log^* n = \log n + \log\log n + \ldots$ and c_0 being a constant so that $L_{\mathbb{N}}(n)$ satisfies the Kraft-inequality (Rissanen, 1983). Then, for each pattern P, we transmit the number of clauses via $L_{\mathbb{N}}(|\gamma(P)|)$. For each such clause, we transmit the items it contains using a log binomial,

requiring $\log \binom{|\mathcal{I}|}{|cl|}$ bits plus a parametric complexity term $L_{pc}(|\mathcal{I}|)$. The log binomial along with the parametric complexity form the normalized maximum likelihood code for multinomials, which is a refined MDL code. The parametric complexity for multinomials is computable in linear time (Kontkanen and Myllymäki, 2007). Lastly, we transmit the parametric complexities of all binomials used in the data encoding.

Combining the above, the overall cost of the model is

$$L(M) = L_{\mathbb{N}}(|\mathcal{P}|) + \sum_{P \in \mathcal{P}} (L_{\mathbb{N}}(|\gamma(P)|) + L_{pc}(|D^+|) + \tag{8.6}$$

$$L_{pc}(|D^-|)) + \sum_{cl \in P} \left(\log \binom{|\mathcal{I}|}{|cl|} + L_{pc}(|\mathcal{I}|) \right) + \sum_{I \in \mathcal{I}} L_{pc}(|D|) \,, \tag{8.7}$$

by which we have a lossless MDL score.

8.4.4 *The Problem, formally*

Based on the above, we can now formally state the problem.

MINIMAL LABEL DESCRIPTION PROBLEM

Given data D over $\mathcal{I}$ and a split into D^- and D^+, find the model $M \in \mathcal{M}$ that minimizes the codelength $L(M) + L(D \mid M)$.

Solving this problem through enumeration of all possible models is computationally infeasible as the model space is too large. Specifically, the size of the model space is given by

$$|\mathcal{M}| = 2^{\sum_{i=1}^{|\mathcal{I}|} \binom{|\mathcal{I}|}{i} \times \sum_{j=1}^{i} \left\{ {i \atop j} \right\}} \,,$$

where the first term in the summation specifies the number of possible item combinations in a pattern of length i, the second term counts the number of possible ways to separate them into j different clauses via the Stirling number of the second kind and the exponent is introduced as a model M consists of arbitrary combinations of patterns. Next, we introduce an efficient bottom-up search heuristic for discovering good models.

8.5 PREMISE

To find a good pattern set in practice, we present PREMISE. Instead of enumerating all possible patterns, it efficiently explores the search space in a bottom-up heuristical fashion.

8.5.1 *Creating and Merging Patterns*

PREMISE starts with with an empty set of patterns M, the dataset is initially encoded only using singletons. It then iteratively improves

the model by adding, extending, and merging patterns until no more gain in the MDL score can be achieved. To ease the explanation, we will first introduce the setting with conjunctive patterns only.

Pairs of items for which the transaction sets barely overlap are unlikely to compress well as conjunctive patterns. Hence, we introduce a minimum overlap threshold of 0.05 in all experiments, to speed up the search by pruning infrequent and therewith uninteresting patterns. This straightforwardly leads to algorithm `createCandidates` that, based on a current model M, outputs a set of possible candidate patterns that we will consider as additions to the model. We give the full pseudo-code in Section 8.5.4 below.

8.5.2 *Filtering Noise*

Additionally to the MDL score, Fischer and Vreeken (2020) proposed to use Fisher's exact test as a filter for spurious patterns. Here, we use it to test our candidate patterns. Fisher's exact test allows to assess statistically whether two items co-occur independently based on contingency tables. We assume the hypothesis of homogeneity; in our case that there is no difference in the pattern's probability between D^- and D^+. Fisher showed that the values of the contingency table follow a hypergeometric distribution (Fisher, 1922). We can then compute the p-value for the one-sided test directly via

$$p = \sum_{i=0}^{\min(a,d)} \frac{\binom{a+b}{a-i}\binom{c+d}{c+i}}{\binom{n}{a+c}}, \tag{8.8}$$

with $c = |\sigma_P(D)|$, $a = |D| - c$, $d = |\sigma_P(D^+)|$, $b = |D^+| - d$ and $n = |D|$ for a pattern P labeled with l_+. For patterns labeled with l_-, the other tail of the distribution is tested (with a and b as well as c and d switching places). A general problem for statistical pattern mining is the lack of an appropriate multiple test correction. We here however only use the test to *filter* candidates, false positive patterns passing the test are still evaluated in terms of MDL.

8.5.3 *The* Premise *Algorithm*

Combining the candidate generation and the MDL score from Section 8.4, we obtain Premise. We give the pseudo-code in Algorithm 1. Starting with the empty model, we generate candidates, and for each of those, we compute the (negative-valued) gain in terms of MDL (line 6) as well as the pattern's p-value (line 7). We select the candidate below a significance threshold α that reaches the best gain (line 8-10) and add it to the model. If we created the pattern through a merge, we remove its parent patterns from M. We repeat the process until no candidate provides further gain in codelength.

Algorithm 1: PREMISE

input: D with D^- and D^+, significance threshold α
output: Heuristic approximation M of M^*

```
do
    Δ' ← 0;
    M' ← M;
    C ← createCandidates(M);
    for P ∈ C do
        Δ ← L(D, M ⊕ P) − L(D, M) ;        // (negative) gain
        p ← FisherExactTest(P);             // p-value
        if p < α and Δ < Δ' then
            Δ' ← Δ;
            M' ← M ⊕ P;
        end
    end
    M ← M';
while Δ' < 0;
return M
```

8.5.4 *Mutual Exclusivity*

In all practical applications that we consider, which are from the NLP domain, we are interested in finding clauses expressing words that are synonyms, that reflect similar concepts, or language variations, such as $\bigwedge$(*which,*$\bigotimes$(*color, colour*)) or $\bigotimes$(*could, can*). Such statements, however, require a richer pattern language than given by the purely conjunctive patterns discovered by the state-of-the-art. We discussed above how to identify the best model over such a richer pattern language of clauses in terms of MDL.

For NLP applications, instead of enumerating all possible clauses exhaustively, we follow a more informed approach, taking into account information from pre-trained word embeddings. We are interested in finding words that are synonyms or that reflect similar concepts, such as $\bigotimes$(*color, colour*) or $\bigotimes$(*could, can*). Research in NLP has proposed various techniques for identifying such pairs including manually created ontologies such as WordNet (Miller, 1995) or word embeddings that are learned through co-occurrences in text and map words to vector representations. This information about related words can be used to guide the search for mutually exclusive patterns. Using such pretrained embeddings rather than deriving them from the given input data has the advantage that we are independent of the size of the input data set, and receive reliable embeddings, which were trained on very large, domain independent text corpora.

While our approach is independent of the specific method, we have chosen FastText word embeddings trained on CommonCrawl and

Word	5-nearest neighborhood
photo	photograph, photos, picture, pic, pictures
color	colour, colors, purple, colored, gray
can	could, will, may, might, able
say	know, think, tell, mean, want

Table 8.1: Words and their nearest neighbors on *Visual7W*.

Wikipedia (Grave et al., 2018b). In contrast to word ontologies, word embeddings have a broader vocabulary coverage. They also do not impose strict restrictions such as a particular definition of synonyms and instead reflect relatedness concepts learned from the text. FastText embeddings have the additional benefit that they use subword information, removing the issue of out-of-vocabulary words. The word embeddings are independent of the machine learning classifier we study. As measure of relatedness m between two items I_1, I_2, we use cosine similarity, i.e. $m = cos(emb(I_1), emb(I_2))$ where emb is the mapping between an item/word and its vector representation. We define $nb(I,k)$ as the $I' \in \mathcal{I}$ for which $m(I,I')$ is the k-highest. Examples for words and their neighbours in FastText embeddings are given in Table 8.1.

Based on the information of the embedding, we derive $\otimes$-clauses. For each item I, we explore mutual exclusivity in its $1 \dots K$ closest neighbors, i.e. from $\otimes(I, nb(I,1))$ until $\otimes(I, nb(I,1), \dots, nb(I,K))$ where K is the maximum neighborhood size. For that, we adapt the `createCandidates` algorithm from Section 8.5.1 so that whenever we consider merging with an item I, we also consider merging with the $\otimes$-clauses containing additionally the $1, 2, \dots K$ closest neighbours. We give the full pseudo-code in Algorithm 2.

Since not all words have K neighbors that represent similar words, we additionally filter neighbourhoods such that $\frac{\bigcap_I \sigma_I(D)}{\bigcup_I \sigma_I(D)} < a$ and $m(I, nb(I,k)) > b_k$ for all items I in the clause, i.e. we require that their transactions barely overlap (mutual exclusivity), and that their embeddings are reasonably close. In all experiments we set $K = 5$, $a = 0.05$ and b_k to the 3rd quartile of $\{m(I, nb(I,k)) \mid I \in \mathcal{I}\}$.

In the general case for arbitrary labeled data, we could follow the proposal of Fischer and Vreeken (2020) to search for potential XOR structure, which however would lead to a much increased search space and hence computational costs, without any benefits for the specific NLP applications.

Algorithm 2: createCandidates

input: D, set of patterns $\mathcal{P}$ from the current M, max neighbour distance K
output: Set of candidate patterns $\mathcal{P}$
// Define $nb(I, 0) = I$ for simplicity
$C \leftarrow \{\}$;
// Single item and its neighbours
foreach $I \in \mathcal{I}$ **do**
  $A \leftarrow \{\}$;
  foreach $k \in \{0, \ldots, K\}$ **do**
    $A \leftarrow A \cup \{nb(I, k)\}$;
    $C \leftarrow C \cup \{\otimes(A)\}$;
  end
end
// Pairs of items and their neighbours
foreach $(I_1, I_2) \in \mathcal{I} \times \mathcal{I}$ **do**
  $A_1 \leftarrow \{\}$;
  foreach $k_1 \in \{0, \ldots, K\}$ **do**
    $A_1 \leftarrow A_1 \cup \{nb(I_1, k_1)\}$;
    $A_2 \leftarrow \{\}$;
    foreach $k_2 \in \{0, \ldots, K\}$ **do**
      $A_2 \leftarrow A_2 \cup \{nb(I_2, k_2)\}$;
      $C \leftarrow C \cup \{\oslash(\otimes(A_1), \otimes(A_2))\}$;
    end
  end
end
// Pattern + item and its neighbours
foreach P in $\mathcal{P}$ **do**
  foreach $I \in \mathcal{I}$ **do**
    $A \leftarrow \{\}$;
    foreach $k \in \{0, \ldots, K\}$ **do**
      $A \leftarrow A \cup \{nb(I, k)\}$;
      $C \leftarrow C \cup \{\oslash(\gamma(P) \cup \{A\})\}$;
    end
  end
end
// Pattern + Pattern
foreach $(P_1, P_2) \in \mathcal{P} \times \mathcal{P}$ **do**
  $C \leftarrow C \cup \{\oslash(\gamma(P_1) \cup \gamma(P_2)\}$;
end
/* see Sections 8.4 and 8.5 for filter criteria */
$C \leftarrow$ `Filter`(C) ;
return C

8.5.5 *Complexity*

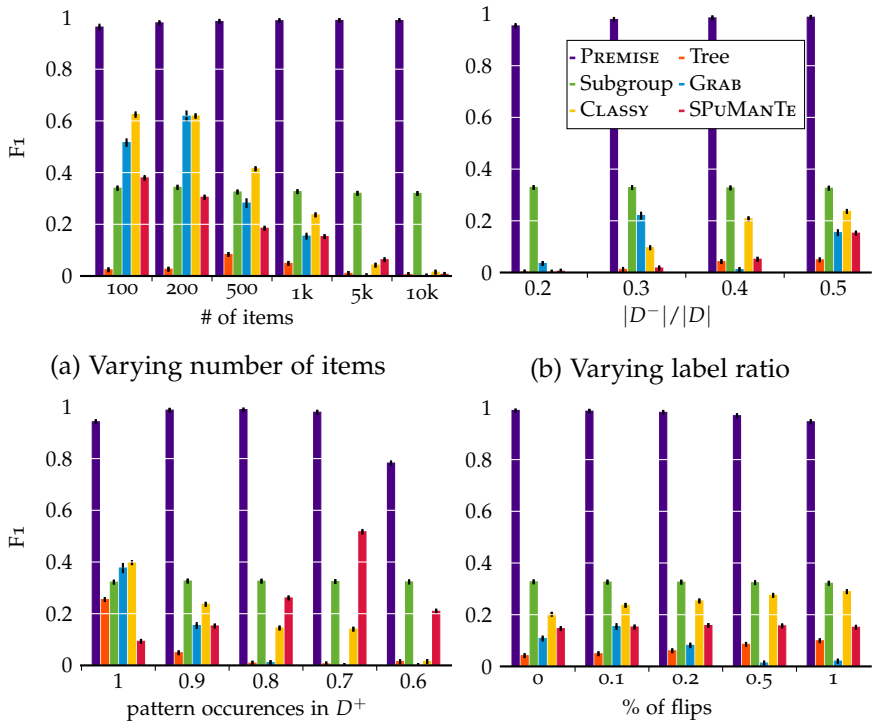
While it is common to consider the complexity in terms of the size of the input, the bound it would give – which is exponential in the number of items as discussed in the theory section – is neither helpful nor tight considering the discovery of small models. As MDL ensures the discovery of such small models, we thus analyze the complexity of PREMISE in terms of the size of the discovered model. Consider PREMISE finds k conjunctive patterns of maximum length l for a dataset with m items. Since in every round either a new singleton or pair is generated that belongs to one of the k final patterns, or two existing patterns are merged, the algorithm runs for $O(k\,l)$ rounds. In each round, the dominating factor is the candidate generation, out of which there are $O(m)$ potential singletons, $O(m^2)$ pairs, and at maximum $O(k\,l)$ pattern merges, corresponding to the case that all parts of the final patterns exist as singleton patterns in the current round. Hence, we get a worst case time complexity of $O(k\,l\,(k\,l + m^2))$.

For clauses containing mutual exclusivity, for all practical applications we consider XOR statements of the c closest words in a given embedding, where c is a small constant. We hence consider $O(m\,c)$ single XOR clauses, $O((m\,c)^2)$ pairs, and at maximum $O(k\,l)$ pattern merges, where again this corresponds to the case that all parts of the final patterns exist as singleton patterns in the current round. Hence we get a worst case time complexity of $O(k\,l\,(k\,l + (m\,c)^2))$. For the general case, when searching for arbitrary AND and XOR combinations, we refer to the work by Fischer and Vreeken (2020).

8.6 EXPERIMENTS

We evaluate and compare our approach on synthetic data with known ground truth, as well as on real world NLP tasks to characterise misclassifications. Describing a labeled database in terms of its features has been studied extensively in various fields. We here compare against the state-of-the-art from each, in particular, subgroup discovery using weighted relative accuracy as quality function (Atzmueller, 2015), significant pattern mining (SPuMANTE by Pellegrina et al. (2019)), rule sets mining (GRABby Fischer and Vreeken (2019)), and rule lists (CLASSY by Proença and van Leeuwen (2020)). As baseline and representative of interpretable machine learning models we consider patterns derived from classification trees. Due to runtime issues, we compare to ANCHORS (Ribeiro et al., 2018) only in the NER experiment. For similar reason, we exclude SLICEFINDER (Chung et al., 2020), and disjunctive emerging patterns (Vimieiro, 2012); neither completed a single run within 12 hours.

Experiments were performed on an Intel i7-7700 machine with 31GB RAM running Linux. All synthetic data experiments finished within



(a) Varying number of items

(b) Varying label ratio

(c) Varying label shift

(d) Varying background noise

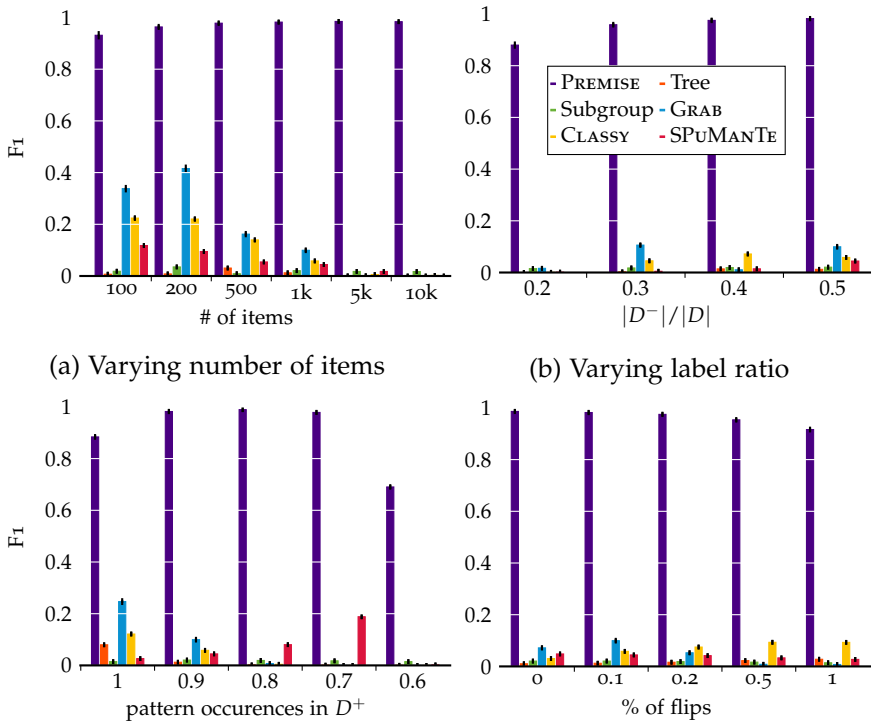
Figure 8.3: *Synthetic data results (soft F1 score).* We visualize results on synthetic data with varying number of items (a), label ratio (b), label shift (c) and amount of background noise (d). The results are in terms of a soft F1 score with respect to the ground truth, which also rewards the discovery of fragments of patterns, as defined in the text.

minutes for the moderately sized data sets, and within hours for the larger datasets with 5k and 10k items. On the VQA datasets PREMISE finished within 20 minutes and on the NER data within 4 hours. We make our code, the data as well as the full list of patterns found by the different methods publicly available.[2]

8.6.1 *Synthetic Data*

We evaluate all methods with respect to scalability, robustness to noise, label imbalance, and the conditional probabilities of patterns. A standard metric to evaluate success of a model is the F1 score – the harmonic mean between precision and recall – which for discovered pattern set P_d and ground truth pattern set P_g is defined as $\mathrm{F1}(P_d, P_g) = |P_d \cap P_g| / \left(|P_d \cap P_g| + \frac{1}{2}|P_d \Delta P_g|\right)$, where Δ is the symmetric difference between two sets. As competitors only recover fragments of patterns and hence they obtain very low F1 scores (see Figure 8.4), for the analysis, we use a soft F1 score that rewards also fragments.

2 https://github.com/uds-lsv/premise



(a) Varying number of items

(b) Varying label ratio

(c) Varying label shift

(d) Varying background noise

Figure 8.4: *Synthetic data results (original F1 score).* We visualize results on synthetic data with varying number of items (a), label ratio (b), label shift (c) and amount of background noise (d). The results are in terms of F1 score with respect to the ground truth.

We define it as harmonic mean between a soft precision and a soft recall:

$$\text{SoftPrec}(P_d, P_g) = \sum_{p_d \in P_d} \operatorname*{argmax}_{p_g \in P_g} \frac{|p_d \cap p_g|}{|p_g|} , \tag{8.9}$$

$$\text{SoftRec}(P_d, P_g) = \sum_{p_g \in P_g} \operatorname*{argmax}_{p_d \in P_d} \frac{|p_d \cap p_g|}{|p_g|} , \tag{8.10}$$

$$\text{F1}(P_d, P_g) = \frac{2 * \text{SoftPrec} * \text{SoftRec}}{\text{SoftPrec} + \text{SoftRec}} . \tag{8.11}$$

Unless specified differently, for each of the experiments we generate a data matrix with 10 000 samples, half of which get label l_+, the other half l_-. The set of items $\mathcal{I}$ has size 1000. We draw patterns of length $2-5$ from $\mathcal{I}$ with replacement until 50% of items are covered. For each pattern we then draw $k \sim \mathcal{N}(150, 20)$ and set the items of the pattern in $.9k$ random transactions from D^+, and $.1k$ transaction from D^- to 1. This corresponds to a typical sparsity level for pattern mining problems. Additionally, for each item that is part of a pattern, we let it occur in $k \sim \mathcal{N}(50, 20)$ random transactions from D. For all items not part of a pattern, we let them occur in $k \sim \mathcal{N}(150, 20)$ transactions

from D. Lastly, we introduce background noise by flipping .1% of the matrix values.

Scalability. First, to investigate how the different methods scale to larger item sets $\mathcal{I}$, we vary the number of items in $\{100, 200, 500, 1000, 5000, 10\,000\}$. We observe that that the performance of most existing methods detoriates already for data with several hundred items, only PREMISE and subgroup discovery are robust when it comes to scalabity (see Fig. 8.3a). Note that we let subgroup discovery retrieve the top k patterns, where k equals the number of ground truth patterns, it hence has an advantage over all other approaches which would not hold in a real-world scenario. Subgroup discovery, however, still only yields (soft) F1 scores of around .35, whereas PREMISE recovers ground truth patterns with scores close to 1. Moreover, we see that all MDL based methods outperform the other approaches for data of smaller dimension.

Label imbalancing. To investigate the effect of label imbalancing, we vary the proportion of transactions having label l_-, i.e. $|D^-|/|D| \in \{0.2, 0.3, 0.4, 0.5\}$. An imbalancing of labels is commonly encountered in real world datasets, where e.g. the number of misclassified samples makes up only a small fraction of overall samples. We again find that only subgroup discovery and PREMISE are robust and perform consistently across the varying levels, with subgroup discovery yielding a soft F1 score of .35 and PREMISE close to 1 (see Fig. 8.3b). All other methods break down when there is a minor imbalance in the data.

Varying label shift. Next, we look at label shift, the effect of patterns occurring not exclusively in one of the labels. This is again a likely event in real world data. We adapt the occurrence of patterns between 1, meaning the pattern exclusive occurs in one partition of the database, to .6, meaning that 60% of the transaction where a pattern occurs have one label, the others have the other label. Similar to before, we observe the subgroup discovery and PREMISE are robust to this change (see Fig. 8.3c). For the statistical testing based SPUMANTE, we find the best performance for a label shift of .7 rather than 1. For CLASSY and especially GRAB, performance drops even for slight label shifts.

Robustness to background noise. Finally, we look at how the methods cope with background noise, by flipping a fraction of $\epsilon \in \{0, 0.001, 0.002, 0.005, 0.01\}$ entries of the data matrix. We find that CLASSY, PREMISE, SPUMANTE, and Subgroup Discovery are robust even to large amount of noise (see Fig. 8.3d).

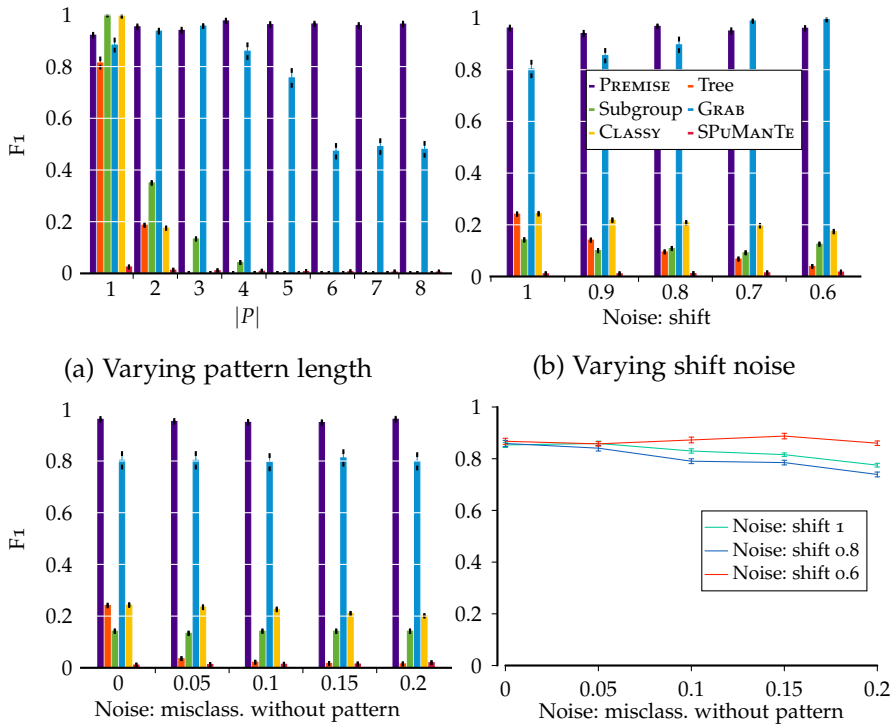
8.6.2 *Synthetic Text Data Experiments*

Before experimenting with real-world scenarios, we also evaluate how well all methods cope with item – or token – distributions similar to real text. To obtain such data, we derive transactions/instances from the around 3.4k sentences in the development set of the PennTreebank

Corpus (Marcus et al., 1993). In particular, we draw 12 distinct patterns, for each pattern choosing items from the vocabulary tokens at random. To ensure that we introduce only new patterns into the data, we verify that none of the items in the patterns co-occur in the original data. We then insert each pattern into a random subset of the PennTreebank instances, where the number of instances to be covered is drawn from a normal $\mathcal{N}(150, 20)$. The data contains 6k unique items. To evaluate settings typical for classification, we then vary two types of noise. *Shift noise* indicates the percentage of instances with a pattern that are actually labeled as misclassifications, lower values mean that the model is still able to predict correctly in some of the instances – e.g. because a network leverages additional information in the data. The second type of noise is labeling instances as misclassification although there is no pattern occurrence – i.e. non-systematic errors – which we refer to as *label noise*. For all samples with pattern occurences, we label a fraction of those as misclassification according to the *shift noise*, and then introduce *label noise*.

EXPERIMENTAL SETUPS We generate four different sets of experiments. In the first set, we introduce conjunctive patterns varying pattern length of the introduced patterns between 1 and 8 without noise. In the second set of experiments we vary the amount of *shift noise*, introducing shifts of $\{0.6, 0.7, 0.8, 0.9, 1\}$, and choosing pattern length uniformly in 1 to 5. In the third set we instead change the amount *label noise*, varying in $\{0, 0.05, 0.1, 0.15, 0.2\}$. In the fourth set of experiments, we introduce patterns consisting of conjunctions of mutual exclusive itemsets. The number of clauses per pattern and the number of items for each clause is chosen uniformly at random between 1 and 5. A pattern is only added to an instance if this would not break the mutual exclusivity assumptions of all patterns. For the word neighborhoods, items in the same clause obtain embeddings located around a randomly chosen centroid. All other items obtain random embeddings. We repeat all experiments 10 times and report the original F1 score as average across repetitions.

RESULTS For the first experiment set of varying pattern length (Fig. 8.5a), we observe that subgroup discovery is able to retrieve short patterns well, failing however to discover any larger patterns, instead retrieving large sets of redundant patterns. Decision trees perform similarly due to overfitting, finding a plethora of highly redundant patterns. SPuMANTE, although based on statistical testing, consistently finds thousands of redundant patterns, performing worst of all in this regard. The rule set miner GRAB recovers small patterns well, it performs however much poorer in retrieving patterns of larger size. PREMISE is the only approach to consistently recover the ground truth in all data sets.



(a) Varying pattern length

(b) Varying shift noise

(c) Varying label noise

(d) Complex clauses

Figure 8.5: *Synthetic text data results (original F1 score).* On synthetic text data, varying the number of items per pattern (a), the amount of *shift noise* (b), and the amount of *label noise* (c), we visualize the results in terms of F1 score with respect to the ground truth for existing methods and PREMISE. We additionally provide the results of PREMISE on data containing patterns of mutual exclusive clauses for varying amounts of noise (d).

For both noise experiments, visualized in Fig. 8.5b and 8.5c, the tree based method completely breaks down already for moderate amounts of noise. Subgroup Discovery and SPuMANTE both perform consistently bad with F1 scores below .2. Out of the existing approaches, only GRAB is able to recover the ground truth well. PREMISE outperforms all existing methods in each of our noise experiments, achieving consistently high F1 scores beyond .92.

Since most baselines do not support discovering mutual exclusivity or proved to fail in the more simple setup of conjunctions, we only evaluate our proposed method on the fourth set of experiments with mutually exclusive items. We observe that PREMISE is still able to retrieve patterns even in this challenging setup of complex clauses, with F1 scores close to .9, and is able to discover clauses in the presence of noise (Fig. 8.5d).

Overall, these results show that existing methods are challenged by datasets of larger scale and label imbalancing. In contrast, PREMISE solves both these challenges also on data with distributions similar to natural language text.

8.6.3 *Real Data: VQA*

Visual question answering (VQA) is the task of answering textual questions about a given image. It is a popular and challenging task at the intersection of vision and natural language processing. In this section, we analyze the misclassification of Visual7W (Zhu et al., 2016) and LXMERT (Tan and Bansal, 2019), both specific architectures for different VQA tasks. The pretrained Visual7W reaches 54% accuracy in 4-option multiple choice, LXMERT a validation score of 70% on their minival split. Both classifiers perform far from optimal and thus serve as interesting applications for describing (misclassification) labels. Here, we derive misclassification data sets from application of the classifiers to the development sets.

In Table 8.2 we provide statistics about the data and retrieved patterns. Both the tree based method and SPuMANTE retrieve several hundred or thousand patterns making it difficult to interpret the results. Furthermore, we know from the previous experiments that these methods find thousands of patterns even when there exist only few ground truth patterns. The subgroup discovery approach requires the user to specify the number of patterns a-priori, which is not known, hence we search for the top 100 patterns to get a succinct set of patterns. For the retrieved results, there are some patterns showing reasons for misclassification. However, the patterns are highly redundant with often ten or more patterns expressing the same cause for misclassification. It is thus hard to get a full description of what goes wrong, it lacks the power of set mining approaches that evaluate patterns *together*. The majority of patterns found by CLASSY consist of only

		Visual7W	LXMERT
	$\lvert\mathcal{I}\rvert$	2429	5351
	$\lvert D\rvert$	28032	25994
PREMISE	k^-	29	41
	k^+	26	34
	$\overline{\lvert p\rvert}$	3.38	2.69
Tree	k	4309	3371
	$\overline{\lvert p\rvert}$	3.55	2.71
Subgroup	k	100	100
	$\overline{\lvert p\rvert}$	2.32	2.52
SPUMANTE	k	575	951
	$\overline{\lvert p\rvert}$	2.92	3.90
CLASSY	k	19	36
	$\overline{\lvert p\rvert}$	1.26	1.28
GRAB	k	1	1
	$\overline{\lvert p\rvert}$	1	1

Table 8.2: *VQA data statistics.* For the two VQA classifiers, we provide general statistics about data dimensions, and for each method the number of discovered patterns ($k = |P|$) or if applicable number of patterns explaining misclassification ($k^- = |P^-|$), respectively correct classification ($k^+ = |P^+|$). Additionally, we provide the average pattern length $\overline{|p|}$.

one token. GRAB fails to retrieve meaningful results, likely due to the heuristic search.

PREMISE is able to provide a succinct and non-redundant description of misclassification, and is the only method to recover patterns that reflect synonyms or different styles of writings. In Tab. 8.3a, we list patterns found by PREMISE for the Visual7W classifier, where we can clearly see the advantage of the richer pattern language, allowing to find patterns such as $\wedge$(*what*, $\otimes$(*color, colors, colour*)). Generally, the patterns found by PREMISE highlight different types of wrongly answered questions, including counting questions, identification of objects and their colors, spatial reasoning, and higher reasoning tasks like reading signs. Furthermore, PREMISE retrieves both frequent patterns, such as $\wedge$(*how, many*), rare patterns, such as $\wedge$(*on, wall, hanging*), and patterns containing related concepts and synonyms.

PREMISE also finds patterns explaining correct classification. These could indicate questions that are easy to answer but also questions that are just memorized by the network, since within the data set the answer is always the same. For instance, $\wedge$(*who, took,* $\otimes$(*photo, picture, pic, photos, photograph*)), although a difficult question, is nearly always answered by "photographer". Thus, patterns that are biased towards correct classification can indicate issues with the dataset. Another type of problematic questions is indicated by the pattern $\wedge$(*clock, time*), where usually the answer is "UNK", the actual time being replaced with the unknown word token by the limited vocabulary of Visual7W. The pattern hence indicates a setting where the VQA classifier undeservedly gets a good score.

By adding additional information as items to each instance, it is possible to gain further insights. Appending for example the correct output to each instance, we observe for the question when the picture was taken two different trends. On the one hand, the discovered pattern $\wedge$(*when,* $\otimes$(*daytime, nighttime*)) is associated with correct classification, the pattern $\wedge$(*when,* $\otimes$(*evening, morning, afternoon, lunchtime*)), on the other hand, points towards misclassification. This is intuitively consistent as the answers "daytime" and "nighttime" are easier to choose based on a picture.

For the LXMERT classifier, a similar set of patterns is discovered by PREMISE, with examples given in Table 8.3b. We observe that both classifiers share certain issues, like the counting questions. However, no patterns regarding color or spatial position are retrieved. This seems to indicate that the more recent LXMERT classifier can handle these better. Instead, many patterns indicate settings that require advanced capabilities like noticing fine-grained details or reading text on the images.

For the considered VQA classifiers, existing methods do not give succinct descriptions, and can not handle the richer language over conjunctions and mutual exclusivity. Such a language, however, pro-

pattern	example
UNK	how are the UNK covered
⋀(*how, many*)	how many elephants are there
⋀(*what*, ⊗(*color, colors, colour*))	what color is the bench
⋀(*on, top, of*)	what is on the top of the cake
⋀(*left, to*)	what can be seen to the left
⋀(*on, wall, hanging*)	what is hanging on the wall
⋀(*how, does, look*)	how does the woman look
⋀(*what, does*, ⊗(*say, like, think, know, want*))	what does the sign say

(a) Visual7W

pattern	example
⋀(*How, many*)	How many kites are flying?
⋀(*hanging, from*)	What is hanging from a hook?
⋀(⊗(*kind, sort*), of)	What kind of birds are these?
⋀(⊗(*would, could, might, can*), *you*)	How would you describe the decor?
⋀(*name, of*)	What is the name of this restaurant?
number	What is the pitchers number?
⊗(*letter, letters*)	What letter appears on the box?
⋀(*How, much*, ⊗(*cost, costs*))	How much does the fruit cost?

(b) LXMERT

Table 8.3: *VQA example patterns.* Our method discovers meaningful and easily interpretable patterns. For Visual7W (top) and LXMERT (bottom), we show a subset of the patterns highlighting different reasons for misclassification along with examples from the corresponding datasets. The full list of retrieved patterns for all methods is given in the additional material.

vides a deeper understanding of misclassification errors. In contrast, the patterns discovered by PREMISE give intuitive and informative descriptions over such a richer language that allow to understand both, the issues of the dataset, as well as limits of the VQA classifier and hence can be used to improve classifier performance.

8.6.4 *Real Data: NER*

A machine learning classifier might perform well on the training and test data, its performance when deployed "in the wild" however is often much worse. Understanding the difference between the restricted training set-up and the open application of the classifier in real-life is important for being able to improve. Here, we investigate the popular LSTM+CNN+CRF architecture (Ma and Hovy, 2016a) for named entity recognition. The classifier is trained on the standard NER dataset CoNLL03 (Tjong Kim Sang and De Meulder, 2003), where it achieves an F1-score of 0.93, a performance good enough for production settings. When evaluated on OntoNotes (Weischedel et al., 2011), a dataset covering a wider range of topics, the performance drops to 0.61 F1 on the development set. We evaluate on this split of the data consisting of 16k sentences and 23k unique items.

Anchor (Ribeiro et al., 2018) allows to obtain conjunctive patterns to explain NLP instances locally. It took, however, several days to analyze all misclassifications on modern GPU hardware due to the necessary, repeated queries to the NER classifier. Anchor finds 4.1k unique patterns with many redundant and overly long and specific patterns. PREMISE retrieves a concise set of 190 patterns. An example is ⩓(*-LRB-*, *-RRB-*) that indicates different preprocessing of the text, where *-LRB-* is an alternative form for the opening bracket, which is specific to the OntoNotes data, and thus should be properly handled by the NER classifier. Patterns also indicate problems with differing labeling conventions. For example, we find the patterns ⩓(*'s*) and ⩓(*Wall, Street*), which turn out to be handled differently for entities in OntoNotes, respectively CoNLL03. Apart from patterns that highlight dataset differences, we can also isolate issues with OntoNotes alone, which contains bible excerpts that are not labeled at all. We discover this through several patterns that describe this domain (*God*, *Jesus*, *Samuel*).

To empirically validate that the found patterns affect the classifier's performance, we select the top 50 patterns according to gain in MDL and for each pattern sample 5 sentences containing it uniformly at random from the OntoNotes training data. The CoNLL03 classifier is then fine-tuned on this data. Sampling and fine-tuning is repeated 20 times with different seeds. Using the pattern-guided data, the performance is improved to 0.67 mean F1 score (SE 0.003) compared to sampling fully at random where only a small improvement to

0.62 (SE 0.005) is achieved. This shows that the patterns discovered by PREMISE provide actionable insights into how a classifier can be improved.

8.6.5 *Experimental Details*

For the decision tree, patterns are exracted from a tree trained on the misclassification data. Each of the tree's inner nodes is a binary decision regarding the presence of an item and a pattern is the conjunctive path from the tree's root to one of its leafs. The model is trained with Gini impurity as decision criterion in the implementation from scikit-learn.

For the subgroup discovery, the implementation by Lemmerich and Becker (2018) is used with depth-first search. The size of the result set and the maximum depth are set to the ground truth for the synthetic data and to 100 and 5 respectively for the VQA datasets. SPuManTe is used with the authors' default parameters, setting its sample size to the dataset size. For GRAB, we use the publicly available implementation by the authors, which we tailored for the task at hand by restricting the possible rule-heads to the labels only, but allowing tails over all other items. For CLASSY, we used the publicly available implementation by the authors as used in the original publication. Minimum support is set to 1 and maximum rule length to the ground truth for the synthetic data and 5 for the VQA datasets.

For Visual7W and LXMERT, we use the published, pretrained models by the corresponding authors. For the LSTM+CNN+CRF classifier for NER, we follow the specific set-up from Chapter 6 with English FastText embeddings. For OntoNotes, the data split by Pradhan et al. Pradhan et al. (2013) is used. The fine-tuning data consists of 240 instances/sentences as two patterns did not match any training data. Fine-tuning on the additional data is performed for 30 epochs. As labels, the intersection between CoNLL03 and OntoNotes is used (PER, LOC, ORG) in the BIO2 format.

8.7 DISCUSSION

Experiments show that PREMISE provides concise and interpretable descriptions of labeled data. On synthetic data we find that the state-of-the-art methods across different fields related to supervised pattern mining, including subgroup discovery, emerging pattern mining, statistical pattern mining, rule mining, and tree based classification, all have severe difficulties finding the ground truth pattern set, while PREMISE accurately retrieves it. Moreover, we observe that PREMISE is the only approach that is at the same time robust to noise, label imbalance, and easily scaling to thousands of items. It thus renders itself as the most suitable method for challenging real world applications

revolving around characterising misclassifications of NLP models. For such tasks, the labels are inherently imbalanced and the sets of items – in this case tokens – is large. Besides, to capture the rich structures of word associations, such as synonyms or language variations, we need a richer pattern language capturing mutual exclusiveness, which only PREMISE is able to express.

On two widely adopted NLP models for visual question answering, we set for characterising their misclassifications comparing PREMISE against existing approaches. While some of the competing methods did retrieve reasonable explanations, these were highly redundant and barely interpretable for human experts with several hundred or thousand patterns. Moreover, important concepts, such as patterns that are similar across related words or synonyms, are completely missed. PREMISE, on the other hand, discovers succinct sets of patterns that provide interesting characterizations of classification errors, revealing that models struggle with counting, spatial orientation, reading, and identifies shortcomings in training data.

To show that the pattern sets are not only interpretable and give interesting insights, but also actionable, we analyze a popular classifier for named entity recognition. In particular, we consider a model applied to text of a different source and characterize the resulting classification errors with PREMISE, and compare it with the recent local explanation method ANCHORS. While PREMISE is able to retrieve a pattern set swiftly in few hours on commodity hardware, ANCHORS requires several days on a modern GPU to deliver results. Inspecting the retrieved patterns confirms that also for NER models PREMISE is able to retrieve meaningful patterns explaining misclassification, while ANCHORS finds a very large set of overly long and redundant patterns. Furthermore, as expected from a local method, the patterns are highly specific and thus identify problems of the model for particular instances rather than identifying the general issues that the model has.

8.8 PROOF: ORDER OF ITEMS

Here, we provide a proof that the codelength is independent on the order of items in mutual exclusive clauses. The proof closely follows that of Fischer & Vreeken Fischer and Vreeken (2020).

Given a clause $cl = \otimes(i, j, k)$ with corresponding margins n_i, n_j, n_k, it does not matter in which order we transmit where the items hold.

We show that we can flip the item order without changing the cost. Assume a new order $P = \bigotimes(k, i, j)$, then we show

$$\log \binom{n}{n_i} + \log \binom{n - n_i}{n_j} + \log \binom{n - n_i - n_j}{n_k} \tag{8.12}$$

$$\stackrel{!}{=} \log \binom{n}{n_k} + \log \binom{n - n_k}{n_i} + \log \binom{n - n_i - n_k}{n_j} . \tag{8.13}$$

With the definition of the binomial using factorials and standard math, adding new terms that add up to 0, we show that the above equation hold.

$$\log \frac{n!}{(n - n_i)! n_i!} + \log \frac{(n - n_i)!}{(n - n_i - n_j)! n_j!} \tag{8.14}$$

$$+ \log \frac{(n - n_i - n_j)!}{(n - n_i - n_j - n_k)! n_k!} \tag{8.15}$$

$$= \log(n!) - \cancel{\log((n - n_i)!)} - \log(n_i!) + \cancel{\log((n - n_i)!)} \tag{8.16}$$

$$- \cancel{\log((n - n_i - n_j)!)} - \log(n_j!) + \cancel{\log((n - n_i - n_j)!)} \tag{8.17}$$

$$- \log((n - n_i - n_j - n_k)!) - \log(n_k!) \tag{8.18}$$

$$\underbrace{+ \log((n - n_k)!) - \log((n - n_k)!)}_{=0} \tag{8.19}$$

$$\underbrace{+ \log((n - n_i - n_k)!) - \log((n - n_i - n_k)!)}_{=0} \tag{8.20}$$

$$= \log \frac{n!}{(n - n_k)! n_k!} + \log \frac{(n - n_k)!}{(n - n_i - n_k)! n_i!} \tag{8.21}$$

$$+ \log \frac{(n - n_i - n_k)!}{(n - n_i - n_j - n_k)! n_j!} . \tag{8.22}$$

Other permutations and larger clauses follow the same reasoning.

8.9 CONCLUSION

We considered the problem of finding interpretable and succinct descriptions of a given label, and proposed to discover succinct pattern sets to describe the labels based on the Minimum Description Length Principle. To solve this formulation in practice, we formulated an efficient bottom-up heuristic PREMISE. While this problem has been studied extensively, PREMISE showed to be the only approach that scales well to data typical in real world problem settings, while at the same time being robust to noise, and label imbalance. With these abilities, combined with a more expressive pattern language compared to the state-of-the-art capturing also mutual exclusive relationship, PREMISE discovered succinct, informative, and actionable pattern sets that characterize misclassifications of NLP models in two challenging settings, which capture general problems of the model rather than instance specific (local) issues. It hence fills the gap of a robust approach

to describe labels in terms of human-interpretable patterns, suited to take on problems such as characterizing misclassifications of deep NLP models.

While our approach scales already to tens of thousands of features, it makes for engaging future work to scale it even further towards hundreds of thousands of features, which would make it applicable other domains, like large-scale biomedical data considering patients vs. healthy individuals, or to extend the work on characterizing misclassifications incorporating elements of the classifier itself, such as neuron activations.

We performed the evaluation for high-resource, black-box NLP models. Low-resource settings bring additional challenges like validation sets of limited size that could make it more difficult to find statistically significant error patterns. On the other hand, the decision processes of weak supervision systems are often easier to understand than that of large, black-box neural networks. This could make it possible for future work to automatically identify the parts within the weak supervision system that cause the error patterns we find with PREMISE.

9 CONCLUSION & FUTURE WORK

This chapter summarizes the results of this thesis and gives some suggestions for future directions in this field.

9.1 SUMMARY

Advances in machine learning and deep learning have transformed the landscape of natural language processing in recent years. In Chapter 3, we saw, however, that many of the advances require large amounts of labeled data and are, therefore, limited to a selected set of languages. There exists a large gap in NLP support between these high-resource languages and many low-resource languages with millions of speakers. Closing this gap is essential to allow speakers of these languages participation in the digital world. Similar to languages, the data for many tasks is also restricted to specific domains, even in English. Lowering the data requirements could enable more small businesses or individuals to set up NLP applications of their own.

9.1.1 *The State of Low-Resource NLP*

The survey in Chapter 3 gave a structured overview of the methods that have been proposed to handle low-resource scenarios. Based on this, we argued that there is no fixed definition for what a low-resource setting is. Instead, one needs to consider multiple dimensions of data availability, namely labeled, unlabeled and auxiliary data. The latter is especially relevant, as most low-resource approaches assume the availability of specific types of auxiliary resources. We identified as open issues, among others, the necessity for realistic evaluations to verify if these assumptions are met in real-life.

9.1.2 *The Weak Supervision Pipeline*

As we saw in the survey, weak supervision is one of the most popular methods to approach resource-lean scenarios. Instead of expensive, manually-labeled data, one uses (semi-)automatic annotation processes. While this kind of supervision is cheaper and quicker to obtain, the rate of incorrect labels is usually also much higher. This label noise can hurt performance during training and label noise modeling has been proposed in the past to overcome these negative effects of the weak supervision. In Chapter 4, we presented a pipeline for NER with weak supervision. It started with the distant supervision obtained

through matching external entity lists, included the modeling of the label noise and concluded with the evaluation of the base model on clean test data.

While many works on noise handling assume that only noisily labeled data is available, we argued that it is often comparatively easy also to obtain a minimal amount of clean supervision, e.g., when a test set is annotated. We, therefore, proposed a new method to leverage the combination of a small amount of clean with a large amount of noisy label data. The model is trained on both types of data while integrating a noise model that is estimated based on pairs of clean and noisy labels.

When testing the weak supervision pipeline, we could see the importance of evaluating across different levels of resource availability. The base model without noise handling, e.g., developed certain noise robustness once it had enough access to clean data. We also saw the effectiveness of the combination of weakly supervised data and noise handling. This approach reached the same performance as the base model with half as much clean data.

9.1.3 *A Tool for Distant Supervision*

Weak and distant supervision can only work in practice if the user can employ it efficiently. To this aim, in Chapter 5, we presented ANEA, a tool to automatically extract entity lists from Wikidata and annotate large amounts of unlabeled text with it. While it minimizes the manual effort for the core annotation process, it also helps the human experts in understanding errors of the annotation and allows them to improve it. This control over the process is given by heuristics that can be enabled and configured. The graphical interface we developed makes our tool accessible to users without programming knowledge. We verified the effectiveness of the distant supervision provided by ANEA for specific domains like movies as well as low-resource languages like Estonian or Yorúbà. This tool was also used to generate distantly supervised data in the following chapters.

9.1.4 *Low-Resource Techniques Meet Pre-Trained Language Models*

Pre-trained language models like BERT have rapidly become an essential part of many NLP architectures. Past work indicated, however, that there could be a gap in performance when using these language models for high and low-resource languages. Also, it was shown that weak supervision might underperform when applied to actual resource lean scenarios compared to simulated ones. Taking the distant supervision and noise handling pipeline from previous chapters, we evaluated in Chapter 6 how they combine with modern pre-trained language model architectures as base models. This was tested on

three actual low-resource languages from African language families (Hausa, isiXhosa and Yorúbà) and two different tasks (NER and text classification).

We also experimented with combining BERT-like models with another popular low-resource technique, namely few-shot transfer learning. We saw that both approaches could help when only limited amounts of clean data are available. With few-shot transfer learning, we obtained the most impressive results, reaching in several cases with only ten labeled sentences in the target language the same performance as the baseline trained on several hundred labeled sentences. Few-shot learning showed, however, also its limitations as it was struggling when the label sets did not match for the transfer.

Distant supervision was helpful when limited amounts of clean data were available. Once we used more clean labels, the additional, weak and noisy supervision actually hurt performance. This indicated that the question of noise handling is far from solved and further research in noise handling methods is necessary to boost performance in these cases.

We also used these settings to reflect on how realistic our evaluations were. One assumption was, e.g., the availability of large GPU hardware. In a low-resource scenario, access to large hardware resources might also be limited. In experiments, we saw that the gap between normal and distilled models (that can run on limited hardware) can be larger in settings with limited amounts of labeled data. One of the other issues we highlighted is the annotation time on which we will expand in the future work section below.

9.1.5 *Noise Model Estimation on Realistic Noise*

Chapter 7 aimed at bridging the practical evaluation with a more theoretical analysis. We revisited the noise modeling approach and derived the expected error of the noise model estimated from pairs of clean and noisy labels. This gave insights into the factors on which the noise estimation depends, such as the noise distribution or the data sampling method. We verified these results on synthetic and realistic noise as well as empirically showing the connection between noise model estimation and downstream performance of the base model.

We highlighted the differences in the noise distribution when comparing common synthetic noises (like single flip or uniform) with realistic noise sources (both from others and from our work). Since we previously showed the importance of the noise distribution on the noise handling method, it is, therefore, also essential to evaluate new methods on realistic forms of noisy labels. For this aim, we presented the NoisyNER dataset. It was created through a distant supervision process and provides multiple different noise levels for the same features. It also includes additional, challenging factors like uneven label

distributions and noise classes with higher probability than the true class.

9.1.6 *Understanding The Reasons for Label Errors*

To improve a weak supervision system, or a machine learning classifier in general, it is important to understand where they make labeling errors. In Chapter 8, we presented a novel approach to characterizing errors of a black-box classifier. Approaching this task from a data mining perspective, we proposed a new method for finding label-descriptive patterns. Instead of local explanations, our Premise algorithm identifies a global set of patterns. We developed an encoding that describes the data using the Minimum Description Length principle. This gave us pattern sets that are concise and statistically significant. We extended the pattern language to contain not only the typical conjunctions but also mutual exclusivity, allowing to model natural language phenomena like synonyms or related words.

Experiments on synthetic data showed that Premise outperformed the state-of-the-art competitors, scaled to large datasets and was robust against several types of noise. Applied to the misclassifications of Visual Question Answering and NER classifiers, we saw that our new method could handle the large search spaces of natural language text inputs and could find a compact set of human-interpretable and actionable insights.

9.2 CONCLUSION

With the growing need for labeled machine learning data, weak supervision has become a popular solution, moving from the academic setting into real-world usage in start-ups and industry projects, e.g. by Bach et al. (2019) and Snorkel AI (2022). In this dissertation, we studied how weak supervision can be used in low-resource settings with a focus on resource-lean languages. We showed that with the right software support, weak supervision can be an efficient way to translate expert insights into labeled data. It is orthogonal to other approaches like transfer learning and, therefore, a useful contribution to the toolbox of low-resource methods.

Incorrect labels are a crucial aspect of weak supervision. We developed a method to make reasons for misclassification easily understandable and studied how noise modeling can remove the negative effects of these label errors on the training process. We also highlighted, however, where weak supervision and noise handling still fall short and why realistic evaluations are necessary to ensure they can impact real-world applications.

9.3 FUTURE DIRECTIONS

One aspect that is seldomly considered in the field is the human expert. Often the expert is just mentioned as an abstract motivation for why weak supervision is needed. When one aims at developing methods that are actually useful in practice, the human element is a part of the system that one should not ignore.

In Chapter 6, we saw that for text classification, weak supervision needed a set-up of 2.5 hours. One of our annotators could annotate over thousand sentences in the same time span. This makes a standard annotation a competitive baseline. In contrast, for NER, setting up the distant supervision took only ca. 30 minutes thanks to the support of the ANEA tool. In the same time, an annotator would only have annotated 30 NER sentences. Additionally, the distant supervision could then be applied to a nearly arbitrary amount of unlabeled text. This shows that it is essential to take the annotating expert into consideration. This includes human factors like willingness to annotate, expertise, personal preferences and mental load. Such research questions connect weak supervision with the field of human-computer-interaction. The goal here is to develop tools that better support the human expert.

From a more general perspective, weak supervision can be seen as one form of transferring the insights and knowledge of a human expert to a machine learning system (via, e.g., heuristics). Other approaches to doing so include normal per instance annotation or active learning. First ideas have been proposed to combine active learning with weak supervision in a practical way, e.g. by Gonsior et al. (2020), Brust et al. (2020) and Biegel et al. (2021). We are interested to see if and how these different forms of knowledge transfer distinguish themselves from an information-theoretic perspective, i.e., if the information they provide is similar or orthogonal. A per instance annotation might provide high accuracy information but with a small coverage compared to a weak supervision approach that can cover a wide range of documents with less accuracy. These different types of information might be useful in different stages of training a machine learning system. Insights on these questions could then guide ways to combine different annotation approaches in an efficient form.

BIBLIOGRAPHY

[1] Idris Abdulmumin and Bashir Shehu Galadanci. "hauWE: Hausa Words Embedding for Natural Language Processing." In: *2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf)* (2019). DOI: 10.1109/nigeriacomputconf45974.2019.8949674.

[2] Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. "Cross-Lingual Word Embeddings for Low-Resource Language Modeling." In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, 2017. URL: https://www.aclweb.org/anthology/E17-1088.

[3] Heike Adel and Hinrich Schütze. "CIS at TAC Cold Start 2015: Neural Networks and Coreference Resolution for Slot Filling." In: *Proceedings of TAC KBP Workshop*. 2015. URL: https://tac.nist.gov/publications/2015/participant.papers/TAC2015.CIS.proceedings.pdf.

[4] 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 Yoruba." In: *Workshop on Practical Machine Learning for Developing Countries at ICLR'20* (2020). URL: https://arxiv.org/pdf/2003.08370.pdf.

[5] David Ifeoluwa Adelani et al. "MasakhaNER: Named Entity Recognition for African Languages." In: *arXiv preprint arXiv:2103.11811* (2021). URL: https://arxiv.org/pdf/2103.11811.pdf.

[6] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. "Docbert: Bert for document classification." In: *arXiv preprint arXiv:1904.08398* (2019). URL: https://arxiv.org/pdf/1904.08398.pdf.

[7] Charu C Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and S Yu Philip. "Active learning: A survey." In: *Data Classification: Algorithms and Applications*. CRC Press, 2014. URL: http://charuaggarwal.net/active-survey.pdf.

[8] Željko Agić and Ivan Vulić. "JW300: A Wide-Coverage Parallel Corpus for Low-Resource Languages." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1310.

[9] Rakesh Agrawal, T. Imielinksi, and A. Swami. "Mining association rules between sets of items in large databases." In: ACM, 1993.

[10] Roee Aharoni and Yoav Goldberg. "Unsupervised Domain Clusters in Pretrained Language Models." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. URL: https://www.aclweb.org/anthology/2020.acl-main.692.

[11] Alan Akbik, Duncan Blythe, and Roland Vollgraf. "Contextual String Embeddings for Sequence Labeling." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018. URL: https://www.aclweb.org/anthology/C18-1139.

[12] Alan Akbik, Vishwajeet Kumar, and Yunyao Li. "Towards Semi-Automatic Generation of Proposition Banks for Low-Resource Languages." In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Jian Su, Xavier Carreras, and Kevin Duh. The Association for Computational Linguistics, 2016. DOI: 10.18653/v1/d16-1102.

[13] Jesujoba O Alabi, Kwabena Amponsah-Kaakyire, David I Adelani, and Cristina España-Bonet. "Massive vs. Curated Word Embeddings for Low-Resourced Languages. The Case of Yorùbá and Twi." In: *Proceedings of LREC 2020*. 2020.

[14] Jesujoba Alabi, Kwabena Amponsah-Kaakyire, David Adelani, and Cristina Espana-Bonet. "Massive vs. Curated Word Embeddings for Low-Resourced Languages. The Case of Yorùbá and Twi." In: *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association, 2020. URL: https://www.aclweb.org/anthology/2020.lrec-1.335.

[15] Jesujoba Alabi, Kwabena Amponsah-Kaakyire, David Adelani, and Cristina España-Bonet. "Massive vs. Curated Embeddings for Low-Resourced Languages: the Case of Yorùbá and Twi." English. In: *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association, 2020. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.335.

[16] Görkem Algan and Ilkay Ulusoy. "Image classification with deep learning in the presence of noisy labels: A survey." In: *Knowl. Based Syst.* 215 (2021), p. 106771. DOI: 10.1016/j.knosys.2021.106771.

[17] Khalid Alnajjar, Mika Hämäläinen, Jack Rueter, and Niko Partanen. "Ve'rdd. Narrowing the Gap between Paper Dictionaries, Low-Resource NLP and Community Involvement." In: *Pro-*

ceedings of the 28th International Conference on Computational Linguistics: System Demonstrations. International Committee on Computational Linguistics (ICCL), 2020. DOI: 10.18653/v1/2020.coling-demos.1.

[18] 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*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/W19-1909.

[19] Christoph Alt, Marc Hübner, and Leonhard Hennig. "Fine-tuning Pre-Trained Transformer Language Models to Distantly Supervised Relation Extraction." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1134.

[20] Maaz Amjad, Grigori Sidorov, and Alisa Zhila. "Data Augmentation using Machine Translation for Fake News Detection in the Urdu Language." English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, 2020. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.309.

[21] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. "Do Not Have Enough Data? Deep Learning to the Rescue!" In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020. URL: https://aaai.org/ojs/index.php/AAAI/article/view/6233.

[22] Martin Atzmueller. "Subgroup discovery." In: *WIREs Data Mining and Knowledge Discovery* 5.1 (2015).

[23] Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. "Generalisation in named entity recognition: A quantitative analysis." In: *Computer Speech & Language* 44 (2017).

[24] Mahmoud Al-Ayyoub, Aya Nuseir, Kholoud Alsmearat, Yaser Jararweh, and Brij Gupta. "Deep learning for Arabic NLP: A survey." In: *Journal of computational science* 26 (2018). URL: https://www.sciencedirect.com/science/article/pii/S1877750317303757.

[25] Stephen H. Bach et al. "Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale." In: *Proceedings of the 2019 International Conference on Management of Data, SIG-*

MOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019. ACM, 2019. DOI: 10.1145/3299869.3314036.

[26] N. Banik, M. H. Hafizur Rahman, S. Chakraborty, H. Seddiqui, and M. A. Azim. "Survey on Text-Based Sentiment Analysis of Bengali Language." In: *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*. 2019.

[27] 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)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.38.

[28] Nahla Barakat and Joachim Diederich. "Eclectic rule-extraction from support vector machines." In: *International Journal of Computational Intelligence* 2.1 (2005).

[29] Muazzam Bashir, Azilawati Rozaimee, and Wan Malini Wan Isa. "Automatic Hausa Language Text Summarization Based on Feature Extraction using Naive Bayes Model." In: *World Applied Science Journal* 35.9 (2017).

[30] Alan Joseph Bekker and Jacob Goldberger. "Training deep neural-networks based on unreliable labels." In: *Proceedings ICASSP 2016*. 2016.

[31] 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1371.

[32] Emily Bender. "The BenderRule: On Naming the Languages We Study and Why It Matters." In: *The Gradient* (2019). URL: https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/.

[33] Esther Maria van den Berg. "Noisy Label Neural Network Approach to Named Entity Recognition." MA thesis. Rijksuniversiteit Groningen, Universität des Saarlandes, 2016.

[34] 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)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.636.

[35] Samantha Biegel, Rafah El-Khatib, Luiz Otávio Vilas Boas Oliveira, Max Baak, and Nanne Aben. "Active WeaSuL: Improving Weak Supervision with Active Learning." In: *Proceedings of the First Workshop on Weakly Supervised Learning (WeaSuL) at ICLR'21*. 2021. eprint: 2104.14847. URL: https://arxiv.org/abs/2104.14847.

[36] Elan Van Biljon, Arnu Pretorius, and Julia Kreutzer. "On Optimal Transformer Depth for Low-Resource Language Translation." In: *CoRR* abs/2004.04418 (2020). URL: https://arxiv.org/abs/2004.04418.

[37] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching Word Vectors with Subword Information." In: *Transactions of the Association for Computational Linguistics* 5 (2017). DOI: 10.1162/tacl_a_00051.

[38] Ondřej Bojar and Aleš Tamchyna. "Improving Translation Model by Monolingual Data." In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2011. URL: https://www.aclweb.org/anthology/W11-2138.

[39] Clemens-Alexander Brust, Christoph Käding, and Joachim Denzler. "Active and Incremental Learning with Weak Supervision." In: *Künstliche Intell.* 34.2 (2020). URL: https://doi.org/10.1007/s13218-020-00631-4.

[40] Timothy I. Cannings, Yingying Fan, and Richard J. Samworth. "Classification with imperfect training labels." In: *CoRR* abs/1805.11505 (2018).

[41] Steven Cao, Nikita Kitaev, and Dan Klein. "Multilingual Alignment of Contextual Word Representations." In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=r1xCMyBtPS.

[42] Yixin Cao, Zikun Hu, Tat-seng Chua, Zhiyuan Liu, and Heng Ji. "Low-Resource Name Tagging Learned with Weakly Labeled Data." 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1025.

[43] Daoyuan Chen, Yaliang Li, Kai Lei, and Ying Shen. "Relabel the Noise: Joint Extraction of Entities and Relations via Cooperative Multiagents." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. URL: https://www.aclweb.org/anthology/2020.acl-main.527.

[44] Junfan Chen, Richong Zhang, Yongyi Mao, Hongyu Guo, and Jie Xu. "Uncover the Ground-Truth Relations in Distant Supervision: A Neural Expectation-Maximization Framework." In: *Proceedings of EMNLP-IJCNLP 2019*. 2019.

[45] Pengfei Chen, Benben Liao, Guangyong Chen, and Shengyu Zhang. "Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels." In: *Proceedings of ICML 2019*. 2019.

[46] 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)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.413.

[47] Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. "Adversarial Deep Averaging Networks for Cross-Lingual Sentiment Classification." In: *Transactions of the Association for Computational Linguistics* 6 (2018). DOI: 10.1162/tacl_a_00039.

[48] Jiacheng Cheng, Tongliang Liu, Kotagiri Ramamohanarao, and Dacheng Tao. "Learning with Bounded Instance and Label-dependent Label Noise." In: *Proceedings of ICML 2020*. 2020.

[49] Artem Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. "TARGER: Neural Argument Mining at Your Fingertips." In: *Proceedings of the 57th Annual Meeting of the Association of Computational Linguistics (ACL'2019)*. 2019.

[50] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation." In: *Proceedings of EMNLP 2014*. 2014. DOI: 10.3115/v1/D14-1179.

[51] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." 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*. ACL, 2014. DOI: 10.3115/v1/d14-1179.

[52] Christos Christodoulopoulos and Mark Steedman. "A massively parallel corpus: the Bible in 100 languages." In: *Lang. Resour. Evaluation* 49.2 (2015). DOI: 10.1007/s10579-014-9287-y.

[53] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. "Automated Data Slicing for Model Validation: A Big Data - AI Integration Approach." In: *IEEE Trans. Knowl. Data Eng.* 32.12 (2020).

[54] Christopher Cieri, Mike Maxwell, Stephanie Strassel, and Jennifer Tracey. "Selection Criteria for Low Resource Language Programs." In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), 2016. URL: https://www.aclweb.org/anthology/L16-1720.

[55] Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. "Semi-Supervised Sequence Modeling with Cross-View Training." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/D18-1217.

[56] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch." In: *Journal of machine learning research* 12.ARTICLE (2011). URL: https://jmlr.org/papers/volume12/collobert11a/collobert11a.pdf.

[57] 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*. 2020. DOI: 10.18653/v1/2020.acl-main.747.

[58] Ryan Cotterell et al. "The CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection." In: *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/K18-3001.

[59] Jan Christian Blaise Cruz and Charibeth Cheng. "Evaluating language model finetuning techniques for low-resource languages." In: *arXiv preprint arXiv:1907.00409* (2019). URL: https://arxiv.org/abs/1907.00409.

[60] 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*. International Committee on Computational Linguistics, 2020. DOI: 10.18653/v1/2020.coling-main.343.

[61] Bhavana Dalvi, Sumithra Bhakthavatsalam, Chris Clark, Peter Clark, Oren Etzioni, Anthony Fader, and Dirk Groeneveld. "IKE - An Interactive Tool for Knowledge Extraction." In: *Proceedings of AKBC 2016*. 2016. DOI: 10.18653/v1/W16-1303.

[62] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. "A Survey of the State of Explainable AI for Natural Language Processing." In: *AACL/IJCNLP 2020*. Ed. by Kam-Fai Wong, Kevin Knight, and Hua Wu. 2020.

[63] Ali Daud, Wahab Khan, and Dunren Che. "Urdu language processing: a survey." In: *Artificial Intelligence Review* 47.3 (2017). URL: https://link.springer.com/content/pdf/10.1007/s10462-016-9482-x.pdf.

[64] Guy De Pauw, Gilles-Maurice De Schryver, Laurette Pretorius, and Lori Levin. "Introduction to the special issue on African Language Technology." In: *Language Resources and Evaluation* 45.3 (2011).

[65] Mathieu Dehouck and Carlos Gómez-Rodriguez. "Data Augmentation via Subtree Swapping for Dependency Parsing of Low-Resource Languages." In: *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 2020. DOI: 10.18653/v1/2020.coling-main.339.

[66] Julia Dembowski, Michael Wiegand, and Dietrich Klakow. "Language Independent Named Entity Recognition using Distant Supervision." In: *Proceedings of Language and Technology Conference*. 2017.

[67] Xiang Deng and Huan Sun. "Leveraging 2-hop Distant Supervision from Table Entity Pairs for 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1039.

[68] Jacob Devlin. *mBERT README file*. 2019. URL: https://github.com/google-research/bert/blob/cc7051dc592802f501e8a6f71f8fb3cf9de95dc9/multilingual.md (visited on 05/29/2020).

[69] 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, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/N19-1423.

[70] Y. Dgani, H. Greenspan, and J. Goldberger. "Training a neural network based on unreliable human annotation of medical images." In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. 2018.

[71] Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. "DAGA: Data Augmentation with a Generation Approach for Low-resource Tagging Tasks." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.488.

[72] G. Dong and J. Li. "Efficient mining of emerging patterns: Discovering trends and differences." In: ACM New York, NY, USA. 1999.

[73] 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1112.

[74] David M. Eberhard, Gary F. Simons, and Charles D. Fennig (eds.) *Ethnologue: Languages of the World. Twenty-second edition.* SIL International, 2019. URL: http://www.ethnologue.com.

[75] Roald Eiselen. "Government Domain Named Entity Recognition for South African Languages." In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), 2016. URL: https://www.aclweb.org/anthology/L16-1533.

[76] Roald Eiselen and Martin J. Puttkammer. "Developing Text Resources for Ten South African Languages." In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*. European Language Resources Association (ELRA), 2014. URL: http://www.lrec-conf.org/proceedings/lrec2014/summaries/1151.html.

[77] Ramy Eskander, Smaranda Muresan, and Michael Collins. "Unsupervised Cross-Lingual Part-of-Speech Tagging for Truly Low-Resource Scenarios." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.391.

[78] Felix Abidemi Fabuni and Akeem Segun Salawu. "Is Yorùbá an endangered language?" In: *Nordic Journal of African Studies* 14.3 (2005). URL: https://www.njas.fi/njas/article/view/262.

[79] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. "Data Augmentation for Low-Resource Neural Machine Translation." In: *Proceedings of the 55th Annual Meeting of the Association for*

Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, 2017. DOI: 10.18653/v1/P17-2090.

[80] Meng Fang and Trevor Cohn. "Learning when to trust distant supervision: An application to low-resource POS tagging using cross-lingual projection." In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. 2016.

[81] Meng Fang and Trevor Cohn. "Model Transfer for Tagging Low-resource Languages using a Bilingual Dictionary." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/P17-2093.

[82] Hao Fei, Meishan Zhang, and Donghong Ji. "Cross-Lingual Semantic Role Labeling with High-Quality Translated Training Corpus." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. URL: https://www.aclweb.org/anthology/2020.acl-main.627.

[83] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling." In: *Proceedings of ACL 2005*. 2005. URL: http://aclweb.org/anthology/P05-1045.

[84] 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 - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017. URL: http://proceedings.mlr.press/v70/finn17a.

[85] J. Fischer and J. Vreeken. "Sets of Robust Rules, and How to Find Them." In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*. Springer, 2019.

[86] Jonas Fischer, Anna Olah, and Jilles Vreeken. "What's in the Box? Exploring the Inner Life of Neural Networks with Robust Rules." In: *ICML*. Vol. 139. Proceedings of Machine Learning Research. 2021.

[87] Jonas Fischer and Jilles Vreeken. "Discovering Succinct Pattern Sets Expressing Co-Occurrence and Mutual Exclusivity." In: *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 2020.

[88] Ronald A Fisher. "On the interpretation of χ 2 from contingency tables, and the calculation of P." In: *Journal of the Royal Statistical Society* 85.1 (1922).

[89] B. Frenay and M. Verleysen. "Classification in the Presence of Label Noise: A Survey." In: *IEEE Transactions on Neural Networks and Learning Systems* 25.5 (2014).

[90] 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*. Association for Computational Linguistics, 2020. URL: https://www.aclweb.org/anthology/2020.acl-main.116.

[91] Annemarie Friedrich and Damyana Gateva. "Classification of telicity using cross-linguistic annotation projection." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/D17-1271.

[92] Nicholas Frosst and Geoffrey Hinton. "Distilling a neural network into a soft decision tree." In: *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017*. 2017.

[93] A.M. García-Vico, C.J. Carmona, D. Martín, M. García-Borroto, and M.J. del Jesus. "An overview of emerging pattern mining in supervised descriptive rule discovery: taxonomy, empirical study, trends, and prospects." In: *WIREs Data Mining and Knowledge Discovery* 8.1 (2018).

[94] Matt Gardner et al. "Evaluating Models' Local Decision Boundaries via Contrast Sets." In: *EMNLP*. 2020.

[95] Siddhant Garg and Goutham Ramakrishnan. "BAE: BERT-based Adversarial Examples for Text Classification." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.498.

[96] Dan Garrette and Jason Baldridge. "Learning a Part-of-Speech Tagger from Two Hours of Annotation." In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2013. URL: https://www.aclweb.org/anthology/N13-1014.

[97] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep sparse rectifier neural networks." In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011.

[98] Jacob Goldberger and Ehud Ben-Reuven. "Training deep neural-networks using a noise adaptation layer." In: *Proceedings of ICLR 2016*. 2016.

[99] Julius Gonsior, Maik Thiele, and Wolfgang Lehner. "WeakAL: Combining Active Learning and Weak Supervision." In: *International Conference on Discovery Science*. Springer. 2020.

[100] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets." In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[101] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. "Learning Word Vectors for 157 Languages." In: *Proceedings of LREC 2018*. 2018.

[102] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. "Learning Word Vectors for 157 Languages." In: *LREC*. 2018.

[103] Daniel Grießhaber, Ngoc Thang Vu, and Johannes Maucher. "Low-resource text classification using domain-adversarial learning." In: *Computer Speech & Language* 62 (2020). URL: https://www.sciencedirect.com/science/article/pii/S0885230819303006.

[104] Aditi Sharma Grover, Karen Calteaux, Gerhard van Huyssteen, and Marthinus Pretorius. "An overview of HLTs for South African Bantu languages." In: *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*. 2010. URL: https://dl.acm.org/doi/pdf/10.1145/1899503.1899547.

[105] Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. "Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data." In: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/W19-4427.

[106] Peter Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

[107] Imane Guellil, Faiçal Azouaou, and Alessandro Valitutti. "English vs Arabic Sentiment Analysis: A Survey Presenting 100 Work Studies, Resources and Tools." In: *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*. 2019. URL: https://ieeexplore.ieee.org/abstract/document/9035299.

[108] 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*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/D17-1256.

[109] Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. "Colorless Green Recurrent Networks Dream Hierarchically." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/N18-1108.

[110] Sonal Gupta and Christopher Manning. "SPIED: Stanford Pattern based Information Extraction and Diagnostics." In: *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. 2014. DOI: 10.3115/v1/W14-3106.

[111] Suchin Gururangan, Ana Marasović, 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*. Association for Computational Linguistics, 2020. URL: https://www.aclweb.org/anthology/2020.acl-main.740.

[112] DN Hakro, AZ TALIB, and GN Mojai. "Multilingual Text Image Database for OCR." In: *Sindh University Research Journal-SURJ (Science Series)* 47.1 (2016). URL: https://sujo-old.usindh.edu.pk/index.php/SURJ/article/view/2299.

[113] Kishaloy Halder, Alan Akbik, Josip Krapac, and Roland Vollgraf. "Task-Aware Representation of Sentences for Generic Text Classification." In: *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*. Ed. by Donia Scott, Núria Bel, and Chengqing Zong. International Committee on Computational Linguistics, 2020. DOI: 10.18653/v1/2020.coling-main.285.

[114] Wilhelmiina Hämäläinen. "Kingfisher: an efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures." In: 32.2 (2012).

[115] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. "Co-teaching: Robust training of deep neural networks with extremely noisy labels." In: *Proceedings of NeurIPS 2018*. 2018.

[116] BS Harish and R Kasturi Rangan. "A comprehensive survey on Indian regional language processing." In: *SN Applied Sciences* 2.7 (2020). URL: https://link.springer.com/article/10.1007/s42452-020-2983-x.

[117] Michael A. Hedderich, David Adelani, Dawei Zhu, Jesujoba Alabi, Udia Markus, and Dietrich Klakow. "Transfer Learning and Distant Supervision for Multilingual Transformer Models: A Study on African Languages." In: *Proceedings of EMNLP 2020*. 2020.

[118] Michael A. Hedderich, Jonas Fischer, Dietrich Klakow, and Jilles Vreeken. "Label-Descriptive Patterns and Their Application to Characterizing Classification Errors." In: *Proceedings of the 39th International Conference on Machine Learning (ICML)*. 2022. URL: https://proceedings.mlr.press/v162/hedderich22a.html.

[119] Michael A. Hedderich and Dietrich Klakow. "Training a Neural Network in a Low-Resource Setting on Automatically Annotated Noisy Data." In: *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP at ACL 2018*. 2018.

[120] 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: (2021). DOI: 10.18653/v1/2021.naacl-main.201.

[121] Michael A. Hedderich, Lukas Lange, and Dietrich Klakow. "ANEA: Distant Supervision for Low-Resource Named Entity Recognition." In: (2021). URL: https://arxiv.org/abs/2102.13129.

[122] Michael A. Hedderich, Dawei Zhu, and Dietrich Klakow. "Analysing the Noise Model Error for Realistic Noisy Label Data." In: *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*. 2021. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16938.

[123] Chaitra Hegde and Shrikumar Patil. "Unsupervised Paraphrase Generation using Pre-trained Language Models." In: *CoRR* abs/2006.05477 (2020). arXiv: 2006.05477. URL: https://arxiv.org/abs/2006.05477.

[124] 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)*. European Language Resources Association (ELRA), 2018. ISBN: 979-10-95546-00-9. URL: https://www.aclweb.org/anthology/L18-1473.

[125] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. "Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise." In: *Proceedings of NeurIPS 2018*. 2018.

[126] Andreas Henelius, Kai Puolamäki, Henrik Boström, Lars Asker, and Panagiotis Papapetrou. "A Peek into the Black Box: Exploring Classifiers by Randomization." In: *Data Min. Knowl. Discov.* 28.5–6 (2014).

[127] Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. "Iterative Back-Translation for Neural Machine Translation." In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/W18-2703.

[128] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997).

[129] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. *spaCy: Industrial-strength Natural Language Processing in Python*. 2020. DOI: 10.5281/zenodo.1212303.

[130] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. "Meta-learning in neural networks: A survey." In: *arXiv preprint arXiv:2004.05439* (2020). URL: https://arxiv.org/abs/2004.05439.

[131] 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: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1607.

[132] 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: *International Conference on Machine Learning* (2020). URL: http://proceedings.mlr.press/v119/hu20b.html.

[133] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. *XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization*. 2020. arXiv: 2003.11080.

[134] Linmei Hu, Luhao Zhang, Chuan Shi, Liqiang Nie, Weili Guan, and Cheng Yang. "Improving Distantly-Supervised Relation Extraction with Joint Label Embedding." 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). Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1395.

[135] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. "Clinicalbert: Modeling clinical notes and predicting hospital readmission." In: *arXiv preprint arXiv:1904.05342* (2019). URL: https://arxiv.org/abs/1904.05342.

[136] Yi Huang, Junlan Feng, Shuo Ma, Xiaoyu Du, and Xiaoting Wu. "Towards Low-Resource Semi-Supervised Dialogue Generation with Meta-Learning." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.findings-emnlp.368.

[137] 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1037.

[138] Patrick Huber and Giuseppe Carenini. "MEGA RST Discourse Treebanks with Structure and Nuclearity from Scalable Distant Sentiment Supervision." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.603.

[139] Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Søgaard. "DaNE: A Named Entity Resource for Danish." English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, 2020. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.565.

[140] Ayush Jain and Meenachi Ganesamoorty. "NukeBERT: A Pretrained language model for Low Resource Nuclear Domain." In: *arXiv preprint arXiv:2003.13821* (2020). URL: https://arxiv.org/abs/2003.13821.

[141] Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. "ARNOR: Attention Regularization based Noise Reduction for Distant Supervision Relation Classification." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1135.

[142] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. "Beyond synthetic noise: Deep learning on controlled noisy labels." In: *Proceedings of ICML 2020*. 2020.

[143] Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. "Better Modeling of Incomplete Annotations for Named Entity Recognition." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/N19-1079.

[144] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. "Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment." In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020. URL: https://aaai.org/ojs/index.php/AAAI/article/view/6311.

[145] Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. "The State and Fate of Linguistic Diversity and Inclusion in the NLP World." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.560.

[146] Armand Joulin, Piotr Bojanowski, Tomas 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*. 2018. DOI: 10.18653/v1/D18-1330.

[147] Jakob Jungmaier, Nora Kassner, and Benjamin Roth. "Dirichlet-Smoothed Word Embeddings for Low-Resource Settings." English. In: *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association, 2020. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.437.

[148] Katharina Kann, Kyunghyun Cho, and Samuel R. Bowman. "Towards Realistic Practices In Low-Resource Natural Language Processing: The Development Set." 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1329.

[149] Katharina Kann, Ophélie Lacroix, and Anders Søgaard. "Weakly Supervised POS Taggers Perform Poorly on *Truly* Low-Resource Languages." In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI*

Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. AAAI Press, 2020. URL: https://aaai.org/ojs/index.php/AAAI/article/view/6317.

[150] Katharina Kann, Ophélie Lacroix, and Anders Søgaard. *Weakly Supervised POS Taggers Perform Poorly on* Truly *Low-Resource Languages*. 2020. URL: https://aaai.org/ojs/index.php/AAAI/article/view/6317.

[151] Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. "Leveraging Just a Few Keywords for Fine-Grained Aspect Detection Through Weakly Supervised Co-Training." 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1468.

[152] Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. "Cross-Lingual Text Classification with Minimal Resources by Transferring a Sparse Teacher." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.findings-emnlp.323.

[153] Giannis Karamanolakis, Subhabrata (Subho) Mukherjee, Guoqing Zheng, and Ahmed H. Awadallah. "Leaving No Valuable Knowledge Behind: Weak Supervision with Self-training and Domain-specific Rules." In: *NAACL 2021*. NAACL 2021, 2021. URL: https://www.microsoft.com/en-us/research/publication/leaving-no-valuable-knowledge-behind-weak-supervision-with-self-training-and-domain-specific-rules/.

[154] 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 Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1586.

[155] Talaat Khalil, Kornel Kiełczewski, Georgios Christos Chouliaras, Amina Keldibek, and Maarten Versteegh. "Cross-lingual intent classification in a low resource industrial setting." 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1676.

[156] 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. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/D18-1176.

[157] 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*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/D17-1302.

[158] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: (2015). Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6980.

[159] 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)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.624.

[160] Sosuke Kobayashi. "Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/N18-2072.

[161] Petri Kontkanen and Petri Myllymäki. "A linear-time algorithm for computing the multinomial stochastic complexity." In: 103.6 (2007). ISSN: 0020-0190.

[162] Sandra Kübler and Desislava Zhekova. "Multilingual coreference resolution." In: *Language and Linguistics Compass* 10.11 (2016). URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/lnc3.12208.

[163] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. "Data Augmentation using Pre-trained Transformer Models." In: *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*. Association for Computational Linguistics, 2020. URL: https://www.aclweb.org/anthology/2020.lifelongnlp-1.3.

[164] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. "Recurrent convolutional neural networks for text classification." In: *Twenty-ninth AAAI conference on artificial intelligence*. 2015.

[165] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. "Interpretable & explorable approximations of black box models." In: *CoRR* abs/1707.01154 (2017). arXiv: 1707.01154.

[166] 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*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-5705.

[167] Lukas Lange, Heike Adel, and Jannik Strötgen. "NLNDE: The Neither-Language-Nor-Domain-Experts' Way of Spanish Medical Document De-Identification." In: *IberLEF@ SEPLN*. 2019. URL: http://ceur-ws.org/Vol-2421/MEDDOCAN_paper_5.pdf.

[168] 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*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.repl4nlp-1.13.

[169] Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. "Adversarial Learning of Feature-based Meta-Embeddings." In: *arXiv preprint arXiv:2010.12305* (2020). URL: https://arxiv.org/pdf/2010.12305.pdf.

[170] Lukas Lange, Michael A. Hedderich, and Dietrich Klakow. "Feature-Dependent Confusion Matrices for Low-Resource NER Labeling with Noisy Labels." In: *Proceedings of EMNLP-IJCNLP 2019*. 2019.

[171] 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*. Association for Computational Linguistics, 2020. URL: https://www.aclweb.org/anthology/2020.repl4nlp-1.14.

[172] Jan Larsen, L Nonboe, Mads Hintz-Madsen, and Lars Kai Hansen. "Design of robust neural network classifiers." In: *Proceedings of ICASSP 1998*. 1998.

[173] Sven Laur, Siim Orasmaa, Dage Särg, and Paul Tammo. "EstNLTK 1.6: Remastered Estonian NLP Pipeline." In: *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association, 2020. URL: https://www.aclweb.org/anthology/2020.lrec-1.884.

[174] Swen Laur. *Nimeüksuste korpus*. Center of Estonian Language Resources. 2013. DOI: 10.15155/1-00-0000-0000-0000-00073L.

[175] 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 EMNLP 2020*. 2020.

[176] Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. "From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers." In: (2020). DOI: 10.18653/v1/2020.emnlp-main.363.

[177] Phong Le and Ivan Titov. "Distant Learning for Entity Linking with Automatic Noise Detection." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1400.

[178] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998).

[179] Gyeongbok Lee, Sungdong Kim, and Seung-won Hwang. "QADiver: Interactive Framework for Diagnosing QA Models." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019.

[180] Ji-Ung Lee, Jan-Christoph Klie, and Iryna Gurevych. "Annotation Curricula to Implicitly Train Non-Expert Annotators." In: *arXiv preprint arXiv:2106.02382* (2021). URL: https://arxiv.org/abs/2106.02382.

[181] Jieh-Sheng Lee and Jieh Hsiang. "Patent classification by fine-tuning BERT language model." In: *World Patent Information* 61 (2020). URL: https://www.sciencedirect.com/science/article/pii/S0172219019300742.

[182] 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." In: *Bioinformatics (Oxford, England)* 36.4 (2020). ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz682.

[183] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. "CleanNet: Transfer Learning for Scalable Image Classifier Training With Label Noise." In: *Proceedings of CVPR 2018*. 2018.

[184] Matthijs van Leeuwen and Arno J. Knobbe. "Diverse subgroup set discovery." In: *Data Mining and Knowledge Discovery* 25.2 (2012).

[185] Florian Lemmerich and Martin Becker. "pysubgroup: Easy-to-use subgroup discovery in python." In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*. 2018.

[186] Junnan Li, Richard Socher, and Steven C. H. Hoi. "DivideMix: Learning with Noisy Labels as Semi-supervised Learning." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=HJgExaVtwr.

[187] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer, 1993.

[188] Shen Li, João Graça, and Ben Taskar. "Wiki-ly Supervised Part-of-Speech Tagging." In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012. URL: https://www.aclweb.org/anthology/D12-1127.

[189] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. "WebVision Database: Visual Learning and Understanding from Web Data." In: *CoRR* abs/1708.02862 (2017). arXiv: 1708.02862.

[190] Yunyao Li, Elmer Kim, Marc A. Touchette, Ramiya Venkatachalam, and Hao Wang. "VINERy: A Visual IDE for Information Extraction." In: *Proc. of the VLDB Endowment* 8.12 (2015). DOI: 10.14778/2824032.2824108.

[191] Yu-Hsiang Lin et al. "Choosing Transfer Languages for Cross-Lingual Learning." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1301.

[192] Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. "Named Entity Recognition without Labelled Data: A Weak Supervision Approach." In: *Proceedings of ACL 2020*. 2020.

[193] D. Liu, N. Ma, F. Yang, and X. Yang. "A Survey of Low Resource Neural Machine Translation." In: *2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*. 2019. URL: https://ieeexplore.ieee.org/abstract/document/8969405.

[194] 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*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/P17-1001.

[195] Qianchu Liu, Diana McCarthy, Ivan Vulić, 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. DOI: 10.18653/v1/K19-1004.

[196] Tongliang Liu and Dacheng Tao. "Classification with Noisy Labels by Importance Reweighting." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 38.3 (2016).

[197] 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." In: *arXiv preprint arXiv:1907.11692* (2019). URL: https://arxiv.org/abs/1907.11692.

[198] Felipe Llinares-López, Mahito Sugiyama, Laetitia Papaxanthos, and Karsten Borgwardt. "Fast and Memory-Efficient Significant Pattern Mining via Permutation Testing." In: *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 2015.

[199] 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*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.findings-emnlp.394.

[200] Melinda Loubser and Martin J. Puttkammer. "Viability of Neural Networks for Core Technologies for Resource-Scarce Languages." In: *Information* 11 (2020).

[201] Jose Lozano, Waldir Farfan, and Juan Cruz. "Syntactic Analyzer for Quechua Language." In: (2013). URL: https://www.researchgate.net/profile/Jose-Lozano-31/publication/259265383_Syntactic_Analyzer_for_Quechua_Language/links/0deec52a9eb8d61f62000000/Syntactic-Analyzer-for-Quechua-Language.pdf.

[202] Bingfeng Luo, Yansong Feng, Zheng Wang, Zhanxing Zhu, Songfang Huang, Rui Yan, and Dongyan Zhao. "Learning with Noise: Enhance Distantly Supervised Relation Extraction with Dynamic Transition Matrix." In: *Proceedings of ACL 2017*. 2017.

[203] Xianbin Lv, Dongxian Wu, and Shu-Tao Xia. "Matrix Smoothing: A Regularization For Dnn With Transition Matrix Under Noisy Labels." In: (2020). DOI: 10.1109/ICME46284.2020.9102853.

[204] Shuming Ma, Pengcheng Yang, Tianyu Liu, Peng Li, Jie Zhou, and Xu Sun. "Key Fact as Pivot: A Two-Stage Model for Low Resource Table-to-Text Generation." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1197.

[205] Xuezhe Ma and Eduard 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 (Volume 1: Long Papers)*. Association for Computational Linguistics, 2016. DOI: 10.18653/v1/P16-1101.

[206] Xuezhe Ma and Eduard Hovy. "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF." In: *Proceedings of ACL 2016*. 2016.

[207] Manuel Mager, Ximena Gutierrez-Vasques, Gerardo Sierra, and Ivan Meza-Ruiz. "Challenges of language technologies for the indigenous languages of the Americas." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018. URL: https://www.aclweb.org/anthology/C18-1006.

[208] Alexandre Magueresse, Vincent Carles, and Evan Heetderks. "Low-resource Languages: A Review of Past Work and Future Challenges." In: *arXiv preprint arXiv:2006.07264* (2020). URL: https://arxiv.org/abs/2006.07264.

[209] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. "Exploring the Limits of Weakly Supervised Pretraining." In: *Computer Vision – ECCV 2018*. Springer International Publishing, 2018.

[210] Ana Marasović, Mengfei Zhou, Alexis Palmer, and Anette Frank. "Modal Sense Classification At Large: Paraphrase-Driven Sense Projection, Semantically Enriched Classification Models and Cross-Genre Evaluations." In: *Linguistic Issues in Language Technology, Volume 14, 2016 - Modality: Logic, Semantics, Annotation, and Machine Learning*. CSLI Publications, 2016. URL: https://www.aclweb.org/anthology/2016.lilt-14.3.

[211] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. "Building a Large Annotated Corpus of English: The Penn Treebank." In: *Comput. Linguist.* 19.2 (1993).

[212] Carmen Martinez-Gil, Alejandro Zempoalteca-Pérez, Venustiano Soancatl-Aguilar, Maria de Jesús Estudillo-Ayala, José Edgar Lara-Ramirez, and Sayde Alcántara-Santiago. "Computer Systems for Analysis of Nahuatl." In: *Res. Comput. Sci.* 47 (2012). URL: https://rcs.cic.ipn.mx/2012_47/Computer%5C%20Systems%5C%20for%5C%20Analysis%5C%20of%5C%20Nahuatl.pdf.

[213] Thomas Mayer and Michael Cysouw. "Creating a massively parallel Bible corpus." In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), 2014. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/220_Paper.pdf.

[214] Stephen Mayhew, Snigdha Chaturvedi, Chen-Tse Tsai, and Dan Roth. "Named Entity Recognition with Partially Annotated Training Data." In: *Proceedings of CoNLL 2019*. 2019.

[215] Stephen Mayhew and Dan Roth. "TALEN: Tool for Annotation of Low-resource ENtities." In: *Proceedings of ACL 2018, System Demonstrations*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/P18-4014.

[216] Stephen Mayhew, Chen-Tse Tsai, and Dan Roth. "Cheap Translation for Cross-Lingual Named Entity Recognition." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/D17-1269.

[217] Dheeraj Mekala, Xinyang Zhang, and Jingbo Shang. "META: Metadata-Empowered Weak Supervision for Text Classification." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.670.

[218] Oren Melamud, Mihaela 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1401.

[219] Aditya Krishna Menon, Brendan van Rooyen, and Nagarajan Natarajan. "Learning from binary labels with instance-dependent noise." In: *Machine Learning* 107.8-10 (2018).

[220] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. "Exploiting Similarities among Languages for Machine Translation." In: *arXiv preprint arXiv:1309.4168* (2013). URL: http://arxiv.org/abs/1309.4168.

[221] George A. Miller. "WordNet: A Lexical Database for English." In: *Commun. ACM* 38.11 (1995).

[222] Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. "Syntactic Data Augmentation Increases Robustness to Inference Heuristics." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.212.

[223] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. "Distant supervision for relation extraction without labeled data." In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics, 2009. URL: https://www.aclweb.org/anthology/P09-1113.

[224] Volodymyr Mnih and Geoffrey Hinton. "Learning to Label Aerial Images from Noisy Data." In: *Proceedings of ICML 2012*. 2012.

[225] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. "TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-demos.16.

[226] Benjamin Müller, Benoit Sagot, and Djamé Seddah. "Can Multilingual Language Models Transfer to an Unseen Dialect? A Case Study on North African Arabizi." In: *CoRR* abs/2005.00318 (2020). arXiv: 2005.00318. URL: https://arxiv.org/abs/2005.00318.

[227] Kaili Müürisep and Pilleriin Mutso. "ESTSUM-Estonian newspaper texts summarizer." In: *Proceedings of The Second Baltic Conference on Human Language Technologies*. 2005.

[228] Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. "Learning with Noisy Labels." In: *Proceedings of NeurIPS 2013*. 2013.

[229] Wilhelmina Nekoto et al. "Participatory Research for Low-resourced Machine Translation: A Case Study in African Languages." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.findings-emnlp.195.

[230] Kamal Nigam, Andrew McCallum, and Tom Mitchell. "Semi-supervised text classification using EM." In: *Semi-Supervised Learning* (2006).

[231] 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*. 2020. URL: https://aclanthology.org/2020.lrec-1.497/.

[232] Farhad Nooralahzadeh, Jan Tore Lønning, and Lilja Øvrelid. "Reinforcement-based denoising of distantly supervised NER with partial annotation." In: *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-6125.

[233] Christopher Norman, Mariska Leeflang, René Spijker, Evangelos Kanoulas, and Aurélie Névéol. "A distantly supervised dataset for automated data extraction from diagnostic studies." In: *Proceedings of the 18th BioNLP Workshop and Shared*

Task. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/W19-5012.

[234] Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. *Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks*. 2021. arXiv: 2103.14749.

[235] Petra Kralj Novak, Nada Lavrac, and Geoffrey I. Webb. "Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining." In: *Journal of Machine Learning Research* 10 (2009).

[236] Hidekazu Oiwa, Yoshihiko Suhara, Jiyu Komiya, and Andrei Lopatenko. "A Lightweight Front-end Tool for Interactive Entity Population." In: *CoRR* abs/1708.00481 (2017). arXiv: 1708.00481. URL: http://arxiv.org/abs/1708.00481.

[237] Fredrik Olsson. "A literature survey of active machine learning in the context of natural language processing." In: (2009). URL: http://eprints.sics.se/3600/.

[238] Yasumasa Onoe and Greg Durrett. "Learning to Denoise Distantly-Labeled Data for Entity Typing." 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/n19-1250.

[239] Juri Opitz. "Argumentative Relation Classification as Plausibility Ranking." In: *Proceedings of the 15th Conference on Natural Language Processing, KONVENS 2019, Erlangen, Germany, October 9-11, 2019*. 2019. URL: https://corpora.linguistik.uni-erlangen.de/data/konvens/proceedings/papers/KONVENS2019%5C_paper%5C_51.pdf.

[240] Diego Ortego, Eric Arazo, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. "Towards Robust Learning with Different Label Noise Distributions." In: (2020). DOI: 10.1109/ICPR48806.2021.9412747.

[241] Hille Pajupuu, Rene Altrov, and Jaan Pajupuu. "Identifying polarity in different text types." In: *Folklore: Electronic Journal of Folklore* 64 (2016).

[242] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning." In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009). URL: https://ieeexplore.ieee.org/abstract/document/5288526/.

[243] Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. "Cross-lingual Name Tagging and Linking for 282 Languages." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/P17-1178.

[244] Laetitia Papaxanthos, Felipe Llinares-López, Dean A. Bodenham, and Karsten M. Borgwardt. "Finding significant combinations of features in the presence of categorical covariates." In: *Advances in Neural Information Processing Systems*. 2016. URL: https://proceedings.neurips.cc/paper/2016/file/0a0a0c8aaa00ade50f74a3f0ca981ed7-Paper.pdf.

[245] Shantipriya Parida and Petr Motlicek. "Abstract Text Summarization: A Low Resource Challenge." 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1616.

[246] Giorgio Patrini, Frank Nielsen, Richard Nock, and Marcello Carioni. "Loss factorization, weakly supervised learning and label noise robustness." In: *Proceedings of ICML 2016*. 2016.

[247] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. "Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach." In: *Proceedings of CVPR 2017*. 2017.

[248] Debjit Paul, Mittul Singh, Michael A. Hedderich, and Dietrich Klakow. "Handling Noisy Labels for Robustly Learning from Self-Training Data for Low-Resource Sequence Labeling." In: *Proceedings of NAACL 2019: Student Research Workshop*. 2019.

[249] Leonardo Pellegrina, Matteo Riondato, and Fabio Vandin. "SPuManTE: Significant Pattern Mining with Unconditional Testing." In: *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 2019.

[250] Leonardo Pellegrina and Fabio Vandin. "Efficient Mining of the Most Significant Patterns with Permutation Testing." In: *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 2018.

[251] Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. "Distantly Supervised Named Entity Recognition using Positive-Unlabeled Learning." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1231.

[252] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation." In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014.

[253] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. "MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.617.

[254] Telmo Pires, Eva Schlinger, and Dan Garrette. "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1493.

[255] Barbara Plank and Željko Agić. "Distant Supervision from Disparate Sources for Low-Resource Part-of-Speech Tagging." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/D18-1061.

[256] 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*. The Association for Computer Linguistics, 2016. DOI: 10.18653/v1/p16-2067.

[257] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. "Towards Robust Linguistic Analysis using OntoNotes." In: *CoNLL*. 2013.

[258] Hugo M. Proença and Matthijs van Leeuwen. "Interpretable multiclass classification by MDL-based rule lists." In: *Information Sciences* 512 (2020).

[259] Kun Qian, Poornima Chozhiyath Raman, Yunyao Li, and Lucian Popa. "Learning Structured Representations of Entity Names using ActiveLearning and Weak Supervision." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.emnlp-main.517.

[260] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. "Pre-trained models for natural language processing: A survey." In: *Science China Technological Sciences* (2020). URL: https://link.springer.com/article/10.1007/s11431-020-1647-3.

[261] 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." In: *Journal of Machine Learning Research* 21 (2020). URL: https://www.jmlr.org/papers/volume21/20-074/20-074.pdf.

[262] Afshin Rahimi, Yuan Li, and Trevor Cohn. "Massively Multilingual Transfer for NER." In: *Proceedings of ACL 2019*. 2019.

[263] Jonathan Raiman and John Miller. "Globally Normalized Reader." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017. DOI: 10.18653/v1/D17-1111.

[264] Alan Ramponi and Barbara Plank. "Neural Unsupervised Domain Adaptation in NLP—A Survey." In: *Proceedings of the 28th International Conference on Computational Linguistics* (2020). DOI: 10.18653/v1/2020.coling-main.603.

[265] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. "Data Programming: Creating Large Training Sets, Quickly." In: *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016.

[266] Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason A. Fries, Sen Wu, and Christopher Ré. "Snorkel: rapid training data creation with weak supervision." In: *VLDB J.* 29.2-3 (2020).

[267] Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. "Training Deep Neural Networks on Noisy Labels with Bootstrapping." In: *ICLR 2015, Workshop Track Proceedings*. 2015.

[268] Ines Rehbein and Josef Ruppenhofer. "Detecting annotation noise in automatically labelled data." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017.

[269] Wendi Ren, Yinghao Li, Hanting Su, David Kartchner, Cassie Mitchell, and Chao Zhang. "Denoising Multi-Source Weak Supervision for Neural Text Classification." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.findings-emnlp.334.

[270] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier." In: *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 2016.

[271] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "Anchors: High-Precision Model-Agnostic Explanations." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2018.

[272] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. "Beyond Accuracy: Behavioral Testing of NLP Models with CheckList." In: *ACL*. 2020.

[273] Sebastian Riedel, Limin Yao, and Andrew McCallum. "Modeling Relations and Their Mentions without Labeled Text." In: *Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-15939-8. URL: https://link.springer.com/chapter/10.1007/978-3-642-15939-8_10.

[274] Jorma Rissanen. "Modeling by shortest data description." In: 14.1 (1978).

[275] Jorma Rissanen. "A Universal Prior for Integers and Estimation by Minimum Description Length." In: 11.2 (1983).

[276] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. "A primer in bertology: What we know about how bert works." In: *Transactions of the Association for Computational Linguistics* 8 (2021). URL: https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00349/96482/A-Primer-in-BERTology-What-We-Know-About-How-BERT.

[277] Marc-Antoine Rondeau and T. J. Hazen. "Systematic Error Analysis of the Stanford Question Answering Dataset." In: *Proceedings of the Workshop on Machine Reading for Question Answering*. 2018.

[278] Brendan van Rooyen and Robert C. Williamson. "A Theory of Learning with Corrupted Labels." In: *J. Mach. Learn. Res.* 18 (2017).

[279] Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. "A survey of noise reduction methods for distant supervision." In: *Proceedings of the 2013 workshop on Automated knowledge base construction*. 2013. URL: https://dl.acm.org/doi/pdf/10.1145/2509558.2509571.

[280] Sebastian Ruder. "The 4 Biggest Open Problems in NLP." In: (2019). URL: https://ruder.io/4-biggest-open-problems-in-nlp.

[281] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. "A survey of cross-lingual word embedding models." In: *Journal of Artificial Intelligence Research* 65 (2019). URL: https://www.jair.org/index.php/jair/article/view/11640.

[282] Gözde Gül Şahin and Mark Steedman. "Data Augmentation via Dependency Tree Morphing for Low-Resource Languages." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/D18-1545.

[283] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2019. arXiv: 1910.01108.

[284] 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*. International Committee on Computational Linguistics, 2020. DOI: 10.18653/v1/2020.coling-main.488.

[285] Timo Schick and Hinrich Schütze. "Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference." In: (2021). URL: https://aclanthology.org/2021.eacl-main.20/.

[286] Timo Schick and Hinrich Schütze. "It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners." In: (2021). DOI: 10.18653/v1/2021.naacl-main.185.

[287] 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, Volume 1 (Long and Short Papers)*. 2019. DOI: 10.18653/v1/N19-1162.

[288] Clayton Scott, Gilles Blanchard, and Gregory Handy. "Classification with Asymmetric Label Noise: Consistency and Maximal Denoising." In: *Proceedings of COLT 2013*. 2013.

[289] Burr Settles. *Active learning literature survey*. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences, 2009. URL: https://minds.wisconsin.edu/handle/1793/60660.

[290] Yong Shi, Yang Xiao, and Lingfeng Niu. "A Brief Survey of Relation Extraction Based on Distant Supervision." In: *International Conference on Computational Science*. Springer. 2019. URL: https://link.springer.com/chapter/10.1007/978-3-030-22744-9_23.

[291] Edwin Simpson, Jonas Pfeiffer, and Iryna Gurevych. "Low Resource Sequence Tagging with Weak Labels." In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2020. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6415.

[292] Alisa Smirnova and Philippe Cudré-Mauroux. "Relation extraction using distant supervision: A survey." In: *ACM Computing Surveys (CSUR)* 51.5 (2018).

[293] Snorkel AI. *Case Studies - Technology Proven in Production at Some of the World's Leading Organizations*. 2022. URL: https://snorkel.ai/case-studies/.

[294] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." In: *Journal of Machine Learning Research* 15 (2014). URL: http://jmlr.org/papers/v15/srivastava14a.html.

[295] Ralf Steinberger. "A survey of methods to ease the development of highly multilingual text mining applications." In: *Language resources and evaluation* 46.2 (2012). DOI: https://doi.org/10.1007/s10579-011-9165-9.

[296] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. "brat: a Web-based Tool for NLP-Assisted Text Annotation." In: *Proceedings of the Demonstrations at EACL 2012*. 2012. URL: http://aclweb.org/anthology/E12-2021.

[297] Stephanie Strassel and Jennifer Tracey. "LORELEI Language Packs: Data, Tools, and Resources for Technology Development in Low Resource Languages." In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), 2016. URL: https://www.aclweb.org/anthology/L16-1521.

[298] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. "Training Convolutional Networks with Noisy Labels." In: *Workshop Track of the International Conference on Learning Representations (ICLR)*. 2015.

[299] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. "Multi-instance Multi-label Learning for Relation Extraction." In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012. URL: https://www.aclweb.org/anthology/D12-1042.

[300] Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. "Token and Type Constraints for Cross-Lingual Part-of-Speech Tagging." In: *Transactions of the Association for Computational Linguistics* 1 (2013). DOI: 10.1162/tacl_a_00205.

[301] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. "A survey on deep transfer learning." In: *International conference on artificial neural networks*. Springer. 2018. URL: https://link.springer.com/chapter/10.1007/978-3-030-01424-7_27.

[302] Hao Tan and Mohit Bansal. "LXMERT: Learning Cross-Modality Encoder Representations from Transformers." In: *EMNLP*. 2019.

[303] Jörg Tiedemann. "Parallel Data, Tools and Interfaces in OPUS." In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), 2012. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

[304] Jörg Tiedemann. "Parallel Data, Tools and Interfaces in OPUS." In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), 2012. ISBN: 978-2-9517408-7-7.

[305] Erik F. Tjong Kim Sang and Fien De Meulder. "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition." In: *Proceedings of HLT-NAACL 2003*. 2003.

[306] Alexander Tkachenko, Timo Petmanson, and Sven Laur. "Named Entity Recognition in Estonian." In: *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing, BSNLP@ACL 2013, Sofia, Bulgaria, August 8-9, 2013*. 2013.

[307] Antonio Torralba, Robert Fergus, and William T. Freeman. "80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 30.11 (2008).

[308] Jennifer Tracey and Stephanie Strassel. "Basic Language Resources for 31 Languages (Plus English): The LORELEI Representative and Incident Language Packs." English. In: *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*. European Language Resources association, 2020. ISBN: 979-10-95546-35-1. URL: https://www.aclweb.org/anthology/2020.sltu-1.39.

[309] Jennifer Tracey et al. "Corpus Building for Low Resource Languages in the DARPA LORELEI Program." In: *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages*. European Association for Machine Translation, 2019. URL: https://www.aclweb.org/anthology/W19-6808.

[310] Tatiana Tsygankova, Francesca Marini, Stephen Mayhew, and Dan Roth. "Building Low-Resource NER Models Using Non-Speaker Annotations." In: (2021). DOI: 10.18653/v1/2021.dash-1.11.

[311] Aminu Tukur, Kabir Umar, and Anas Saidu Muhammad. "Tagging Part of Speech in Hausa Sentences." In: *2019 15th International Conference on Electronics, Computer and Computation*

(ICECCO). IEEE. 2019. URL: https://ieeexplore.ieee.org/abstract/document/9043198/.

[312] Clara Vania, Yova Kementchedjhieva, Anders Søgaard, and Adam Lopez. "A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages." 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1102.

[313] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need." In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[314] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J. Belongie. "Learning From Noisy Large-Scale Datasets With Minimal Supervision." In: *Proceedings of CVPR 2017*. 2017.

[315] R. Vimieiro. "Mining disjunctive patterns in biomedical data sets." PhD thesis. The University of Newcastle, Australia, 2012.

[316] Hao Wang, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li. "Learning with Noisy Labels for Sentence-level Sentiment Classification." In: *Proceedings of EMNLP-IJCNLP 2019*. 2019.

[317] Geoffrey I. Webb. "Discovering Significant Patterns." In: 68.1 (2007).

[318] Jason 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1670.

[319] Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. "OntoNotes Release 4.0." In: *LDC2011T03* (2011).

[320] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning." In: *Journal of Big data* 3.1 (2016). URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0043-6.

[321] Garrett Wilson and Diane J Cook. "A survey of unsupervised deep domain adaptation." In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 11.5 (2020). URL: https://dl.acm.org/doi/abs/10.1145/3400066.

[322] Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. "Cross-Lingual Part-of-Speech Tagging through Ambiguous Learning." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014. DOI: 10.3115/v1/D14-1187.

[323] Thomas Wolf et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2019. arXiv: 1910.03771.

[324] Stefan Wrobel. "An algorithm for multi-relational discovery of subgroups." In: *Principles of Data Mining and Knowledge Discovery*. Springer, 1997.

[325] 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1077.

[326] Shijie Wu and Mark Dredze. "Are All Languages Created Equal in Multilingual BERT?" In: *Proceedings of the 5th Workshop on Representation Learning for NLP*. Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.repl4nlp-1.16.

[327] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. "Errudite: Scalable, Reproducible, and Testable Error Analysis." In: *ACL*. 2019.

[328] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. "Part-dependent Label Noise: Towards Instance-dependent Label Noise." In: *Proceedings of NeurIPS 2020*. 2020.

[329] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. "Are Anchor Points Really Indispensable in Label-Noise Learning?" In: *Proceedings of NeurIPS 2019*. 2019.

[330] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. "Learning from massive noisy labeled data for image classification." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[331] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. "Unsupervised Data Augmentation for Consistency Training." In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/44feb0096faa8326192570788b38c1d1-Abstract.html.

[332] Hu Xu, Bing Liu, Lei Shu, and Philip Yu. "DomBERT: Domain-oriented Language Model for Aspect-based Sentiment Analysis." In: *Findings of the Association for Computational Linguistics: EMNLP 2020* (2020). DOI: 10.18653/v1/2020.findings-emnlp.156.

[333] Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. "Distantly Supervised NER with Partial Annotation Learning and Reinforcement Learning." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018. URL: https://www.aclweb.org/anthology/C18-1183.

[334] Ze Yang, Wei Wu, Jian Yang, Can Xu, and Zhoujun Li. "Low-Resource Response Generation with Template Prior." 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1197.

[335] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. "Dual T: Reducing Estimation Error for Transition Matrix in Label-noise Learning." In: *Proceedings of NeurIPS 2020*. 2020.

[336] David Yarowsky, Grace Ngai, and Richard Wicentowski. "Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora." In: *Proceedings of the First International Conference on Human Language Technology Research*. 2001. URL: https://www.aclweb.org/anthology/H01-1035.

[337] Michihiro Yasunaga, Jungo Kasai, and Dragomir 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, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/N18-1089.

[338] Qinyuan Ye, Liyuan Liu, Maosen Zhang, and Xiang Ren. "Looking Beyond Label Noise: Shifted Label Distribution Matters in 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). Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1397.

[339] Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. "Automatic Annotation Suggestions and Custom Annotation Layers in WebAnno." In: *Proceedings of ACL 2014: System Demonstrations*. 2014. DOI: 10.3115/v1/P14-5016.

[340] Jihene Younes, Emna Souissi, Hadhemi Achour, and Ahmed Ferchichi. "Language resources for Maghrebi Arabic dialects' NLP: a survey." In: *LANGUAGE RESOURCES AND EVALUATION* (2020). URL: https://link.springer.com/article/10.1007%5C%2Fs10579-020-09490-9.

[341] 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, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018. DOI: 10.18653/v1/N18-1109.

[342] BI Yude. "A Brief Survey of Korean Natural Language Processing Research." In: *Journal of Chinese Information Processing* 6 (2011). URL: http://en.cnki.com.cn/Article_en/CJFDTotal-MESS201106022.htm.

[343] Boliang Zhang, Ying Lin, Xiaoman Pan, Di Lu, Jonathan May, Kevin Knight, and Heng Ji. "ELISA-EDL: A Cross-lingual Entity Extraction, Linking and Localization System." In: *Proceedings of NAACL-HLT 2018: Demonstrations*. 2018. DOI: 10.18653/v1/N18-5009.

[344] Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. "WRENCH: A Comprehensive Benchmark for Weak Supervision." In: *arXiv preprint arXiv:2109.11377* (2021). URL: https://arxiv.org/abs/2109.11377.

[345] 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/D19-1092.

[346] Rui Zhang, Caitlin Westerfield, Sungrok Shim, Garrett Bingham, Alexander Fabbri, William Hu, Neha Verma, and Dragomir Radev. "Improving Low-Resource Cross-lingual Document Retrieval by Reranking with Deep Bilingual Representations." In: *Proceedings of the 57th Annual Meeting of the Association for*

Computational Linguistics. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1306.

[347] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level Convolutional Networks for Text Classification." In: *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015. URL: http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification.pdf.

[348] Shun Zheng, Xu Han, Yankai Lin, Peilin Yu, Lu Chen, Ling Huang, Zhiyuan Liu, and Wei Xu. "DIAG-NRE: A Neural Pattern Diagnosis Framework for Distantly Supervised Neural Relation Extraction." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1137.

[349] 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 Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/P19-1336.

[350] Mengfei Zhou, Anette Frank, Annemarie Friedrich, and Alexis Palmer. "Semantically Enriched Models for Modal Sense Classification." In: *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*. Association for Computational Linguistics, 2015. DOI: 10.18653/v1/W15-2705.

[351] Yi Zhu, Benjamin Heinzerling, Ivan Vulić, 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)*. Association for Computational Linguistics, 2019. DOI: 10.18653/v1/K19-1021.

[352] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. "Visual7W: Grounded Question Answering in Images." In: *IEEE CVPR*. 2016.