Linguistically Aware and Augmentation-Driven Methods for Enhancing Natural Language Understanding

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
Maren Runa Judith Pielka
aus
Köln

Bonn, 31.03.2025

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachter/Betreuer:	Prof. Dr. Rafet Sifa
Gutachterin:	Prof. Dr. Lucie Flek
Tag der Promotion:	02.07.2025
Erscheinungsjahr:	2025

Abstract

Current large language models excel in solving numerous complicated tasks, but are primarily optimized for the English language and popular application domains, delivering sub-optimal outputs for low-resource languages and specialized industry use cases. Also, they rely heavily on large amounts of training data and computing resources. In this work, we aim to tackle those issues by implementing smaller, less resource-intensive models which are trained in an informed way, leveraging linguistic knowledge about semantic and syntactic features.

To this end, we investigate methods for linguistically informed pre-training, incorporating the model with additional semantic and syntactic knowledge prior to fine-tuning on the downstream task. We specifically consider token-level prediction tasks with high semantic and syntactic relevance, such as Part-of-Speech-Tagging and Synset Prediction based on semantic webs. Our experimental results show that smaller models perform on par with larger ones when being pre-trained on those tasks, suggesting that this method can contribute to making language modeling more efficient.

Another direction of research is the creation of prototypical training corpora, exploiting both linguistic knowledge and the generative power of large pre-trained language models. We hypothesize that using those prototypical data sets in language model training will help reduce the total amount of data needed, while keeping a similar performance on the downstream task. This conjecture is being confirmed by experimental results, underlining the goal of this thesis.

To further test our hypothesis, we evaluate the informed pre-training and data generation approaches in low-resource scenarios, namely on the tasks of Natural Language Inference and Contradiction Detection in German and Arabic. We find that language model performance in those domains can be significantly improved using the aforementioned methods. Machine Translation is also being introduced as an effective method to obtain training corpora in under-researched languages.

Finally, we evaluate the efficiency of those approaches on the basis of three real-world use cases from the financial domain. We specifically look at Causality Detection, Critical Error Detection, as well as Contradiction Detection in financial reports. In all three cases, our methods provide a significant performance boost, combined with insights into the nature of language for this specific domain.

Overall, this thesis significantly contributes to the language modeling research field, exploring options to improve current paradigms for specialized scenarios and with a resource-aware objective.

Acknowledgements

First and foremost, I would like to thank my supervisors, Prof. Dr. Rafet Sifa and Prof. Dr. Lucie Flek, for guiding me through my PhD journey and providing helpful feedback in all stages of this work. I also want to express my gratitude for Prof. Dr. Maren Bennewitz and Prof. Dr. Claudia Wich-Reif, who completed my dissertation comittee.

Another big thank you goes out to all (current and former) colleagues from Uni Bonn and Fraunhofer IAIS, who have supported me during my PhD years by collaborating on projects and papers, providing invaluable feedback and having my back when things got rough. I especially would like to name here: Lars Hillebrand, Tobias Deußer, Armin Berger, Lorenz Sparrenberg, Priya Tomar, Daniel Uedelhoven, Thiago Bell, Benjamin Wulff, Maurice Günder, Thore Gerlach, Max Lübbering, David Biesner, Rajkumar Ramamurthy, Kostadin Cvejoski, Anna Ladi, Laura von Rüden, and Robin Stenzel.

I also would like to mention some exceptionally gifted Bachelor's and Master's students I had the honor to supervise, and who contributed significantly to this thesis: Svetlana Schmidt, Lisa Pucknat, Marie-Christin Freischlad, Bouthaina Abdou, Majd Saad al Deen, and Maria Chiara Talarico.

Three other people who have shaped my PhD journey, and who deserve a mention here, are: Sven Giesselbach for our very open and honest conversations, as well as Luise Schneider and Alex Zoll for helping me build professional self-confidence and figure out my work-life goals with their invaluable coaching skills.

Finally, I want to express my gratefulness for my friends and family, who gave me much needed mental and emotional support during my PhD years. I would like to specifically thank my mom for proof-reading this thesis and supporting me on my PhD journey, both by providing feedback from a linguistic and philological perspective and by having an open ear at all times.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Research in the domain of language modeling and natural language understanding (NLU) has been progressing rapidly over the course of the last years. While statistical methods for word representations, such as Tf-idf [1] and GloVe [2], combined with classical machine learning (ML) algorithms, were the state of the art for many language understanding tasks up until the late 2010s, transformer-based language models have taken over the field since [3]. Especially large, generative pre-trained transformer models such as OpenAI's GPT [4–6] and DeepSeek [7, 8] excel in solving complicated semantic tasks with little to no task-specific fine-tuning. Their great advantage over previous methods can be mainly attributed to two reasons. Firstly, they employ an attention mechanism that enables the model to grasp interdependencies and semantic relations between different words or concepts in the input. It also allows for efficient parallel processing of input sequences. Secondly, their architecture can be easily scaled to comprise large numbers of parameters without significant loss in computational performance, thereby enabling them to implicitly store large amounts of knowledge and even develop emerging task-solving capabilities that they were not explicitly trained for.

While current large language models (LLMs) achieve remarkable results on many real-world use cases that require a deep semantic understanding, such as Natural Language Inference or Causality Detection, there are still some drawbacks. For one, LLMs tend to perform less accurately for languages and industry use cases where not much training data is available. Depending on the situation, this can have devastating consequences and render these models unsafe for use in many application scenarios. Secondly, state of the art models rely on extensive computing infrastructure for training and inference. This poses problems with respect to the applicability of those models, if only limited resources are available. Also, it is not favorable from a sustainability point of view, as training those models requires a high amount of energy. At the same time, methods that exploit linguistic features such as part of speech and semantic meaning are relatively under-researched in the context of language model training, even though there would be many possibilities to employ them. Those issues give rise to the idea of systematically utilizing linguistic concepts, in order to improve performance in low-resource scenarios, make language modeling more resource-efficient, and possibly help with understanding and forging the concepts an LLM uses in comprehending and producing language.

Considering that humans learn foreign languages not by memorizing huge amounts of text, but by applying syntactic and morphological rules and relating words in the new language to already

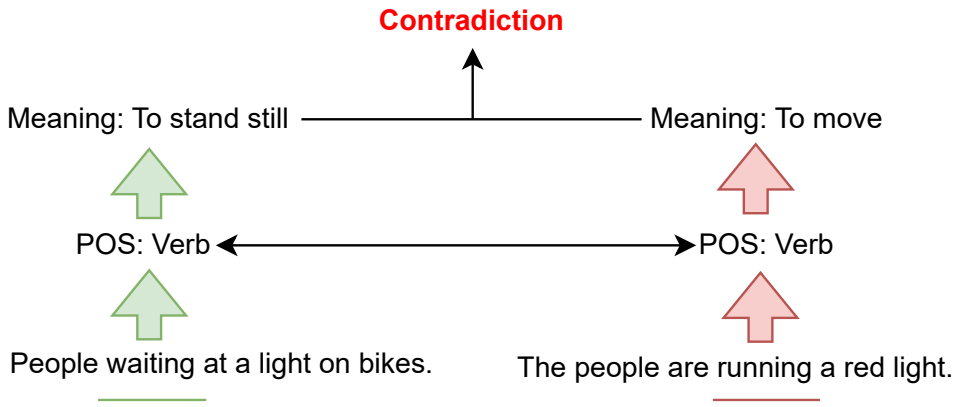


Figure 1.1: Exemplary sentence pair for showcasing the advantage of linguistically informed approaches. In order to comprehend the contradiction between the two sentences, the system first needs to find the two main verbs in the sentences and align them to each other (e.g. via Part-of-Speech-Tagging). It needs to further comprehend the contextual meaning of the two verbs, and draw the conclusion that there is a contradiction, based on the semantic concepts it has learned during training. This process of language understanding can be supported and optimized using linguistically aware methods for data augmentation and model training.

known concepts, it is a promising idea to systematically incorporate linguistic knowledge into the learning process. Previous work in this direction is mainly focused on two different methodologies: Augment data sets and enrich them with linguistic information [9], e.g. by leveraging the power of large language models [10], or infusing the linguistic knowledge during the training process [11, 12]. In this work, we comprehensively examine both directions, with the goal to identify effective methods for enhancing language modeling with semantic, syntactic and morphological knowledge. We investigate the use of LLMs to produce prototypical training data and explicit knowledge formulations, which can be used to improve performance and efficiency on Natural Language Understanding tasks. Furthermore, we use linguistically informed pre-training approaches to infuse languages models with additional knowledge during training. The motivation for using linguistically informed approaches is illustrated in figure 1.1.

Extracting and utilizing linguistic features has so far mostly been attempted in the context of simple ML algorithms and rule-based approaches. We argue that combining the power of large generative models with explicit knowledge about the structure of language can help lifting NLU algorithms to a new level of semantic insight, and further bridge the gap between human and artificial intelligence (AI) in terms of a general world understanding.

1.2 Problem Statement and Challenges

As outlined above, there are some issues and challenges that motivate the implementation of informed methods for training (large) language models. Even though it seems like transformer models can in principle be scaled to arbitrarily large sizes, just by adding more parameters and training data, it is not necessarily possible or meaningful, and in some regards even harmful to do so.

This leads to our main problem statement: **Can we systematically utilize linguistic concepts to train streamlined models for low-resource languages and specialized industry domains?** In order to answer this question, we investigate a number of different methods for linguistically informed, resource-aware LM training. There are four main challenges we identify to this end, which we plan to address in the scope of this thesis.

Challenge 1: Unavailability of Data and Resources for LLM Training In many application scenarios, it is not possible to train language models of larger size. This could be due to lack of high-quality data, compliance constraints (meaning the model has to be trained locally or on redacted data, which makes the training more complex), unavailability of IT infrastructure, or any combination of those reasons. Especially smaller companies are effectively not able to compete with large players in the LLM business for this reason. In order to democratize access to LLM training capabilities, it would be favorable to investigate methods for more data- and resource-efficient training.

Challenge 2: High Computing Cost of LLM Training Computing resources for LLM training are generally associated with high costs. For pre-training an LLM with multiple billion parameters in a reasonable amount of time, large server farms with hundreds of high-performance GPUs are necessary. This of course implies an enormous amount of electricity consumption and, given the currently available options for producing energy, some substantial amount of CO_2 being released into the atmosphere. Considering the global sustainability development goals defined by the United Nations, it is necessary to reduce the world's energy consumption as far as possible. This being merely an ethical and idealistic goal for many companies at the moment, it is possible that monetary penalties might be associated with it in the future, rendering this also a financial issue. Under these circumstances, it is necessary to drastically reduce the amount of energy needed for LLM training and inference, i.e., build models with considerably less parameters that require fewer data points for training.

Challenge 3: LLMs for Low-Resource Languages While the majority of NLP focuses on English and a few other frequently spoken languages, there are considerably less resources (both in terms of high-quality data and pre-trained models) available for languages such as Arabic, Turkish and - in some regards - even German. The reasons for this are manifold. Firstly, as English is the most popular language on the planet and de facto "lingua franca", naturally, the vast majority of publicly available text to date is in English language. Secondly, languages such as (e.g.) Arabic or Turkish come from different language families (namely, Afroasiatic and Turkic, as opposed to Indo-European, which is the root of most Western language spoken today). This implicates the usage of a completely different vocabulary, grammar, and (in case of Arabic) even a different alphabet and writing convention. Thus, it is not at all easy, and in many cases impossible to transfer models that were trained on English or other Indo-European languages to those foreign systems. In order to make LLMs accessible to all of the world's population, this aspect urgently needs to be addressed.

Challenge 4: Streamlined LLMs for Industry Use Cases Most industry scenarios require fine-tuned LLMs that solve specific problems. Consider for example the problem of finding causal or contradictory statements in a financial report. While this task can be tackled with relatively low effort if we look at generic, open-domain texts, it gets considerably more complex when a specific

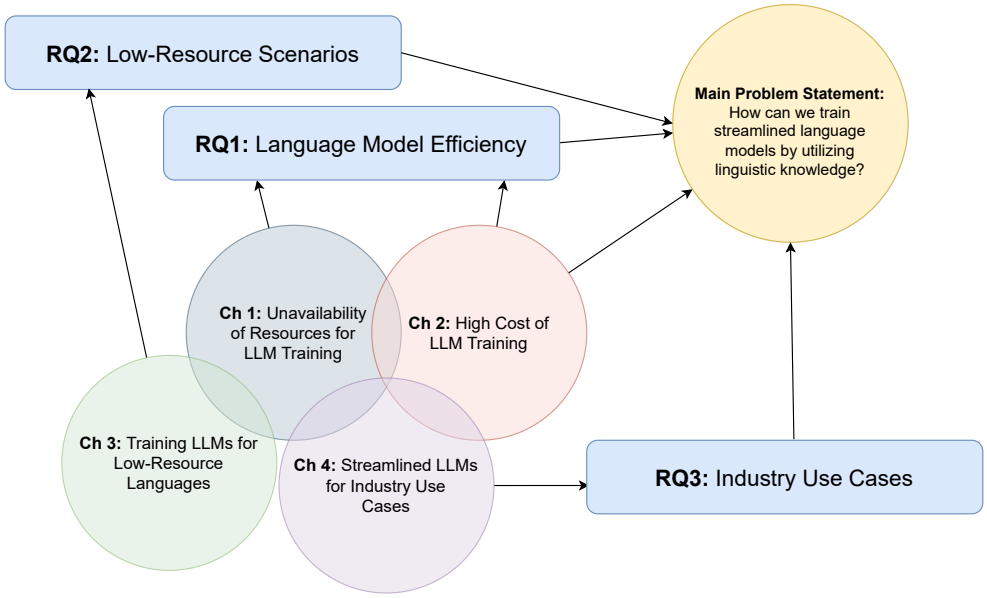


Figure 1.2: Illustration of the relations between the challenges and research questions, as well as the main problem statement.

type of language (e.g. accounting terminology) is involved. Even for a human, it would take some training to fully grasp the content of such a text, and be able to adequately answer questions about its meaning. Using pre-trained commercial models without customization is therefore often not sufficient in these cases, as those are not adapted to the domain-specific language and cannot capture subtle nuances that might be important. Also, training customized LLMs might not be possible due to lack of data and resources, and/or compliance issues (see Challenge 2). This, again, gives rise to the idea of finding methods for more efficient and task-oriented training of smaller LMs, that can be deployed in low-resource scenarios.

1.3 Research Questions

Given the challenges outlined above, as well as our main problem statement, we devise the following three research questions. See figure 1.2 for an overview on how the problem statement, the challenges and the research questions relate to each other.

Research Question 1 (RQ1)

Can training with linguistically informed objectives and/or data augmentation improve language model efficiency in terms of training set size, model size and/or training time, while maintaining downstream performance?

One main drawback of current LLMs is their extensive usage of resources, both with respect to

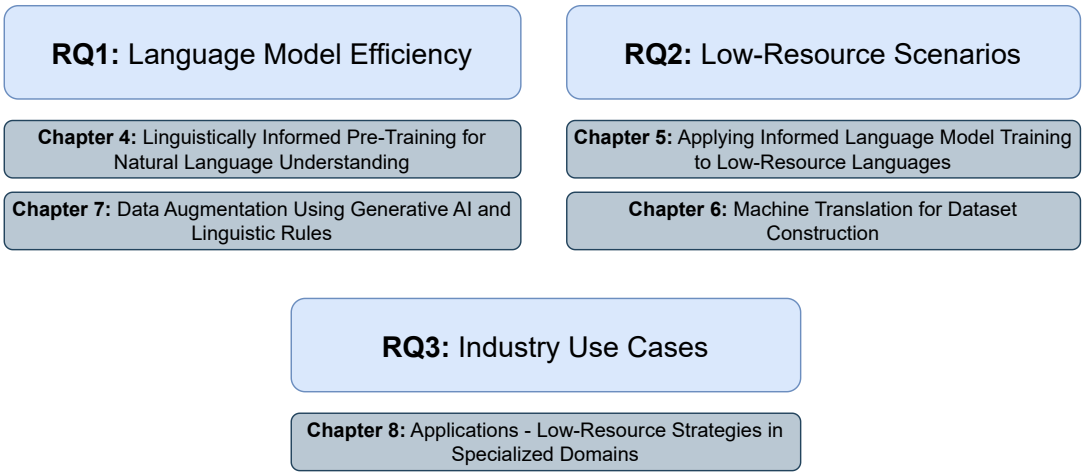


Figure 1.3: Overview of our three research questions, together with the chapters in which those are addressed.

energy consumption, as well as to the amount of data and time needed for model training (see section 1.2, challenges 1 and 2). In the scope of this thesis, we analyze whether infusing LM training with linguistic knowledge can help alleviate those concerns and allow for training of more light-weight models. We investigate two directions of research in this regard. Firstly, we employ methods for integrating linguistic knowledge into the language model during training, i.e. using informed objectives for (continued) pre-training. Secondly, we look into data augmentation based approaches, namely designing and manipulating the training data in a way that would help the model recognize linguistic properties of specific phenomena (such as contradictions or translation errors). In both cases, the goal is to encode information about sentence structure and semantics more efficiently, such that the model is able to develop its language understanding capabilities in a guided fashion, and thereby loses its dependency on huge computing infrastructure and data sets for training to some extent.

Research Question 2 (RQ2)

How do smaller, language-specific models trained with linguistically informed objectives and/or data augmentation perform compared to larger language-agnostic models for low-resource scenarios?

Another shortcoming of LLMs to date is their underperformance on languages that are less represented in online corpora, and/or harder to learn because their syntactic structure is very different from English and other Indo-European languages (see challenge 3 from section 1.2). We investigate informed approaches to teach models the intricacies of those low-resource and (in the context of NLU) under-researched languages without relying on large training corpora. Similar to RQ1, our hypothesis is that training on condensed knowledge and/or with informed objectives will help the model learn the languages' characteristics more quickly, leading to comparable or even better results than if trained on a huge corpus.

Research Question 3 (RQ3)

How do linguistically informed methods influence language modeling and understanding performance in real-world industry use cases?

Similar to low-resource languages, specific industry use cases often come with a custom vocabulary and intricate terms that are not easily understandable for a non-expert (see section 1.2, challenge 4). In many of those cases, it is therefore not sufficient to perform transfer learning or few-shot prompting with a pre-trained language model. Instead, we need to devise methods for teaching a model the terms and peculiarities of a specific domain explicitly. We hypothesize this can be achieved using similar approaches as for low-resource languages, namely employing data augmentation and informed training methods that are streamlined to the downstream use case. We will be focussing on financial text in this thesis, as we have access to valuable expert knowledge through industry projects in this domain. Nevertheless, the results would be applicable to other fields (such as law or medicine) as well, given the respective domain expertise.

To answer those three research questions, we develop and compare different methods for incorporating linguistic knowledge, with the goal to determine which of those approaches prove effective in achieving our objectives. Among those are linguistically informed pre-training using POS-Tagging, synset and syntactic parent prediction, as well as data augmentation using machine translation, linguistic heuristics and generative LLMs. We apply those methods in different low-resource scenarios to test their efficiency and robustness.

1.4 Thesis Overview

In the following, we outline the contributions of this thesis with respect to the four previously formulated research questions, and relate them to the respective publications that were compiled in the scope of this work. An overview on the relation between research questions and chapters can be found in figure 1.3.

1.4.1 Contributions

In our work, we focus mainly on three downstream tasks: Natural Language Inference (NLI) - or more specifially Contradiction Detection (CD) -, Critical Error Detection (CED) in machine translations, and Causality Detection, as those are particularly hard NLU problems that require a high amount of syntactic and semantic understanding, as well as a certain degree of world knowledge [9, 13]. Also, those problems are very relevant in many industrial use cases. While the NLI problem is quite well-researched with respect to the English language, there is little to no existing work on other languages, especially low-resource ones such as Arabic. For CED, we face a similar problem, as not much attention has been paid to this topic in general, apart from a shared task at the WMT conferences in 2021 and 2022 [14]. Causality Detection - especially with respect to financial text - has not received a lot of attention either, except for the Fincausal challenge at the Financial Narrative Processing Workshop 2020, 2022 and 2023 [15–17] which covers exactly this problem.

Contributions for RQ1

A toolset of linguistically informed training and data augmentation methods, that is being evaluated on a variety of downstream tasks.

Corresponding work: Sifa et al. [18], Pielka et al. [19], Pielka et al. [20], Pielka et al. [21], Pielka et al. [10]

With respect to NLI, we pursue this goal by collecting our own data set for the German language [18, 19], using a machine translation driven approach, and training dedicated models. We find that although current language models already achieve good performance on this task, there are still drawbacks with respect to complicated syntactic and semantic phenomena such as metaphors and world knowledge [20]. To this end, we investigate multi-lingual models such as XLM-RoBERTa [22] and pre-train them for linguistically informed objectives such as Part-of-Speech (POS-) Tagging, syntactic parent prediction [23] and synset prediction based on semantic webs, before fine-tuning on the downstream task [21].

Another direction of research in this regard is the augmentation of data sets using linguistic features and generative AI, creating synthetic training samples for the CD task. Our goal is to obtain a data set of prototypical contradictory statements, based on a linguistic typology [10]. To achieve this, we apply linguistic rules to produce contradictions by negations, antonymity or numerical mismatch. Additionally, we use large generative language models to create contradictions of more complex types. We want to show that by training a model on a data set that has been augmented with those prototypical contradictions, less resources are needed in terms of training data, training time and model size.

Contributions for RQ2

Development and evaluation of linguistically informed methods for low-resource languages.

Corresponding work: Saad al Deen et al. [24], Pielka et al. [18], Pielka et al. [19], Pielka et al. [20]

With respect to low-resource languages, we investigate language- and task-specific models for Natural Language Inference in Arabic. [20] This is a particularly interesting task, as there has been little previous work and no comprehensive large-scale evaluation on this topic to date. We collect a data set from different sources, which are partially human and machine translated from English sources, and partially original Arabic texts. On this data, we train two transformer-based models, namely AraBERT [25] and XLMRoBERTa, and compare their performance with and without linguistically informed pre-training.

Our research on CD in German (see contributions for RQ1) can also be seen as relevant w.r.t. this RQ, as there are relatively few training resources for this language and problem. We provide a large-scale training corpus, as well as some investigation on linguistic phenomena the model struggles with in the context of real-world data.

Contributions for RQ3

Development and evaluation of linguistically informed methods for applications in the financial domain.

Corresponding work: Pielka et al. [26], Deußer et al [27]

As for applications, we include three real-world use cases from the financial domain. The first one is about detecting causality in financial text, using a sequential bi-directional language modeling approach [26, 28]. For the second use case, we train a transformer network to identify translation mistakes in bilingual (German-English) financial reports [29]. The third one comprises finding contradictions in financial reports [27], transferring our learnings from domain-agnostic NLI directly into industry practice. All three applications are especially relevant for financial consultants and auditors, and the second and third one have already successfully been deployed in an industrial setting. We also inspect whether smaller task-specific models can succeed in solving specific industry problems, alleviating the need for large multi-purpose models and giving the customers complete control over the deployment setup.

1.4.2 Publications

This thesis is based on a number of publications that have been compiled over the course of the last years, and that serve as cornerstones for this work. Every one of those publications has contributed significantly to its content and outcomes.

- *Peer Reviewed*
  1. Rafet Sifa, **Maren Pielka**, Rajkumar Ramamurthy, Anna Ladi, Lars Hillebrand, and Christian Bauckhage. 2019. "Towards Contradiction Detection in German: a Translation-Driven Approach." In 2019 IEEE Symposium Series on Computational Intelligence (SSCI), pages 2497-2505, Xiamen, China. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/SSCI44817.2019.9003090
  2. **Maren Pielka**, Rafet Sifa, Lars Hillebrand, David Biesner, Rajkumar Ramamurthy, and Anna Ladi. 2020. "Tackling Contradiction Detection in German Using Machine Translation and End-to-End Recurrent Neural Networks." In 2020 25th International Conference on Pattern Recognition (ICPR), pages 6696-6701, Milan, Italy. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/ICPR48806.2021.9413257
  3. **Maren Pielka**, Rajkumar Ramamurthy, Anna Ladi, Eduardo Brito, Clayton Chapman, Paul Mayer, and Rafet Sifa. 2020. "Fraunhofer IAIS at FinCausal 2020, Tasks 1 & 2: Using Ensemble Methods and Sequence Tagging to Detect Causality in Financial Documents." In Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 64-68, Barcelona, Spain. Association for Computational Linguistics.
  4. **Maren Pielka**, Felix Rode, Lisa Pucknat, Tobias Deußer, and Rafet Sifa. 2022. "A Linguistic Investigation of Machine Learning based Contradiction Detection Models: An

Empirical Analysis and Future Perspectives." In 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), pages 1649-1653, Nassau, Bahamas. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/ICMLA55696.2022.00253

5. Tobias Deußer, **Maren Pielka**, Lisa Pucknat, Basil Jacob, Tim Dilmaghani, Mahdis Nourimand, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. 2023. "Contradiction Detection in Financial Reports." In Proceedings of the Northern Lights Deep Learning Workshop 2023, Tromsø, Norway. Septentrio Academic Publishing. DOI: https://doi.org/10.7557/18.6799
6. **Maren Pielka**, Svetlana Schmidt, Lisa Pucknat, and Rafet Sifa. 2023. "Towards Linguistically Informed Multi-Objective Transformer Pre-Training for Natural Language Inference." In Advances in Information Retrieval (ECIR 2023), pages 553–561, Dublin, Ireland. Springer. DOI: https://doi.org/10.1007/978-3-031-28238-6_46
7. Mohammad Majd Saad Al Deen, **Maren Pielka**, Jörn Hees, Bouthaina Soulef Abdou, and Rafet Sifa. 2023. "Improving Natural Language Inference in Arabic Using Transformer Models and Linguistically Informed Pre-Training." In 2023 IEEE Symposium Series on Computational Intelligence (SSCI), pages 318-322, Mexico City, Mexico. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/SSCI52147.2023.10371891
8. **Maren Pielka**, Svetlana Schmidt, and Rafet Sifa. 2023. "Generating Prototypes for Contradiction Detection Using Large Language Models and Linguistic Rules." In 2023 IEEE International Conference on Big Data (BigData), pages 4684-4692, Sorrento, Italy. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/BigData59044.2023.10386499
9. **Maren Pielka** and Rafet Sifa. 2024. "Insights About Causalities in Financial Text - Towards an Informed Approach." In 2024 IEEE International Conference on Big Data (BigData), pages 8801-8804, Washington, USA. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/BigData62323.2024.10825863
10. **Maren Pielka**, Marie-Christin Freischlad, Svetlana Schmidt, and Rafet Sifa. 2025. "Improving Language Model Performance by Training on Prototypical Contradictions." In Advances in Information Retrieval (ECIR 2025), pages 148-155, Lucca, Italy. Springer. DOI: https://doi.org/10.1007/978-3-031-88714-7_12
11. **Maren Pielka**, Max Hahnbück, Tobias Deußer, Daniel Uedelhoven, Moinam Chatterjee, Vijul Shah, Osama Soliman, Jannis von der Bank, Writwick Das, Maria Chiara Talarico, Cong Zhao, Carolina Held Celis, Christian Temath, and Rafet Sifa. 2025. "Automating Translation Checks of Financial Documents Using Large Language Models." In Language Resources and Evaluation (2025). Springer. DOI: https://doi.org/10.1007/s10579-025-09862-z

A detailed list of all publications that were compiled during the work on this thesis can be found in the Appendix B.

1.5 Thesis Outline

The thesis is structured into the following six chapters (excluding this introduction):

Chapter 2 - Preliminaries This chapter provides the methodological framework necessary for understanding the content of this thesis. We first provide an overview on Machine Learning methods as well as the notion of Neural Networks and Deep Learning. Then we introduce the topic of Natural Language Processing, outlining its progress over the course of the last decades, followed by an in-depth introduction of transformer models, including popular implementations and training paradigms. We further discuss some linguistic concepts that are relevant for our research, namely part of speech, semantic webs (including the notions of synonymity, antonymity and synsets), contradictions and causality. Finally, we introduce the language understanding tasks that this thesis will be focusing on: Natural Language Inference and Contradiction Detection, Critical Error Detection and Causality Detection.

Chapter 3 - Background In this chapter, we introduce the notion of Knowledge Distillation and delve into previous work on utilizing linguistic knowledge for Natural Language Processing. It serves as a backbone for the ideas that we will be developing in the remainder of this thesis. Related work on this subject can be roughly split into three main paradigms, namely simple feature- and rule-based approaches, data augmentation approaches, and infusion of knowledge during training. While we will be mostly focussing on the latter two for our research, we are also including the earlier feature-based ones for completeness, and because they are forming the foundation for many of the more sophisticated methods.

Chapter 4 - Linguistically Informed Pre-Training for Natural Language Understanding In this chapter, we explore how linguistically informed LM training methods can be used to train streamlined and more efficient models. This research direction is motivated by an empirical analysis of existing approaches, which underlines the lack of methods that explicitly take linguistic features into account. Specifically, we introduce an informed approach for continued pre-training of encoder-based LMs, which uses information about part of speech, syntax dependencies and synsets. This approach is further evaluated on applications in low-resource specialized domains (see chapters 5 and 8).

Chapter 5 - Applying Informed Language Model Training to Low-Resource Languages This chapter provides a first real-world application for our informed training routines, namely Natural Language Inference in Arabic. As there are not many existing resources for this task, we collect our own training data set from different sources, and apply an informed pre-training approach which is inspired by the methods that were introduced in the previous chapter. We thereby provide a significant contribution to advancing NLU and NLI research for the Arabic language.

Chapter 6 - Machine Translation for Dataset Construction In this and the following chapter, we present a number of approaches for creating and augmenting data sets with the aim of simplifying and enhancing LM training. The first method is based on Machine Translation, namely creating a German version of the SNLI data set using an automated translation engine. This helps us to train more potent models for the German language, as existing resources for this specific task are sparse.

We experiment with different featurization methods and model architectures, and provide a detailed evaluation.

Chapter 7 - Data Augmentation Using Generative AI and Linguistic Rules We further investigate approaches for linguistically informed data generation, leveraging knowledge about linguistic features such as antonymity, negations and sentence structure, in the context of Contradiction Detection. We also exploit the potential of current large language models to implement more complex linguistic concepts, and to increase the diversity of the generated data. The resulting data set is used to augment existing corpora such as SNLI, in order to achieve state-of-the-art results while signifcantly decreasing the overall data set size.

Chapter 8 - Applications for Low-Resource Strategies in Specialized Domains This chapter focuses on real-world problems from the financial domain, assessing the performance of our previously developed methods in a realistic scenario. We focus on the tasks of Causality Detection, Critical Error Detection and Contradiction Detection, as those are particularly relevant and complex problems that arise in the context of financial report analysis. The performance of our approaches is evaluated using feedback from experts in the financial domain.

Chapter 9 - Conclusion In the final chapter, we revisit the outcomes of this dissertation and draw a conclusion with respect to our previously formulated research questions. We also outline the limitations of our work, as well as perspectives for future research directions.

CHAPTER 2

Preliminaries

This chapter provides a comprehensive summary of the theoretical concepts that are necessary for understanding the content of this thesis. We will first introduce the concepts of Machine Learning (ML), and particularly Neural Network (NN) based Deep Learning (DL). Following this, we delve into Natural Language Processing (NLP), describing its development from simple bag-of-words based and statistical methods to the DL paradigms used to date. We also introduce the transformer architecture and its key component, namely multi-head attention, which lays the foundation for most of the currently used NLP approaches. Then we approach the topic of language understanding from a different perspective, by introducing the linguistic concepts that are being exploited and examined throughout this thesis. To conclude this chapter, we discuss the three main Natural Language Understanding (NLU) tasks that will be the focus of this work, namely Contradiction Detection, Critical Error Detection and Causality Extraction.

2.1 Machine Learning and Neural Networks

The paradigm of Machine Learning emerged in the second half of the 20th century, with the objective to develop algorithms that can recognize patterns in data. Technically, ML can be described as a sub-field of Artificial Intelligence (AI), which means any kind of machine or algorithm that behaves in an intelligent way. Instead of explicitly programming a certain kind of behavior, ML methods employ a set of trainable parameters which are incrementally adjusted throughout the learning process, inferring knowledge from a training data set which is assumed to be representative of the underlying problem. Those methods can thus be described as mathematical models that are being optimized to approximate an objective function $f : X \rightarrow Y$ which optimally describes the distribution of the data, where X is the set of input data points $x \in X$ and Y the set of all possible outputs $y \in Y$. This function or hypothesis is unknown at training time and can only be estimated by minimizing the loss or empirical risk using a specific data set. After training, the model is usually evaluated on a dedicated test set which was not observed during training, in order to test its generalization performance.

Common problems in ML model training are over- and underfitting. Overfitting means that the model learns the characteristics of the training set by heart and fails to transfer this knowledge to new data, leading to suboptimal generalization performance. This might occur if the model is overly complex for the task, or if it is trained on similar data for too long. In case of underfitting, the model doesn't learn any conclusive patterns, possibly because of low data quality or quantity, or because

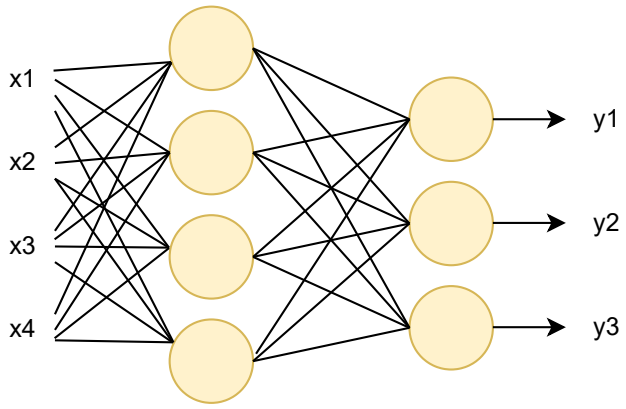


Figure 2.1: Exemplary depiction of a feed-forward NN with four input features, one hidden layer, and three outputs.

the model is too simple for the problem it is being trained for. In both cases, measures need to be taken to improve either the data, the modeling approach, or both. For this reason, training ML models requires both statistical and mathematical knowledge, as well as some understanding of the underlying problem and its characteristics.

To this end, different concepts have been developed over the last decades, ranging from simple and interpretable Decision Trees over Support Vector Machines to artificial Neural Networks (NNs). To date, NNs are the most popular ML method and widely used for a variety of use cases, due to their flexibility and scalability. Being universal function approximators, NNs are theoretically able to learn any arbitrary mathematical function, given they are provided with enough data and computational resources. They comprise one or multiple layers of **neurons**, where each neuron implements a weighted sum of the previous layer's outputs (or, in case of the first layer, of the inputs), followed by a non-linear **activation function**. Common activation functions are, for example, the rectified linear unit:

$$\text{ReLu}(x) = max(0, x) \tag{2.1}$$

or sigmoid:

$$\text{sig}(x) = \frac{1}{1 + e^{-x}}. \tag{2.2}$$

Non-linearity is important in this context, as it enables the NN to learn more complex functions. Many real-world tasks are not solvable using NNs with only linear transformations. A prominent example for this fact is the XOR-problem, where four points in a two-dimensional space are assigned to two classes in a way that it is not possible to separate them using a linear decision boundary. In higher-dimensional spaces, such as those used for most NLP tasks, it is crucial to employ non-linearity in order to find optimal solutions.

The weight matrices and biases of the NN are trainable parameters that are being adjusted while the network is trained for a task. For an exemplary depiction of a simple NN, see figure 2.1.

The process of adapting the parameters in an NN during training is commonly done via **back-propagation of error**. This means that in each training step, one or multiple samples are fed to the LM, and the error w.r.t. the gold annotation and an **objective function** is being measured. A common objective function is the mean squared error (MSE), which is defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{2.3}$$

n being the number of samples, Y the gold labels, and $\hat{Y}$ the predicted labels.

As this function measures the error or **loss** for a given batch of samples, the training objective is to minimize this value. This is achieved using **gradient descent** [30] and backpropagating the error layer by layer, starting at the output. In each step, the gradient of the objective function at a specific neuron is calculated and the parameters of the neuron's input function are adjusted in the direction of decreasing error, which can be inferred from the gradient. For more detail on this procedure please refer to [31] and [32]. Some hyperparameters such as learning rate, weight decay and dropout can be configured in order to optimize the training process for a specific task. By choosing a higher learning rate, the changes to the parameters based on backpropagation of error will be larger, resulting in faster convergence. On the other hand, a low learning rate increases the chance of finding a good solution, while convergence will be slower and it is possible that the model gets stuck in a local minimum. In practice it can be favorable to gradually decrease the learning rate during training in order to find an optimal solution. Weight decay and dropout are used for regularization, preventing the model from overfitting to the training data by inducing some amount of randomness to the weight updates.

Commonly, the output values y of an NN are normalized using a softmax function:

$$softmax(x) = \frac{e^{y_j}}{\sum_{k=1}^{K} e^{y_k}}, \tag{2.4}$$

mapping them to the interval $(0, 1)$ and enforcing their sum to be 1. This step facilitates further processing, as the output values can be interpreted as probabilities over all possible outcomes.

In **Deep Learning** (DL), NNs with a large number of neurons and layers are being trained. This is the state of the art for most relevant use cases to date. Popular examples for DL architectures are Multi-Layer Perceptrons (MLPs), Convolutional Neural Networks (CNNs) [33], Recurrent Neural Networks (RNNs) [34], autoencoders [35] and transformer models [3]. Especially in transformer models (see subsection 2.3), emergent capabilities can be observed when the number of trainable parameters is scaled to multiple billions. While CNNs and autoencoders are still widely used e.g. for vision tasks, most use cases including written text are commonly tackled by transformer models nowadays.

2.2 Natural Language Processing

Natural Language Processing (NLP) is the subfield of ML which covers all use cases related to written text. Examples for those are Sentiment Analysis [36], Named Entity Recognition [37], Contradiction Detection and Critical Error Detection (see also section 2.5), and text generation. This research area has been actively explored since the 1970s, with the earliest methods being **bag-of-words** based, such as Tf-idf [1]. In this paradigm, pieces of text are represented as high-dimensional, sparse vectors,

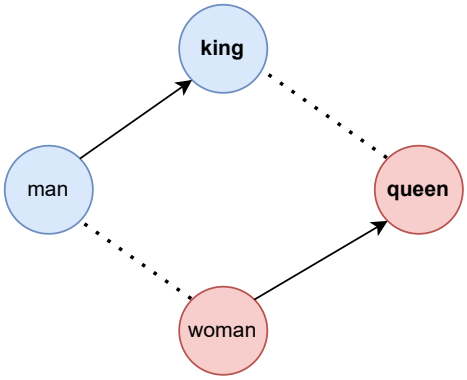


Figure 2.2: Exemplary visualization of the distances between the representations for “man”, “woman”, “king” and “queen” in a semantic embedding space.

where each entry refers to a word from the vocabulary. Only if a word is present in the text, the respective entry will be non-zero. While this approach is quite effective and sufficient for many NLP tasks, it has significant shortcomings. First and foremost, the bag-of-words representation does not take word order into account, thereby losing important contextual information. Also, the vectors do not preserve any kind of semantic meaning, as they only encode the occurence or non-occurence of words in a specific piece of text.

In order to address these shortcomings, more sophisticated methods were developed over the last 10-20 years. Paradigms such as Word2Vec [38] and GloVe [2] encode single words in a relatively small (300 - 500 dimensions) embedding space. The representations are obtained based on word co-occurences in a training corpus, with the assumption that words which appear in the same context are semantically similar and should receive similar embeddings. This process can be treated as a statistical optimization problem (in case of GloVe), or as an ML task (in case of Word2Vec). In the resulting representation space, it is possible to roughly infer certain semantic relationships. For example, the words “king” and “man” are likely to have a similar distance vector as “queen” and “woman” (see figure 2.2).

While semantic embeddings such as Word2Vec and GloVe achieve remarkable results on many NLP tasks, they were mostly replaced by contextual embedding approaches to date, especially since the development of potent DL-models. In this paradigm, word embeddings are learned end-to-end by an NN (e.g. an RNN or transformer model). This allows for more flexible, context-aware embeddings that can be fine-tuned for a specific downstream task. Examples for contextual embeddings are ELMo [39], Flair [40] or BERT [41]. Especially the advent of transformer models led to contextual embeddings becoming the state of the art.

2.3 Transformer Models

Historically speaking, transformer models replaced RNNs as the state of the art for most NLP tasks. While the recurrent architecture showed good performance in processing text due to its sequential

nature, the main advancement that significantly boosted performance was the introduction of the **attention mechanism**. It enables the model to augment word representations not only by taking into account their context, but by weighting different parts of that context w.r.t. their importance for a specific word. The transformer architecture is based on the assumption that attention is the key feature in processing natural language [3]. By applying it repeatedly, both in parallel and in consecutive layers, it becomes possible to capture different levels of semantic meaning and encode them into the word representations.

As a pre-processing step, a text has to be split up into **tokens** in order to be processed by the transformer. A token can be a word or part of a word, depending on its significance in a given context. For example, in the sentence "This is a tokenized text", the word "text" would be treated as a single token, while "tokenized" would be split up into "token" and "ized". The splitting is done using a **tokenizer**, which is a statistical model that has been fit on large quantities of text. Common tokenization methods are WordPiece [42], SentencePiece [43] and Byte-Pair Encoding [44]. They work on the assumption that frequently occuring sequences of characters should be considered as a token, while less frequent ones should be split up into multiple tokens. From a linguistic point of view, there is a rough correspondence between tokens and morphemes (the smallest entities of text carrying semantic meaning), while there is no explicit linguistic theory being applied in this process. The correct choice and parameterization of the tokenizer is crucial for the performance of the transformer model, especially when dealing with low-resource languages [45].

Since transformer models do not employ any kind of recurrent architecture, they do not have any information per se about the order of tokens in an input sequence. Still, this order needs to be considered, as it may convey crucial semantic information (see also subsection 2.2). This issue is addressed by introducing positional encodings, which are added element-wise to the input embeddings and capture their relative position in the sequence. They can be defined using sine and cosine, i.e.:

$$\mathrm{PE}(n, 2i) = \sin\left(\frac{n}{10000^{\frac{2i}{d}}}\right) \tag{2.5}$$

and

$$\mathrm{PE}(n, 2i+1) = \cos\left(\frac{n}{10000^{\frac{2i}{d}}}\right), \tag{2.6}$$

where n is the index of the input token in the sequence, d is the dimensionality of the embedding space, and i is the index of the input vector.

Another option is to train the positional embeddings alongside the other parameters of the model [46]. The authors of the paper which first introduced the transformer architecture [3] suggest to use fixed embeddings, as this allows the model to perform inference on sequences that are longer than those observed during training.

Key part of the transformer model is its attention mechanism, which allows for learning semantic relationships between words in a specific context. The idea is to obtain contextual representations of the input tokens, which capture all semantic aspects from the surrounding text, weighted by their relative importance for the respective token. To achieve this, the input embeddings are transformed into three distinct representations called **queries**, **keys** and **values**. The parameters for those linear transformations are learned during training. In a first step, the queries are compared to the keys using dot product matrix multiplication, which results in a matrix of scores that is representative of the

relative importances for all pairs of tokens from the input. Those scores are being scaled down by dividing by the square root of the vectors' dimensionality, and normalized using softmax. Optionally, masking may be applied between those steps, meaning that the weights for some word pairs will be set to 0 (e.g. to avoid attending to future tokens in text generation). Finally, they are being multiplied with the values, s.t. each resulting vector is a weighted mean of the original values, where the weights correspond to the attention scores. In formal terms, this can be written as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.7}$$

where Q, K, V are the query, key and value matrices, and d_k is the dimensionality of the key vectors. In multi-head attention, multiple of those attention heads are being applied to the input independently, and their outputs are being combined back together afterwards, commonly using concatenation:

$$\text{MultiHeadAttention}(Q, K, V) = (\text{head}_1 \oplus \text{head}_2 \oplus \ldots \oplus \text{head}_h)W^O, \tag{2.8}$$

where

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.9}$$

with parameter matrices W^O, W_i^Q, W_i^K, W_i^V, and $\oplus$ being the concatenation operator.

This allows for efficiently learning different aspects and levels of semantics in parallel. The functionality of a single attention head is illustrated in figure 2.3.

One stack of the transformer model consists of a multi-head attention layer, followed by a residual connection with element-wise addition or concatenation and subsequent normalization:

$$\text{TransformerStack}(X) = \text{LayerNorm}(\text{MultiHeadAttention}(Q_X, K_X, V_X) \oplus X), \tag{2.10}$$

where X is the input embedding matrix, and Q_X, K_X, V_X are the query, key and value representations of X.

This step can be repeated multiple times and applied consecutively to the input. The last layer is a linear feed-forward as in a classical MLP, followed by residual addition and normalization. Finally, the output is being processed by a linear output layer and normalized using a softmax function, to obtain output probabilities.

There are different learning paradigms that may be implemented using transformer models. The most common ones are text classification, token classification and text generation or sequence-to-sequence prediction. Depending on the downstream task, the architecture of the transformer model needs to be altered slightly, while leaving the key components unchanged. The three main paradigms are **encoder-only**, **encoder-decoder** and **decoder-only** models. The functionality of the encoder-only paradigm is depicted in 2.4, while the encoder-decoder and decoder-only paradigms are represented in figure 2.5.

Encoder-only models employ an additional linear layer and softmax for the output, allowing for sentence-level or token-level classification. In case of sentence classification, an extra token (usually called [CLS]) is pre-pended at the beginning of the input, and its embedding is being trained alongside the token representations. The idea is to capture the semantics of the whole input text in this token. Consequently, the output layer only takes this condensed representation as an input and disregards the

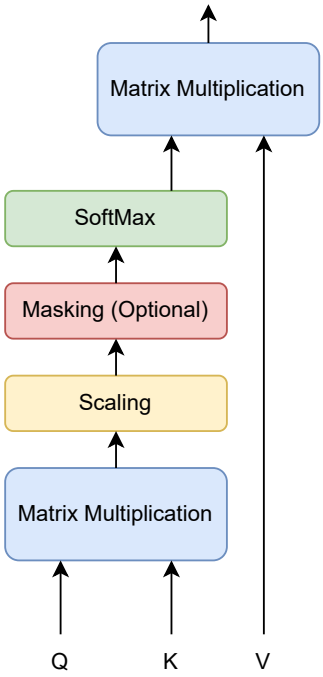


Figure 2.3: Visualization of the steps involved in calculating attention in a transformer. Q, K and V refer to queries, keys and values, respectively.

token embeddings. If the input consists of multiple sentences or text pieces that should be compared, an additional [SEP] token may be included to mark the separation between those. This setup is used, for example, in Contradiction Detection and Critical Error Detection (see section 2.5). On the other hand, Causality Extraction would be a token-level prediction task, where an output layer is being applied to each token representation from the last transformer layer, allowing for classifying each input token individually.

Encoder-decoder and decoder-only models employ an **autoregressive** mechanism, meaning that the last output token from the decoder is appended back to the input in each time step, allowing for causal language modeling (i.e. text generation). Their main difference is that in encoder-decoder models, the encoding of the input is kept separate from the output generation, meaning the decoder only has access to the final input representation from the encoder. This makes it most suitable for translation tasks, where the input distribution is significantly different from the output. On the other hand, for most text generation objectives, a decoder-only setup is an efficient approach, meaning that the concatenated input and current output would be processed together in each generation step [47].

2.4 Linguistic Concepts

In this section, we introduce four linguistic concepts that will be relevant for the research discussed in this thesis, namely part of speech, semantic webs, contradiction and causality. Understanding those will be helpful in getting a better idea about the intuition behind the developed methods, as well as the use cases we apply those on.

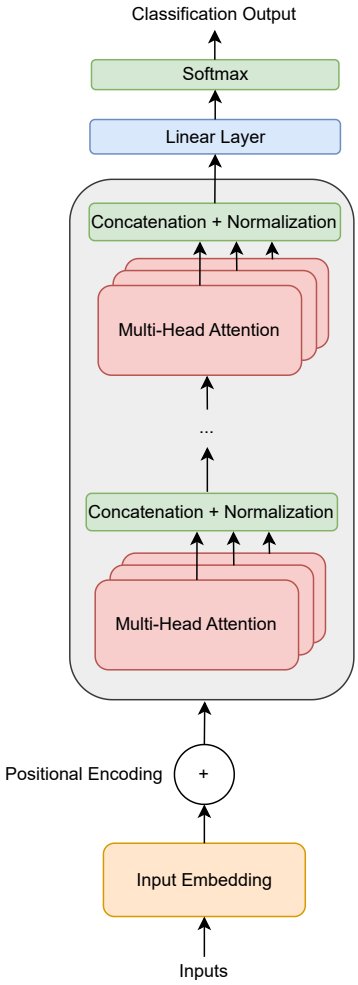
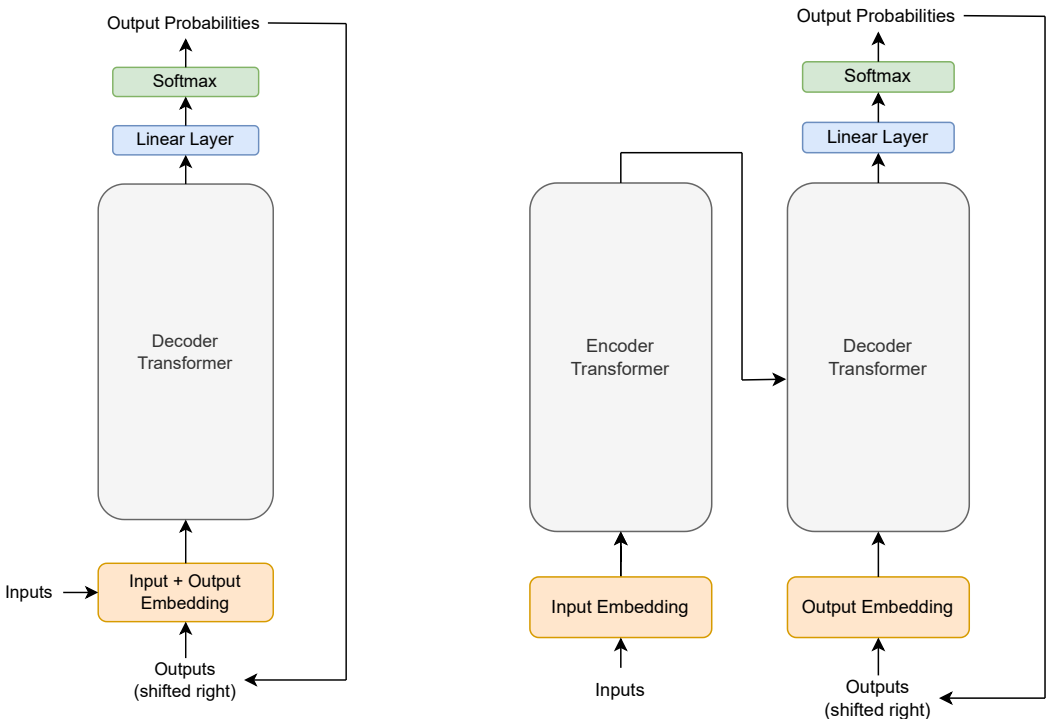


Figure 2.4: Illustration of the encoder-only architecture (used for classification tasks), including multi-head attention.

2.4.1 Part of Speech

Part of speech (POS) refers to assigning a word to its grammatical category [48]. Common POS labels are:

- noun (e.g. “house”, “dog”)
- verb (e.g. “to walk”, “to be”)
- adjective (e.g. “black”, “big”)
- adverb (e.g. “to walk”, “to be”)
- pronoun (e.g. “I”, “she”)
- preposition (e.g. “on”, “under”)



(a) Illustration of the decoder-only architecture (used for text generation tasks).

(b) Illustration of the encoder-decoder architecture (used for translation tasks).

Figure 2.5: High-level depiction of decoder-only and encoder-decoder transformer models.

- conjunction (e.g. “and”, “but”)
- interjection (e.g. “oh”, “aw”)
- determiner (e.g. “the”, “a”)

Depending on the underlying taxonomy and the required granularity, POS can be further split up into sub-categories (e.g. prepositions can be divided into prepositions of time, locality or possession). For our purposes, we stick with the more coarse definition above.

POS can be helpful in analyzing a language’s structure, as it conveys important information about the function of a word in a sentence. It is noteworthy that the POS label of a word might change depending on the context, e.g. “flight” can be a verb or a noun in different contexts.

2.4.2 Semantic Webs and Relations

Another important linguistic concept is the precise semantic meaning of a specific word. Same as for POS, this meaning might change depending on the surrounding words. Also, different words might have a similar meaning in a specific context. In order to capture those phenomena, the concept of semantic webs was developed. In these data structures, word are organized into synonym sets, or short **synsets**, where one synset conveys a distinct meaning (e.g., “to suggest” and “to propose” could be

grouped together into one synset with the meaning "to make a suggestion"). The purpose of a semantic web is also to model relationshops between words and synsets. Among the most important ones are:

- **Synonymy:** Words with the same meaning, i.e. belonging to the same synset (e.g. "obtain" ↔ "gain")
- **Antonymy:** Words belonging to synsets with opposite meaning (e.g. "black" ↔ "white")
- **Hypernymy:** Words being a more general expression (umbrella term) of others (e.g. "vehicle" → "car")
- **Meronymy:** Words signifying a part of a larger entity (e.g. "window" → "house")

It is important to note, that synonymy and antonymy are undirected relations, while hypernymy and meronymy are directed (their counterparts being hyponymy and holonymy). Also, as stated above, the same word could be sorted into different synsets depending on the context it occurs in. The task of assigning a word its distinct synset is called **Word Sense Disambiguation**. There are different approaches to tackle this problem, ranging from knowledge-based rules to supervised methods [49].

A popular semantic web for the English language is WordNet [50], which is openly available for download and over a web browser[1]. There are similar data bases available for other languages, e.g. GermaNet [51, 52] for German.

2.4.3 Contradiction

The concept of contradiction is one of the most complex linguistic phenomena. This is due to the many different circumstances a contradiction can arise from, as well as to the fact that it is highly subjective whether a specific statement shall be considered contradictory or not. There is also a certain amount of world knowledge required to truly understand the deeper meaning of some potential contradictions. One widely accepted definition of contradiction, which we are also going to adopt in the remainder of this work, is the following:

> Two statements are to be considered contradictory, if it is extremely unlikely that they could be true at the same time [9].

This, of course, is a rather open definition and leaves some room for interpretation. We consider it well-suited for our purposes, as we aim to employ an understanding that aligns as much as possible with human intuition, while slightly deviating from a strictly logical definition. This means that two sentences could still be considered contradictory, even if there is a very small chance that both could be true, but it does not align with common sense, i.e. most people would not agree to this conjecture. Consider e.g. the following example by [9]:

Premise: Sally sold a boat to John.
Hypothesis: John sold a boat to Sally.

[1] https://wordnet.princeton.edu/

Even though both events could technically have occured at the same time, it is very unlikely from a common sense perspective. Note that we are making the important assumption here, that both events are considered to occur (almost) simultaneously and are referencing the same objects and people.

All of the above mentioned facts make it very hard for an automated system to detect and understand contradictions (see subsection 2.5.1).

2.4.4 Causality

Causality describes the semantic relationship between two statements in which one can be described as the cause of the other. Same as for contradiction, we apply a more loose and intuitive definition:

> Causality means one event, process, state or object (the **cause**) contributing significantly to the occurence of another (the **effect**). [53]

This phenomenon is subject to ambiguity, and background knowledge might be required to determine whether a given statement conveys causality. Consider this example:

> After the Wall fell, living standards in East Germany rose.

In this case, it is important to understand the historic event that is being referred, as well as the possible cause-effect relation that is being implied (the reunion of Germany leading to rising living standards in the Eastern part). Causality, by its nature, always has a temporal component, which is why it may often be implicitly described as a before-after sequence of events.

2.5 Natural Language Understanding

This section covers the three main NLP tasks we will be investigating in the scope of this thesis, namely Natural Language Inference/Contradiction Detection, Critical Error Detection, and Causality Extraction. All of those are particularly hard Natural Language Understanding (NLU) problems that require a high degree of language understanding, which is why we chose them as benchmarks for our informed approaches.

2.5.1 Natural Language Inference and Contradiction Detection

Natural Language Inference (NLI), formerly also known as Recognizing Textual Entailment [54], is the task of identifying a semantic relationship between two pieces of text, commonly called **premise** and **hypothesis**. Possible relations are **entailment** (meaning the hypothesis follows from the premise), **contradiction** (the premise and the hypothesis contradict each other) or **neutral** (neither of those two relations is present). Contradiction Detection (CD) emerged more recently as a sub-field of NLI. Here, the *entailment* category is discarded, and the task is reformulated, s.t. the goal becomes to simply determine whether a given sentence pair contains a contradiction or not. From a technical point of view, NLI is a three-class and CD a two-class text classification task.

While NLI has been investigated in-depth over the course of the last years (at least with respect to the English language), CD has not been given the same amount of attention. We focus on the latter for this thesis, as we argue that contradictions arise from distinct semantic phenomena that we want

to investigate in isolation. To this end, we build upon the work of [9, 55], who derived a linguistic typology of contradictions and constructed some simple ML-based methods using this paradigm as a basis. Our work aims to extend their research by applying its findings in the context of LLMs and using it for data augmentation rather than as features for training.

A standard data set for the NLI task is the Stanford Natural Language Inference (SNLI) corpus [13], which we will be using as a main data resource throughout this thesis. It consists of more than 500 000 premise-hypothesis pairs that have been manually annotated by human workers. Annotations for NLI can very easily be converted to adhere to the CD objective, by replacing all *neutral* and *entailment* labels with a common one: *no contradiction*. We will make use of this convenient fact in multiple places of this work. Some exemplary sentence pairs from the SNLI data set are displayed in table 2.1. For more details on the data set, please refer to the respective chapters (4, 6).

Premise	Hypothesis	Label
A person on a horse jumps over a broken down airplane.	A person is training his horse for a competition.	neutral
A person on a horse jumps over a broken down airplane.	A person is at a diner, ordering an omelette.	contradiction
A person on a horse jumps over a broken down airplane.	A person is outdoors, on a horse.	entailment

Table 2.1: Exemplary sentence pairs with labels from the SNLI data set.

2.5.2 Critical Error Detection

Critical Error Detection (CED) is a sub-task of Machine Translation Quality Estimation (MTQE or just QE), and was first introduced as a distinct use case in the scope of the 2021 WMT QE Shared Task [14]. Its objective is to identify critical errors in a (machine) translated text, where "critical" means that it could have grave consequences in an application scenario if the error is not detected. In the original problem formulation, it refers to one of the following categories:

- **Toxicity:** The level of toxicity, i.e. use of harmful language and insults, is very different between the two texts.
- **Sentiment:** The sentiment, i.e. the general tone of the text (positive or negative) is very different.
- **Named Entities:** There are different Named Entities (e.g. people or places) referred to in the two texts.
- **Health and Safety Risks:** There are significantly different informations being conveyed w.r.t. health- or safety critical topics.
- **Numbers:** There are different numerical values being mentioned in the two texts.

Of course, it depends highly on the specific application, which of those categories would be considered more or less relevant. For example, toxicity is not really an issue in financial text and can for the most part be disregarded in this context. The understanding of critical and non-critical errors may also vary between application domains. Depending on the task, the problem could be formulated a bit differently. Some exemplary sentence pairs from the 2021 WMT data set are displayed in table 2.2.

Source	**Translation**	**Label**
So what you're saying is the internet police AREN'T coming to my door???	Also was du sagst ist , dass die Internetpolizei nicht zu meiner Tür kommen ? ? ?	NOT
What the fuck? Ur a dick, urv never even been 2 helensville ya mommas boi!!!	Was ist das ? Ur ein dick , urv noch nie 2 helensville ya mommas boi ! ! !	ERR
This is a fact for which reliable, published sources do exist!	Das ist eine Tatsache , für die zuverlässige , veröffentlichte Quellen existieren !	NOT

Table 2.2: Exemplary sentence pairs from the 2021 WMT Critical Error Detection [14] dataset.

Technically, CED is a binary text classification task, where the two possible labels are *critical error* and *no critical error*. It can thus be modeled in a similar way to NLI/CD.

2.5.3 Causality Extraction

In Causality Extraction (or Causal Relation Extraction) [56, 57], the goal is to extract cause and effect statements from a piece of text, a simple example being:

Original text: As climate change progresses, temperatures around the world rise to new highs.
Cause: climate change progresses
Effect: temperatures around the world rise to new highs.

Similar to CED, it is a relatively under-researched field of NLP, which only gained more attention in recent years. Nevertheless, it can be considered a core NLU task, as the comprehension of causal relationships is crucial for general text understanding. It is also a very complex problem, as causality is one of the most ambiguous and difficult linguistic phenomena, which also requires a certain amount of world knowledge (see subsection 2.4.4).

In the context of financial text, causality has been explored by the Fincausal challenge [15–17], which we will be referencing in section 8.1.

CHAPTER 3

Background

In this chapter, we provide some theoretical background on the approaches that are being explored throughout this thesis. First, we give an overview about related work on the topic of Knowledge Distillation, to lay the foundation for our proposed framework. Secondly, we introduce different methods for incorporating linguistic knowledge into LM training. Specifically, we sketch the evolution of NLU by introducing feature- and rule-based approaches, followed by data augmentation approaches, and finally current methods to train LMs in an informed fashion. To this end, we revisit previous work that laid important milestones in this area.

3.1 Knowledge Distillation: An Overview

While the extraordinary performance of generative language models can for the most part be attributed to drastically scaling up their number of parameters, there has been an opposite research direction for some years now. Given the fact that resources are limited, and many application scenarios put constraints on the size of models, researchers have been investigating how to downscale large language models, while preserving their predictive capabilities as well as possible. Those approaches can for the most part be summarized under the terms **Knowledge Distillation** and **Model Compression**. While Model Compression is a promising research direction as well, we specifically want to take a closer

Figure 3.1: Illustration of the idea behind the Knowledge Distillation paradigm. A teacher model predicts labels and/or probabilities for a number of given samples, and the student model is being trained on those predictions as ground truth labels.

look at the first approach, as it lays the conceptual foundation for what we are attempting to achieve.

The paradigm of Knowledge Distillation, which was first introduced by [58], describes a setup in which a (smaller) *student* model is trained by a larger *teacher* model. The goal is for the student to reach comparable results w.r.t. the teacher, while requiring considerably less trainable parameters and possibly achieving better generalization performance. It is commonly being trained using the logit outputs of the teacher as soft targets, and a loss function with continuous input, such as the Kullback-Leibler divergence. An illustration of the approach is displayed in figure 3.1. In the context of language, this paradigm has already been successfully applied, e.g. for the Phi3 language model [59], which was partially trained on data generated by GPT-4.

3.2 Usage of Linguistic Knowledge in Language Modeling

There are a number of different methods to incorporate linguistic knowledge into language model training. They are for the most part applied with the objective to improve the models' downstream performance, as well as their efficiency w.r.t. training and inference time, model size and resource usage. While in the early years of natural language processing, the first incentive was considered most important, the second one became more and more imminent especially with the recent rise of large language models. Also, LLMs offer new possibilities for creating large amounts of data and thereby "teaching" smaller models, as described above.

In the following subsections, we will outline relevant approaches in a chronological fashion. We start with simple feature- und rule-based approaches, which have been applied before large transformer-based models and text embeddings were available. Following that, we talk about more sophisticated data augmentation approaches using generative AI, as well as hybrid methods that use a mix of heuristics and text generation. Lastly, we look into the newly emerged field of informed language model training using linguistic concepts.

3.2.1 Feature- and Rule-Based Approaches

In the earlier days of Machine Learning based text analysis, linguistically informed methods would mostly rely on handcrafted features and rules. In the context of textual entailment, [54] present one of the first approaches to tackle the Natural Language Inference task in an automated way. They calculate a bag-of-words based similarity score, followed by a thresholding approach that yields a binary decision as to whether the hypothesis is entailed by the premise or not. The word similarities are calculated both using a dependency-based approach and one using semantic webs. They obtain an accuracy of 0.55 on the test set of the 2005 PASCAL Recognizing Textual Entailment Challenge [60], indicating that the approach does not generalize well and is highly volatile w.r.t. the choice of the classification threshold, which was done based on the development set.

In a similar context, [55] and [61] introduce an alignment approach based on semantic graphs, followed by classification using linguistic features. Their first step comprises obtaining a semantic graph representation of both premise and hypothesis, and identifying related pairs of words s.t. semantically similar subgraphs are more likely to be aligned. Based on those aligned graphs, linguistic features are extracted. Among those are indicators of polarity ("no", "not"), antonymity (do aligned words in premise and hypothesis have an antonymity relation?), factivity (embedding of the verb phrase), as well as numbers and dates. Those features can be aggregated to obtain an entailment score

providing an indication whether the two statements display a contradiction or entailment relation. In a later work [9], the authors train a simple classifier on the same features and obtain a promising performance on their handcrafted data set.

Utilizing all available semantic and syntactic knowledge for solving text understanding tasks is a reasonable approach, especially in the absence of large pre-trained language models. Nevertheless, those methods can only incorporate a pre-defined set of rules, and will not be able to detect more subtle nuances. They also tend to overfit, as their hyperparameters are very sensitive w.r.t. the choice of the development set. Furthermore, they rely on the availability of extensive knowledge bases at inference time, making them a suboptimal choice in many real-world scenarios.

While those approaches are not frequently used any more nowadays, given the recent developments in machine-based text understanding, they still offer valuable insights into the linguistic intricacies of contradictory statements. They can also serve as a backbone to more advanced methods (see the two following subsections), both as a knowledge base for LLM-powered text generation and informed LM training.

3.2.2 Data Augmentation Approaches

With respect to (informed) data augmentation and generation, a number of different approaches have been explored over the course of the last years. Some of them make use of linguistic features and rules to obtain new (labeled) data from existing corpora. For example, [62] introduce a data augmentation approach for the CED task using semantic features of synonymy and antonymy. They utilize a parallel corpus in German-English and produce training samples for CED by exchanging words in the original or translated version with either a synonym or antonym. Training a model on this semi-synthetic data shows good results, even when using a significantly smaller corpus compared to similar approaches.

Utilizing state-of-the-art LLMs for data generation is another promising approach in this context. The creators of the Phi family LMs [59, 63] make use of this approach in a large-scale fashion. They produce a huge amount (> 1B tokens) of coding textbooks and exercises generated by GPT-3.5, which they utilize to train their own, relatively small-sized (1.3-3.8B parameters) models. According to the authors, the biggest challenge is to assure that the resulting data is diverse enough. This is achieved by injecting controlled randomness into the prompts for data generation, programmatically exchanging key aspects such as topic and target audience in the model's instructions. Their approach proves to be successful in creating large quantities of highly diverse data, which is being used to augment the training data for the Phi models.

In the context of computer vision and image classification, [64] introduce the notion of training an ML model on formalized knowledge ("prototypes") instead of raw data. They demonstrate the effectiveness of this idea by augmenting existing data sets (e.g. MNIST [65] for handwritten digit recognition) with constructed prototypes, and measure the effect on training resources and model robustness. Their results show that adding prototypes in an initial training step helps reduce the overall training time and improve generalization performance. This effect is especially evident when only a small portion of the original training set is used.

We want to combine all the three above-mentioned methods for data augmentation in this work, by utilizing linguistic knowledge, generative AI and the idea of prototypical training samples in the context of Natural Language Understanding.

3.2.3 Informed LM Training Approaches

Another research direction in this regard is the idea to inject the model with additional (linguistic) knowledge during training. This is done by introducing additional pre-training objectives that transport valuable semantic or syntactic information. [11] and [12] present approaches for informed cross-lingual pre-training in the context of machine translation, by using a custom loss function that minimizes the distance between source and target embeddings in feature space. Their approach shows effective not only for translation, but for several other NLU tasks as well.

With respect to syntactic features, [23] introduce a novel pre-training regime for BERT, namely syntactic parent prediction. The model is instructed to predict the parent of each token in a dependency parse tree (see figure 3.2 for illustration). This structure can be obtained from external tools such as spaCy [66] and therefore requires no additional annotations.

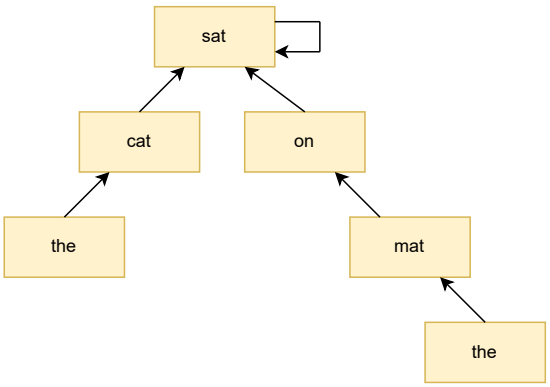


Figure 3.2: Illustration of a syntactic parse tree that is being used for parent prediction. The model is trained to predict the parent in the tree for each word (e.g., the label for “cat” word be “sat”). By definition, the main verb (in this case, “sat”) is the root of the parse tree and therefore its own parent.

The approach shows efficient in improving downstream results on multiple NLU tasks, compared to vanilla BERT [41]. This motivates further studies along the lines of this paradigm, using different linguistic features such as POS and semantic labels as targets.

Overall, there is not much research about this aspect of DL to date, as most approaches are either data-driven or focussig on architectural improvements to the models. This emphasizes the importance of further investigating that direction.

CHAPTER 4

Linguistically Informed Pre-Training for Natural Language Understanding

This chapter marks the first scientific contribution of the thesis. We start by conducting a linguistic analysis to identify the intricacies of contradictory statements that are especially hard for an automated system to identify. Based on our findings, we introduce an informed pre-training approach which aims to infuse linguistic knowledge to an LM, using semi-supervised labels for part of speech tagging, synsets and syntactic relations. Our hypothesis is, that this approach would help the LM to grasp essential features of language more easily, thereby requiring less parameters and resources for training. Thus, we are investigating the first research question (RQ1):

Research Question 1 (RQ1)

Can training with linguistically informed objectives and/or data augmentation improve language model efficiency in terms of training set size, model size and/or training time, while maintaining downstream performance?

To this end, we implement and evaluate our approach using transformer models of different sizes, and compare their performance on the SNLI test set. We also assess different combinations of pre-training objectives and determine which are the most effective ones for this specific downstream task.

The key contributions of this chapter are:

- We collect a new data set for Contradiction Detection in German language, comprising examples with real-world relevance.
- We conduct a qualitative assessment of Contradiction Detection systems with a focus on linguistic features.
- We present a novel pre-training regime for transformer models based on linguistic concepts, which can be implemented without any additional labeling effort and/or any extra data. Its effectiveness is being demonstrated on the NLI task.

The initial idea for this research direction was developed by Maren Pielka, who also conducted some part of the implementation and model training, and supervised the research work. The linguistic analysis was initially performed by Felix Rode and refined by Maren Pielka. The experiments were designed and conducted by Maren Pielka, Svetlana Schmidt and Lisa Pucknat, while Lisa Pucknat did most of the implementation work. The papers were mainly written by Maren Pielka, with contributions from Felix Rode, Lisa Pucknat and Svetlana Schmidt.

This chapter is based on the following publications [20, 21]:

- **Maren Pielka**, Felix Rode, Lisa Pucknat, Tobias Deußer, and Rafet Sifa. 2022. "A Linguistic Investigation of Machine Learning based Contradiction Detection Models: An Empirical Analysis and Future Perspectives." In 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), pages 1649-1653, Nassau, Bahamas. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/ICMLA55696.2022.00253
- **Maren Pielka**, Svetlana Schmidt, Lisa Pucknat, and Rafet Sifa. 2023. "Towards Linguistically Informed Multi-Objective Transformer Pre-Training for Natural Language Inference." In Advances in Information Retrieval (ECIR 2023), pages 553–561, Dublin, Ireland. Springer. DOI: https://doi.org/10.1007/978-3-031-28238-6_46

In the following, we will first introduce the problem and motivate our approach. Then we present our custom data set in section 4.3 and delve into the results of our linguistic analysis (section 4.4). Building upon our findings, in section 4.5 we introduce the new pre-training objectives, and outline our training pipeline. Section 4.6 summarizes the experimental results and draws a conclusion with respect to our research objective. We close the chapter with a comprehensive summary.

4.1 Introduction

Contradiction Detection and Natural Language Inference are particularly hard problems from the Natural Language processing (NLP) domain [18, 19, 67]. A variety of Machine Learning (ML) approaches have been introduced to tackle this task, the state-of-the-art being transformer-based methods such as GPT-4 [6] and XLM-RoBERTa [22, 68]. While achieving overall good classification performance, those methods still lack understanding of linguistic features, and are relying heavily on extensive amounts of data for language model pre-training. To this end, we investigate the well-known SNLI data set [13] with the aim to discover distinct linguistic properties that are important in recognizing contradictions. In addition, we also collect a data set in German language from various online sources, which is being labeled by human annotators for the contradiction detection task. The objectives of this work are two-fold:

- We want to find out, whether the types of contradictions differ between the synthetically created SNLI data, and the data we collected from online sources. This will also help us building a model that works well in a real-world application scenario.
- We want to identify relevant linguistic features, that could help an ML model learn to recognize contradictions, without relying on extensive amounts of pre-training data. To this end, we analyze the predictions of an XLM-RoBERTa model with respect to those syntactic and semantic features that lead to wrong classifications.

Based on those findings, we develop some first ideas for informed ML approaches that would help improve those results in the future, by injecting linguistic knowledge into the learning process. Our analysis is limited to the German language, but many of the results could most likely be applied to other languages as well.

To further address those shortcomings, we present a linguistically enhanced approach for multi-objective pre-training of transformer models, using the results of the linguistic analysis outlined above. We inject extra knowledge into the model by pre-training for three additional language modelling tasks, one of which is a novel approach. Specifically, we utilize external information about part of speech tags, syntactic parsing, and semantic relations between words. Our main contribution can be summarized as follows:

> We aim to become independent of huge data resources for pre-training, and having to train models with a large number of parameters, by injecting as much external knowledge to the model as possible. This goal is being quantified by evaluating our model on the Stanford Natural Language Inference (SNLI) [13] data set. We compare different implementations of the transformer architecture (BERT [41] and XLM-RoBERTa [22, 68]), with the aim to show that the smaller model, BERT, is able to perform competitively when being enhanced with additional knowledge during pre-training.

Our approach does not require any additional data or annotations for pre-training. It is pre-trained for the additional tasks on the same data set that it is later being fine-tuned on. The labels for the word-level pre-training are generated in a semi-supervised fashion, by utilizing existing models and knowledge bases for those tasks. We therefore argue, that we can achieve close to state-of-the-art performance with a comparatively small (BERT-base) model and minimal additional effort in terms of data and computation time.

4.2 Related Work

In Natural Language Inference (NLI), first introduced by [69], one has to determine whether a given *hypothesis* can be inferred from a given *premise*, or whether it contradicts the premise. Further, a new research field emerged from the NLI task named Contradiction Detection (CD). Multiple languages, besides the commonly used English language, such as Persian [70], Spanish [71], and German [18, 19, 67, 72] were studied. We follow up on the latter research, in which a portion of the SNLI dataset was machine-translated into German. They found that RNNs handle machine-translated data quite well, with difficulties in complicated sentence structures, translation artifacts, and understanding of world knowledge. A fine-tuned XLM-RoBERTa model seems to be most promising with regard to the difficulties mentioned above.

BERT [41] and XLM-RoBERTa [22, 68] are among the state of the art transformer-based encoder models for text classification tasks. They use the pre-training objectives of Masked Language Modeling and Next Sentence Prediction (only BERT) to obtain a large amount of language understanding in an unsupervised way. Dependency Injected Bidirectional Encoder Representations from Transformers (DIBERT) [23] utilizes a third pre-training objective called Parent Prediction injecting syntactic structures of dependency trees.

There is little prior work on the subject of linguistically aware modelling for Natural Language

Inference. Marneffe et al. [9] are the first to comprehensively define Contradiction Detection as a distinct problem, and present some initial methods based on handcrafted semantic features such as antonymity and polarity. They point out, that detecting contradictions is a particularly hard task, because a deep level of language understanding is needed, as well as some background knowledge that cannot necessarily be inferred from the analyzed text alone.

Li et al. [73] attempt to learn contradiction-specific word embeddings by enforcing words with opposite meaning to be mapped into different regions of feature space. This addresses the issue, that opposite words (antonyms) tend to appear in similar contexts, such that a conventional word embedding model would learn similar embeddings for those. Using this method, the authors report state-of-the-art results on the SemEval task.

The approach of integrating the external semantic knowledge into a transformer model was first presented by [74]. In their work, WordNet embeddings are combined with the BERT architecture in two ways: during *external combination* the outputs of WordNet and BERT are combined for the additional classification and in *internal inclusion* the WordNet representations are integrated into the internal BERT architecture. The resulting models were evaluated on four GLUE [75] datasets for Sentiment Analysis, Linguistic Acceptability, Sentence Similarity and Natural Language Inference tasks [74].

A more recent approach is presented by [76]. They pre-train a BERT model on five different, linguistics aware tasks such as POS-Tagging, semantic role labeling and syntactic parsing, achieving competitive results on GLUE benchmark tasks. The main difference between this work and ours is that we focus on minimizing the amount of pre-training data and model parameters, utilizing the same data sets for both custom pre-training and fine-tuning. In addition, we introduce the novel synset prediction objective. Unlike the approach of [74], we utilize only one synset extracted for each word in the data. They achieve competitive results on the GLUE benchmark.

Pucknat et al. [67] evaluate different neural network based approaches on the Contradiction Detection task in German language. While XLM-RoBERTa [22, 68] achieves the best results of all models under investigation, it still has problems with complicated syntactic structures and real-world language use. This gave rise to the idea of investigating the role of linguistic features more closely, in order to come up with an informed learning approach.

4.3 Data

For our analysis and experiments, we use two different data sources: A machine-translated version of the Stanford Natural Language Inference (SNLI) data set, and a collection of real-world examples from various online sources in German language.

4.3.1 SNLI

The Stanford Natural Language Inference (SNLI) data set was first introduced by Bowman et al. [13]. It is the largest collection of human-generated premise and hypothesis pairs for the NLI task to date, with over 570,000 examples. The data was collected in a crowd-source campaign, where both samples and labels were created by human annotators. The final labels were decided upon by a majority vote, thus minimizing noise due to human error and ambiguity. In the original data set, there are three possible labels: *entailment*, *neutral* and *contradiction*. For the purpose of the Contradiction

Detection objective, we binarize those labels by consolidating the *neutral* and *entailment* labels to *no contradiction*.

A large portion of the SNLI training set (100 000 examples), as well as the whole validation and test set were machine-translated to German [18] (see chapter 6 for more details), using the DeepL API[1]. The data set was found to be of overall sufficient quality, but there are some artifacts and inconsistencies, due to the Machine Translation and the annotation setup. Because of those issues, it is not completely representative of a real-world setting.

4.3.2 Online Data Set

To address those shortcomings, we collect a data set from various online sources in German language. Those include news [2], tweets [3], company and employer ratings [4], game reviews [5] and product reviews [6].

The data is being manually annotated by six workers using two different modes. For the first one, the annotators are presented with random examples from all five sources, and shall come up with contradicting or non-contradicting hypotheses for each of those examples. There is also the option to exclude sentences, if no meaningful hypothesis can be found. Since this procedure is quite costly, we additionally use another annotation mode, presenting the annotators with pairs of sentences from the online sources, where the premise and hypothesis have already been matched. To achieve this, the samples are first being grouped into different categories, according to the meta-data from the website (e.g. similar keywords on Twitter). Additionally, a text similarity measure is applied to identify samples that are likely to refer to the same topic. Those text pieces that belong to the same category, and show a high similarity are then being matched and presented to the reviewers as premise-hypothesis pairs. Given this setup, the annotators only have to add the respective label: "contradiction", "no contradiction" or "exclude" (for cases where the sentences do not relate to each other, or one of them makes no sense).

We create 10 000 data points using the first annotation procedure, and another 31 000 using the second approach. The 10 000 manually created examples are being reviewed by a second annotator, to minimize noise due to subjectivity and human error. After those steps, 531 samples had to be excluded, so that we end up with a total of 40 589 examples. A random 60-20-20 training-validation-test split is being applied to the remaining data set. Some examples from the internet data set can be seen in table 4.2.

4.4 Linguistic Analysis

We perform a linguistic analysis of the two data sets, focussing on the qualitative differences between SNLI and internet data, and those instances that impose problems for the classifier. For this evaluation, an XLM-RoBERTa model [22, 68] is used, which has been pre-trained for the Masked Language

[1] https://github.com/fraunhofer-iais/snli_translated
[2] https://correctiv.org, https://nachrichtenleicht.de/
[3] https://twitter.com/
[4] https://de.trustpilot.com/, https://www.kununu.com/
[5] https://store.steampowered.com/
[6] https://www.amazon.de/

Modeling task on 100 languages, and fine-tuned on the respective training set for the Contradiction Detection task (translated SNLI / online data). For details on the architecture and training procedure, we refer to section 2.3 and [67].

4.4.1 Differences between the two Data Sets

The two analyzed data sets differ primarily in syntactic structure. The first (SNLI) has basic syntactic structures and semantics as well as grammatical simplicity, whereas the second data set (internet data) virtually lives from syntactic versatility. Here, the data sets differ not only in sentence-related verbosity and sentence length but also in the juxtaposition of such sentences. The sentences from the internet data are also formulated more homogeneously and thus come much closer to real language use. Some example sentence pairs from the two data sets can be found in tables 4.1 and 4.2.

Premise	Hypothesis	Label
"Eine Person auf einem Pferd springt über ein zusammengebrochenes Flugzeug." - *"A person on a horse jumps over a broken down airplane."*	"Eine Person trainiert ihr Pferd für ein Turnier."- *"A person is training his horse for a competition."*	"no contradiction"
"Kinder lächeln und winken der Kamera zu." - *"Children smiling and waving at camera"*	"Die Kinder runzeln die Stirn." - *"The kids are frowning"*	"contradiction"

Table 4.1: Examples from the SNLI data set, machine-translated German version and English original (in italic)

4.4.2 Challenges for the NLI model

The model put out faulty analyses whenever it was confronted with grammatically incomplete and incorrect sentences. As soon as one of the sentences showed grammatical deficiencies in the form of sentence breaks (anacoluth) or word cuts, problems arose with the recognition of the reference word or the sentence's meaning. Another area of concern is the record length. The model often failed to recognize the syntactic and semantic keywords/signifiers when confronted with longer and more complicated sentences.

Premise: "Ich finde den Ansatz mit den Bioölen sehr gut. Deshalb habe ich mich auch für eine Bestellung entschieden." – *"I think the approach with the organic oils is very good. That is why i decided to place an order"*
Hypothesis: "Ich habe selten so viel Kulanz und Entgegenkommen von einem Händler erlebt. Ich habe am dritten Februar bestellt, leider kam das Paket nicht zum angezeigten Liefertermin. Ein kurzer E-Mail Kontakt mit dem Händler zeigte, dass das Paket beim Zusteller verloren gegangen war. Ohne Umstände wurde sofort ein neues Paket losgeschickt, was auch 3 Tage später ankam doch leider war dort eine Flasche kaputt. Ein erneuter E-Mail Kontakt und schon wurde die Flasche ersetzt, aber nicht nur das mir wurde auch noch eine Flasche, wegen den ganzen Umständen geschenkt." – *"I have rarely experienced so much goodwill and responsiveness from*

Premise	Hypothesis	Label
"Die Qualität der Kette ist sehr gut. Die Kette sieht hochwertig aus und die Lieferung war wirklich schnell :)" - *"The quality of the necklace is very good. The necklace looks high-quality and delivery was really fast :)"*	"Keine Benachrichtigung über Sendung und keine sendungsverfolgung möglich. Zu lange Lieferzeiten."- *"No notification on the shipment and shipment tracking not possible. Delivery times too long."*	"contradiction"
"Das Unternehmen schreibt sich das Thema hoch auf die Fahne. Leider steht es nur da.Angefangen von der Mülltrennung bis zum Versand von E-Teilen, die in überdimensionierten Kartons versendet werden." - *"The company claims to highly prioritize the topic. Unfortunately, that is all it does. Starting with waste separation, as well as mailing spare parts in oversized boxes."*	"Es werden gerne mal zum CSD etc. Marketingaktionen gestartet oder das T-Logo in Regenbogenfarben angemalt. Nachhaltig ist das aus meiner Sicht nicht" - *"It is common for the company to start marketing campaigns on the occasion of CSD etc., or paint the T-logo in rainbow colors. From my point of view, none of this is sustainable."*	"no contradiction"

Table 4.2: Examples from the internet data set, original German version and English translation (in italic), with labels

a retailer. I ordered on 3 February, but unfortunately the package did not arrive on the date indicated. A brief email contact with the retailer showed that the parcel had been lost by the delivery company. A new parcel was immediately sent, which arrived three days later, but unfortunately one of the bottles was broken. Another email contact and the bottle was replaced, but not only that, I was also given a bottle as a gi because of all the circumstances."

Gold label: No contradiction
Prediction: Contradiction

In this example, a grammatical construction consisting of two main clauses is juxtaposed with a multi-membered construction, in this case consisting of main and subordinate clauses. The syntactically and semantically important signifier of the premise “with the bio-oils” is suppressed in the hypothesis and not explicitly emphasised again. Even if there is a clear connection in terms of content, it is lost in the stringing together of individual semantic hierarchies. A clear semantic analysis is hardly possible for the model in this form due to this structure and the lack of reference words or similarities.

The following example also shows problems with the accumulation of sentences in juxtaposition to short or even elliptical sentences. The meaning and structure of the conditional construction is radically changed in the hypothesis. Furthermore, it is questionable whether the meaning of such short phrases as “fits exactly” can really be determined and related. The same problem naturally occurs with grammatically incomplete sentences. Missing reference words and inter-syntactic relations cannot be sufficiently captured, even though these sentence constructions are accepted in both German and English and may fall under the genre of linguistic devices.

Premise: “Ja okay, Basic heißt nicht hochwertig. Auf dieses Laken möchte man sich nun wirklich nicht legen. Das Laken greift sich sehr unangenehm ist eigentlich kaum zu beschreiben, ähnlich Plastik. Ich hatte dieses vorgesehen für meinen Mieter einer möblierten Wohnung, aber ich denke das möchte ich ihm nicht zumuten." – *“Yes okay, basic does not mean high quality. You really don’t want to lie on this sheet. The sheet is very unpleasant to the touch, it’s hard to describe, similar to plastic I had intended this for my tenant in a furnished flat, but I don’t think I want to put him through that."*
Hypothesis: “Passt genau!" – *“Fits perfectly!"*

Gold label: No contradiction
Prediction: Contradiction

In addition the model struggled with negations. If a negation was not directly related to the signifier, flawed results were produced. To this effect, there have been cases in which additional words could stand out and shift the meaning. Furthermore, the model did not deal well with technical or rare terms and was unable to compare them adequately. At the same time ambiguity, antonyms, homonyms and homonymous verbs were not recognized and could therefore not be linked correctly. The model seemed to recognize metaphors and allegories only to a limited extent. The previously mentioned uncertainty with individual terms, as well as the semantic capturing of individual parts of sentences, lead to most errors in terms of recognizing contradictions.

Premise: “Auf der Demonstration hatten die Demonstranten mit viel Rauch und Nebel zu kämpfen." – *“On the demonstration, the demonstrators had to deal with a lot of smoke and fog"*

Hypothesis: “Die Polizei setzte Tränengas gegen die Demonstranten ein." – *“The police used tear gas against the demonstrators."*

Gold label: No contradiction
Prediction: Contradiction

This example shows another interesting problem of the model. The premise receives the state description of an environmental occurrence. “Smoke and fog” stand together here as a methaporic synonym to the statement of "tear gas" contained in the hypothesis. The model cannot recognize the metaphor. The same applies to homonyms and homonymous verbs. Once the meanings are not presented in a direct way, the model cannot analyze possible contradictions due to lack of understanding. Furthermore, such word types can often only be evaluated from the context. Thus, this bivalent problem appears to be a great challenge for the model.

The greatest error rate, however, was seen in the analysis and assignment of local prepositions. These could only rarely or not at all be distinguished from one another and were treated in the same way by the model although they fulfil a major semantic function.

Premise: “Tim sitzt neben der Badewanne." – *“Tim sits next to the bathtub."*

Hypothesis: “Tim wäscht sich." – *“Tim washes himself."*

Gold label: Contradiction
Prediction: No contradiction

This example was not part of one of the data sets, but was created by us to test the ability of the

model to recognize semantic differences when the sentence is being slightly altered. We experimented with replacing the respective local preposition by other local prepositions (“in/auf/neben/unter”), which, however, give a completely different semantic implication. We wanted to test whether the model’s prediction would change, but it yielded the same result as the original sentence. Furthermore, we added a subordinate clause (“... während eine Frau sich wäscht" / “while a woman is washing herself") that directs the meaning to another object and yet the model stuck to an incorrect analysis and made no distinctions among the prepositions or the sentence content. It was interesting that these errors did not occur with other preposition types.

Based on these findings, we explore informed pre-training approaches that capture syntactic and semantic knowledge. Specifically, we investigate Part-of-Speech-Tagging and semantic webs, as we hypothesize that additional syntactic and semantic knowledge can help the model correctly identify those contradictory samples it currently struggles with.

4.5 Methodology: Pretraining Methods

As outlined above, we aim to inject syntactic and semantic information into the transformer model architecture by training with different pre-training objectives. All of those objectives are word-based, meaning that we utilize the output vector mapping to the corresponding input-token for these tasks instead of the special [CLS] token, which is commonly used for sentence level classification tasks. All of our labels are generated in a semi-supervised manner. We take advantage of already present and well working architectures to predict labels for POS-Tagging and dependency parsing, and create labels for different synsets with the NLTK wordnet[7] interface supporting the WordNet[8] [50] lexical database.

4.5.1 POS-Tagging

The main objective of Part of Speech (POS-) Tagging is to predict the syntactic function of a word in a sentence. Words can have different meanings in different contexts. Therefore, POS-Tagging is used, among other things, to identify the context in which a word occurs. The used tagset includes common parts of speech such as adjective, noun and verb, but also finer graduations such as numerical and symbol words. We extract labels from spaCys implementation for POS-Tagging [66]. Among the common POS-tags are: NOUN (noun), VERB (verb), ADJ (adjective), ADV (adverb), DET (determiner), PRON (pronoun). The full list can be found at the spaCy repository[9].

The following example shows semi-supervised generated tags for a tokenized sentence from the SNLI dataset. An underscore corresponds to the beginning of a word. As POS-tags are associated with complete words, but some words are being split into multiple tokens during tokenization, each input-token is assigned the POS-tag for the complete word. So, tokens for words that are split up by the tokenizer all map to the same POS-tag.

_A	_person	_on	_a	_horse	_jump	s	_over	_a	_broken	_down	_air	plan	e	.
DET	NOUN	ADP	DET	NOUN	VERB	VERB	ADP	DET	VERB	ADP	NOUN	NOUN	NOUN	PUNCT

[7] https://www.nltk.org/howto/wordnet.html
[8] https://wordnet.princeton.edu/
[9] https://github.com/explosion/spaCy/blob/master/spacy/glossary.py

4.5.2 Parent Prediction

For Parent Prediction (PP) [23] the parent of each word is predicted. The parent is deduced from a corresponding dependency tree of the sentence, which was created using the NLP library Stanza [77]. The dependency tree provides information about the syntactic dependency relation between words. Each word is assigned to exactly one other word, so each word has precisely one parent. The central clause, i.e. the root clause without parent, is a (finite) verb.

4.5.3 Synset Prediction

In order to enhance the model with semantic knowledge, we take advantage of the WordNet [50, 78] knowledge graph, which is the lexical database for the English language. The nouns, verbs, adjectives and adverbs in WordNet are organized in groups, based on their semantic similarity, called synsets (synonyms sets). One synset represents one distinct concept, thus one synset can contain several lexical units, where each of the lexical units represents one meaning of a word. Since words have several meanings, they can be associated with several synsets. For instance, the synset for the word *lady* in the sentence *"The lady is weeding her garden."* contains three possible meanings, as it can be seen below.

Synset('lady.n.01'), Synset('dame.n.02'), Synset('lady.n.03')

For more detail on synsets and semantic webs, please refer to section 2.4.2.

The main objective of this pre-training task is the prediction of labels representing semantic knowledge. We extract the synsets from WordNet for nouns, verbs and adjectives. The WordNet[10] NLTK corpus reader is used for the extraction of synsets. The first synset in a set of synsets represents the most common meaning of a word. Thus, we utilize the first synset for semantic representation of a word. For example, for the word *lady* the synset *Synset('lady.n.01')* is chosen. We argue that since most words have a unique meaning, this approach is a reasonable heuristic, even though it will introduce a small amount of noise by assigning the wrong synset to uncommon words. To our best knowledge, it is the first attempt to utilise the synsets for pre-training the model with semantic knowledge.

The following example shows the tokenized sentence from above and the corresponding labels. Similar to the example in 4.5.1 the label for a complete word is assigned to each of the subword tokens, just as in case with *_we ed ing*.

_The	_lady	_is	_we	ed	ing	_her	_garden
no_syn	lady_n_01	be_v_01	weed_v_01	weed_v_01	weed_v_01	no_syn	garden_n_01

4.6 Experiments and Results

In the next section, we describe the experimental setup and further evaluate our proposed pre-training objectives quantitatively and qualitatively. We do not use any additional data, other than the SNLI training set, and prolong the overall training only by a few epochs.

The main model[11] is based on a BERT architecture with approximately 110M parameters, 12-layers,

[10] https://www.nltk.org/_modules/nltk/corpus/reader/wordnet.html

[11] https://huggingface.co/bert-base-cased

12 attention heads and a hidden state of size 768. A simple feed-forward layer is used for classification and shared across each output vector or, in case of finetuning, for the special [CLS] token. The BERT model has been pre-trained for the Masked Language Modeling and Next Sentence Prediction tasks on a large corpus of English data from books [79] and Wikipedia. Further, we compare to a large XLM-RoBERTa[12] with 355M parameters. Binary Cross Entropy Loss in combination with AdamW optimizer [80] is used for all experiments. For pre-training a learning rate of 6e-5 is used. For fine-tuning we utilize a learning rate of 5e-6.

Evaluating our main model, the overall best results are achieved when we pre-train for POS-Tagging and Synset Prediction, yielding a significant performance boost over the baseline model (see table 4.3). This proofs that linguistically informed pre-training does in fact help the model to capture additional knowledge that is helpful for the classification task. Apparently, not all combinations of pre-training methods work equally well. For example, combining all three approaches yields slightly worse results than combining only POS-Tagging and Parent Prediction, or POS-Tagging and Synset Prediction. A possible explanation for this behavior is that the model "forgets" previously learned knowledge, if it is trained for multiple tasks in a row. It is yet to be explored, whether it would help the model if the objectives were applied subsequently to specific layers of the transformer.

Pretraining Configuration	**Acc.**	**F1-Score (Cont.)**	**F1-Score (Ent.)**	**F1-Score (Neut.)**
No additional pretraining	88.6	91.6	89.7	84.5
POS	90.0	92.4	90.9	86.7
PP	89.5	92.1	90.4	85.9
POS+PP	90.2	92.8	91.1	86.5
Syn	89.9	92.3	90.8	86.6
POS+Syn	**90.4**	**93.2**	**91.7**	**86.7**
PP+Syn	89.9	92.6	90.6	86.3
POS+PP+Syn	89.9	92.5	90.7	86.4

Table 4.3: Performance comparison for different pre-training configurations on the SNLI test set, in percent. The abbreviations stand for: POS=POS-Tagging, PP=Parent Prediction, Syn=Synset Prediction.

Configuration	**Base model**	**Num. param.**	**Acc.**	**F1 (Cont.)**	**F1 (Ent.)**	**F1 (Neut.)**
Current SOTA (EFL)	roberta-large	355 M	93.1	n.a.	n.a.	n.a
No add. pretraining	xlm-roberta-large	345 M	91.5	94.5	92.1	87.7
POS+Syn	xlm-roberta-large	345 M	91.5	94.5	92.0	88.1
No add. pretraining	bert-base-cased	110 M	88.6	91.6	89.7	84.5
POS+Syn	bert-base-cased	110 M	90.4	93.2	91.1	86.7

Table 4.4: Performance comparison for different model architectures on the SNLI test set, in percent. We compare our approaches with (POS+Syn) and without pre-training to the current best result on the data set by [81].

Comparing the different model architectures (table 4.4), it is apparent that adding further pre-training

[12] https://huggingface.co/xlm-roberta-large

tasks helps the smaller models achieve competitive results compared to xlm-roberta-large, while it does not yield a huge performance boost for the large model itself. In order to prove that the improvement is significant and due to pre-training tasks, we compare the mean performance for five training and evaluation runs of the model architectures with additional pre-training and without it in table 4.5. While the difference between xlm-roberta-large performances with and without additional pre-training is almost not noticeable, the mean of evaluation results of the smaller model with additional pre-training shows improvement. This suggests that enhancing the smaller models with additional knowledge could make them competitive, and thereby not having to rely on extensive computational resources. At the same time, both models achieve results that are comparable to the current state of the art [81, 82].

Configuration	**Base model**	**Mean Acc.**	**Mean F1 (Cont.)**	**Mean F1 (Ent.)**	**Mean F1 (Neut.)**
No add. pretraining	xlm-roberta-large	91.8(±0.10)	94.5(±0.08)	88.4(±0.10)	87.6(±0.15)
POS+Syn	xlm-roberta-large	91.8(±0.06)	94.5(±0.09)	88.6(±0.12)	87.8(±0.09)
No add. pretraining	bert-base-cased	89.5(±0.10)	91.8(±0.08)	86.0(±0.10)	84.3(±0.15)
POS+Syn	bert-base-cased	89.6(±0.06)	92.0(±0.09)	86.1(±0.12)	84.5(±0.09)

Table 4.5: Mean with standard deviation of different model architectures performance on the SNLI test set, in percent. Each of the models was evaluated five times and the mean was calculated over all five evaluation results per setting.

4.7 Conclusion and Summary

In this chapter, we first presented an in-depth error analysis of a transformer-based Contradiction Detection model from a linguistic perspective, based on two data sets in German language. In doing so, we discovered a number of syntactic and semantic features that pose a challenge to the model. Based on those findings, we developed a combination of linguistically informed pre-training methods for transformers. The experimental results illustrate that the performance of the transformer models on the NLI task can be improved by enhancing the models with syntactic and semantic knowledge. The novel method of Synset Prediction shows that enriching transformer models with semantic knowledge positively affects the ability of the models to learn semantic correlations in data. Moreover, it is not required to utilize a large model for handling the CD task, as smaller models perform competitively when trained with linguistically informed objectives. Another important advantage of our approach is that the improvement can be achieved with no additional training data. Part of this research has already successfully been applied in an industry context, for finding contradictions in financial reports [27], showing that an informed approach also facilitates domain adaptation (for more details, see section 8.3).

A significant limitation of our work is the rule-based annotation procedure for the Synset Prediction task, utilizing always the first (most probable) synset extracted from WordNet as a label for a given word. This is clearly not ideal, as it introduces some noise, and less common synsets are not represented in the labels. Nevertheless, the results show that we can already achieve a performance improvement by using this simplified approach. It would be an interesting direction of research, to treat this problem as a Machine Learning task on its own and train a dedicated model, which would most likely enhance

the downstream performance even further. This, of course, would require a certain amount of manual annotations. In this regard, it could also be meaningful to reduce the number of predicted synsets by grouping them together into clusters or hypernym groups, which would make the learning problem easier and less sparse.

In the upcoming chapter, we will provide an application of the presented method on a CD use case in a low-resource language.

CHAPTER 5

Applying Informed Language Model Training to Low-Resource Languages

In this chapter, we apply the idea of linguistically informed pre-training to a low-resource application scenario, namely Natural Language Inference in Arabic. Our research objective is to determine whether those methods can help to improve the results in this specific domain, thereby investigating RQ2:

Research Question 2 (RQ2)

How do smaller, language-specific models trained with linguistically informed objectives and/or data augmentation perform compared to larger language-agnostic models for low-resource scenarios?

In order to answer this question, we collect a data set in Arabic language from diverse sources and apply our informed methods on it.

The key contributions of this chapter are:

- We present a novel data collection for Natural Language Inference in Arabic.
- We evaluate different transformer based models on the Arabic NLI task, showcasing the effectiveness of informed pre-training using NER labels.

The research in this chapter was conceptualized and supervised by Maren Pielka. Majd Saad al Deen collected the data set and conducted the experiments as part of his Bachelor's thesis [83] under the supervision of Maren Pielka and Jörn Hees. The paper was written by Maren Pielka and Majd Saad al Deen to equal parts.

This chapter is based on the following publication [24]:

- Mohammad Majd Saad Al Deen, **Maren Pielka**, Jörn Hees, Bouthaina Soulef Abdou, and Rafet Sifa. 2023. "Improving Natural Language Inference in Arabic Using Transformer Models and Linguistically Informed Pre-Training." In 2023 IEEE Symposium Series on Computational Intelligence (SSCI), pages 318-322, Mexico City, Mexico. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/SSCI52147.2023.10371891

We start the chapter with an introduction on Arabic NLP and its specific intricacies related to the structure of the language. We further introduce our custom data set in section 5.3, and our experimental methodology in section 5.4. The results are presented and interpreted in section 5.5, followed by a conclusion and summary.

5.1 Introduction

Natural Language Processing (NLP) in Arabic, also known as Arabic NLP, is a subfield of Artificial Intelligence (AI) that focuses on processing and analysing textual data in the Arabic language. It encompasses various technologies and methods for automating tasks such as Text Classification, Sentiment Analysis, and Machine Translation. The goal is to teach computers to understand and process Arabic language, enabling a range of applications including chatbots, text mining tools, and translation services [84, 85]. This is a challenging field due to the limited availability of training data and pre-trained models for the Arabic language.

Natural Language Inference (NLI), also referred to as Recognizing Textual Entailment (RTE), is a subfield of Natural Language Processing (NLP) that aims to determine the semantic relationship between two pieces of text, known as the "premise" and the "hypothesis". The task, as described by MacCartney et al. [86], involves identifying possible connections such as "entailment" (if the premise is true, the hypothesis must also be true), "contradiction" (if the premise is true, the hypothesis cannot be true, and vice versa), or "neutral" (there is no logical relationship between the two sentences; both can be true or false simultaneously). Accomplishing this task requires a Machine Learning algorithm to comprehend the semantics of a text, which poses a particularly challenging problem.

In this chapter, we apply existing deep learning approaches, namely AraBERT and XLM-RoBERTa, to the task of Natural Language Inference and Contradiction Detection in Arabic. To our best knowledge, this has not been attempted before in such a comprehensive manner, as previous studies were limited to a smaller number of data sources and models. We also employ an informed language modeling approach, by further pre-training the transformer model on an NER task, before fine-tuning it on the downstream tasks. This is a novel direction of research with respect to Arabic text. In addition, we introduce a new data collection for NLI/CD in Arabic language, which is publicly available on Github[1].

5.2 Related Work

In the field of Natural Language Inference (NLI) and considering its impact on Question Answering (QA) tasks, Mishra et al. [87] conducted a research study. They utilized the RACE data set [88], which is a large-scale reading comprehension data set consisting of questions and answers from English exams for Chinese students. The authors converted a subset of RACE (containing 48,890 training examples, 2,496 validation examples, and 2,571 test examples) into an NLI format and compared the performance of a state-of-the-art model, RoBERTa [68], in both formats. To convert a reading comprehension question into a Natural Language Inference (NLI) format, the question was used as the premise, and each answer option was paraphrased as individual hypotheses. The same model architecture, comprising a RoBERTa encoder and a two-layer feed-forward network as the

[1] https://github.com/fraunhofer-iais/arabic_nlp/

classification head, was employed for both QA and NLI. The results showed that the NLI model outperformed the QA model on the subset of the RACE dataset. This can be attributed to the more natural form of the hypotheses in the NLI model compared to the combination of question and answer option in the QA model.

With respect to NLI in Arabic, Jallad et al. [89] conducted a similar study and created their own dataset called arNLI, which consists of over 6,000 data points. The data was obtained using Machine Translation from two English sources, namely SICK[2] and PHEME[3]. The authors developed a system with three main components: text preprocessing (cleaning, tokenization, and stemming), feature extraction (contradiction feature vector and language model vectors), and a Machine Learning classification model. The morphological units were processed using the Snowball Stemmer (Porter2) algorithm [90]. Various types of features were employed for feature extraction, including features for named entities, similarity, specific stopwords, number, date, and time, which were processed using different embeddig models such as TF-IDF [1] and Word2Vec [38]. To determine the relationship type between two sentences (contradiction, entailment, or neutral), various traditional Machine Learning classifiers were used, including Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), Decision Tree (DT), ADA Boost Classifier, K-Nearest Neighbors (KNN), and Random Forest. The proposed solution by the authors was trained and evaluated on their own dataset (arNLI), and they reported that Random Forest achieved the best results on the arNLI dataset with an accuracy of 75%.

Contradiction detection has received relatively limited attention in the literature compared to other tasks. De Marneffe et al. [9] define a contradiction as a conflict between two statements that mutually negate each other. In a stricter logical sense, there is no possible world in which both statements can be simultaneously true. A looser definition, aligning better with human intuition, suggests that a contradiction exists when it is highly improbable for two statements to be simultaneously true. To identify these contradictions, the authors employ an approach based on syntactic analysis and semantic understanding of the text. They utilize syntactic parsing tools to establish the logical structure of the text, and semantic features such as antonymy, polarity, and numerical deviation to comprehend the meaning of words and sentences in the text. The authors note that detecting contradictions may be more challenging than recognizing entailment and requires a deep semantic understanding, possibly augmented by world knowledge.

Pucknat et al. [67] conducted a study comparing the performance of four state-of-the-art models in NLI, particularly for contradiction detection, on German text data. These models were evaluated based on their performance on a machine-translated version of the well-known Stanford Natural Language Inference dataset (SNLI) [13] and the German test set of the Cross-Lingual NLI Corpus (XNLI) [91]. One key focus was to determine if the models were robust with respect to data selection and could potentially be applied in real-world scenarios. The XLM-RoBERTa model significantly outperformed the other models, likely due to its extensive pre-training and multi-head attention. However, the models did not generalize well to the XNLI data, indicating that the training corpus was limited in terms of topics and types of contradictions. The authors report an accuracy of 86.5% when testing XLM-RoBERTa on the XNLI dataset.

Another methodology by Pielka et al. [21], which was introduced in chapter 4, focused on pre-training methods to integrate syntactic and semantic information into state-of-the-art model architectures. The authors presented a linguistically enhanced approach for pre-training transformer

[2] https://alt.qcri.org/semeval2014/task1/

[3] https://www.kaggle.com/datasets/usharengaraju/pheme-dataset

models. They incorporated additional knowledge about part-of-speech tags, syntactic analysis, and semantic relationships between words into the model. Their goal was to become independent of massive pre-training data resources by integrating as much external knowledge as possible into the model. Their approach was evaluated on the SNLI dataset, and they demonstrated that the smaller BERT model can be competitive with XLM-RoBERTa when enhanced with additional knowledge during pre-training. Their approach did not require additional data for pre-training, as it was trained on additional tasks using the same dataset that would later be used for fine-tuning. In this work, we will build upon those results and extend the approach to non-Indoeuropean languages.

5.3 Data

For this study, a self-constructed corpus was used, comprising data from three different sources.

- XNLI (Cross-Lingual NLI Corpus) [91]: The Arabic-translated section of the XNLI dataset was included as a source for the corpus. XNLI is a well-known benchmark dataset for cross-lingual Natural Language Inference tasks, containing 7500 text pairs in 15 languages.

- SNLI (Stanford Natural Language Inference Corpus) [13]: The Arabic-translated section of the SNLI dataset by [92], comprising 1332 manually translated sentence pairs, was also incorporated into the corpus. SNLI is a widely used dataset in English language for NLI, consisting of sentence pairs labeled with three relationship types: entailment, contradiction, and neutral.

- arNLI (Arabic Natural Language Inference) [89]: The arNLI dataset, specifically created for the NLI task in Arabic language, was an additional source of data. This dataset consists of 6366 data points and was obtained through Machine Translation from two English sources.

Some examples from the data set are displayed in Table 5.1. As the data was compiled from different sources, necessary standard normalizations with respect to encoding, column names, label mappings etc. were performed. In the context of using transformer-based models for NLI/CD tasks, additional preprocessing steps such as stemming or stopword removal are not necessary (see section 2.3).

After performing the general preprocessing for the data from the three sources, the next step is to merge it and create a unified dataset. To ensure a fair distribution of training, testing, and validation data, the merged data is being randomly shuffled before splitting. The split is then done with a distribution of 60% for training data, 20% for testing data, and 20% for validation data.

The final data set consists of a total of 14,758 pairs of premises and hypotheses, each accompanied by an English translation and a label. The labels are encoded as 0 (entailment), 1 (neutral), and 2 (contradiction). Retaining the English translation of the data can assist in better comparing and understanding different texts and contexts, facilitating the comparison of various models.

For the NER pre-training, the ANERcorp corpus from CamelLabSplits [93] is being used, which contains 3973 text samples in Arabic language with annotations for the NER task. The entity types "person", "location", "organization" and "miscellaneous" are being used and annotated according to the IOB-scheme.

premise	hypothesis	label	premise_en	hypothesis_en
لقد دخل أول فريق للتدخل السريع التابع لشرطة نيويورك ردهة الشارع الغربي للبرج الشمالي واهم مستعدون لبدء التسلق حوالي ٩:١٥ صباحا	كان البرج لا يزال قائماً في الساعة ٩:١٥ صباحا	0	**The first NYPD ESU team entered the West Street-level lobby of the North Tower and prepared to begin climbing at about 9:15 A.M**	**The tower was still standing at 9:15 AM.**
يتطلب الأمرشراكة بين الدعم الخاص والتمويل الجامعي لمدرستنا القانونية لمواصلة النمو في المكانة والتأثي	مدرسة القانون لدينا مدعومة جزئياً بواسطة مؤسسة ميلندا وبيل جيت	1	**It takes a partnership of private support and University funding for our law school to continue to grow in stature and influence**	**Our law school is supported in part by the Melinda and Bill Gates Foundation**
يجب على الأمريكيين أيضًا أن يفكروا في كيفية القيام بذلك وتنظيم حكومتهم بطريقة مختلفة	يمكن تنظيم الحكومة فقط بطريقة واحدة وأي محاولة لتغييرها ستكون غبية	2	**Americans should also consider how to do it-organizing their government in a different way.**	**The government can only be organized in one way and any attempt to change it would be foolish**
البوابة التي تمثل جزءًا من جدار المدينة، لم يكن المقصود منها من البروسيين الأكثر براغماتية أكثر من مجرد قوس النصر كمعبر لفرض الرسوم	كانت البوابة مجرد قوس نصر.	2	**Forming part of the city wall, the gate was intended by the more pragmatic Prussians not so much as a triumphal arch as an imposing tollgate for collecting duties.**	**The gate was just a triumphal arch**

Table 5.1: Four examples showing the Arabic text data, English translation and label (0: entailment, 1: neutral, 2: contradiction).

5.4 Methodology

In the scope of this study, analyses are conducted for Natural Language Inference (NLI) and Contradiction Detection (CD). The data processing pipeline remains the same for both tasks since the input data consists of the "premise" and "hypothesis" columns in both cases. However, differences arise in terms of the labels used. The labels of the datasets for the CD task are modified to treat the problem as a binary classification. Originally, "0" was used for entailment, "1" for neutral, and "2" for contradiction. However, since only a binary outcome is required in case of CD, "0" and "1" are mapped to "0" for evaluation, meaning there is no contradiction, and "2" to "1", meaning there is a contradiction.

In this study, two state-of-the-art models, namely AraBERT [94] and XLM-RoBERTa [22], are being investigated. AraBERT is based on the BERT [41] paradigm and pre-trained on 24 GB of news corpora in Arabic language with the Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives. A special sub-word segmentation algorithm is used to account for the semantic granularity of the Arabic language. For this study, the base version of AraBERT with 136 M parameters is being used. XLM-RoBERTa is a multi-lingual language model, which has been

pre-trained for MLM on 2.5 TB of CommonCrawl data in 100 languages. It is shown to achieve state-of-the-art results on many NLP tasks across multiple languages. We use the base version with 279 M parameters. To this end, we conduct the following modeling and evaluation steps:

- Loading the pre-trained model: We utilize the checkpoints *xlm-roberta-base*[4] and *aubmindlab/bert-base-arabertv02*[5] for XLMRoBERTa and AraBERT, respectively.
- Additional pre-training: We employ Named Entity Recognition (NER) as an additional pre-training step before fine-tuning the model on the downstream task. Following the findings from [21], the idea is to provide the transformer model with semantic knowledge that would help in identifying contradictions and entailments. To this end, a word-level classification head is being attached to the encoder during continued pre-training and later discarded.
- Model fine-tuning: The pre-trained model is then fine-tuned on the downstream tasks (NLI and CD) using the created datasets.
- Hyperparameter optimization is performed during the finetuning process to enhance the model's performance. Techniques such as learning rate scheduling, dropout, and batch size adjustment are employed (see section 5.5 for details).
- Model evaluation: After model fine-tuning, the performance of the model is evaluated using metrics such as accuracy or F1 score.

The final preparation of input data involves merging the "premise" and "hypothesis" columns in the training, testing, and validation datasets using special tokens ([CLS] and [SEP]). The separator token [SEP] is recognized by the BERT tokenizer as a special token to separate different parts of the text. This allows the BERT model to process the meaning of each text part separately and then combine them in a single step. The classification token [CLS] in the BERT architecture signals the model to perform classification on the input text. It is placed at the beginning of the text and its embedding will be used by the model to make predictions about the text's class membership.

Overall, this study demonstrates the effective utilization of transformer-based models, showcasing their adaptability and performance in NLI and CD tasks in the Arabic language.

5.5 Experiments and Results

We mainly compare AraBERT and XLM-RoBERTa with respect to their performance on the NLI and CD task. An extensive hyperparameter search is being conducted, using the Optuna [95] tool with 150 optimization runs. We train the models for a maximum of 5 epochs, with the option of early stopping if the validation performance does not improve any further. The AdamW optimizer [80] is being applied. The other hyperparameters, including learning rate, weight decay and batch size, are chosen according to the best result from the respective Optuna run. The experimental results on the two tasks are displayed in tables 5.2 and 5.3.

All models achieve overall good results on both tasks. It is especially noteworthy that AraBERT performs competitively with XLM-RoBERTa, even though it was pre-trained on a considerably smaller

[4] https://huggingface.co/xlm-roberta-base

[5] https://huggingface.co/aubmindlab/bert-base-arabertv02

Model	Multitask Finetuning	Accuracy	F1-Score
AraBERT	x	75.3	75.4
XLM-R	x	78.7	78.8
AraBERT	✓	76.8	76.8
XLM-R	✓	**78.9**	**79.0**

Table 5.2: Results for the NLI task with AraBERT & XLM-RoBERTa, in %. Accuracy and macro average F1-score are being reported.

Model	Multitask Finetuning	Accuracy	F1-Score
AraBERT	x	87.4	82.3
XLM-R	x	86.9	81.0
AraBERT	✓	**88.1**	**84.5**
XLM-R	✓	86.8	81.1

Table 5.3: Results for the CD task with AraBERT & XLM-RoBERTa, in %. Accuracy and macro average F1-score are being reported.

amount of data. With respect to the CD task, the best AraBERT model with mutlitask finetuning even outperforms XLM-RoBERTa by two percentage points. This emphasizes the fact that language-specific finetuning can be more effective than extensive multi-lingual pre-training for some downstream tasks. We also find that adding the NER objective as an additional pre-training step improves the performance. Interestingly, this effect is stronger for the smaller AraBERT model, suggesting that it can help bridge the performance gap that is caused by XLMRoBERTa's larger model size and the amount of training data it has access to.

5.6 Conclusion and Summary

We presented the first comprehensive study on Natural Language Inference and Contradiction Detection in Arabic language, in which we applied state-of-the-art transformer methods combined with an informed pre-training approach. The methods achieve promising results on our custom data set, emphasizing the fact that smaller, language-specific models like AraBERT can perform competitively with larger multi-lingual models, if they are being enhanced with additional linguistic knowledge. Further, we collected a large data set for NLI in Arabic language.

Future work includes adding more pre-training methods such as Part of Speech tagging, Word Sense Disambiguation or Semantic Role Labeling. We expect the performance of the AraBERT model to improve even further by adding more linguistic knowledge. Another direction of research is to exploit the potential of large language models such as GPT-4 [6], by casting the classification problem as a text generation task. It would be interesting to see the performance of those resourceful models when confronted with a low-resource language such as Arabic.

This chapter concludes the first part of the thesis, which was focused on injecting linguistic knowledge to transformer models during training. In the upcoming chapters, we will be introducing data augmentation based approaches with a similar objective, namely to reduce the resource footprint of language models and enhance their performance in low-resource scenarios.

CHAPTER 6

Machine Translation for Dataset Construction

The second main contribution of this thesis is the automated construction of data resources for LM training, using existing AI models and rule-based methods. In this chapter, we investigate a Machine Translation based approach for transferring an existing training corpus to another language, with respect to its effect on model performance. Again, we are interested in answering RQ2:

Research Question 2 (RQ2)

How do smaller, language-specific models trained with linguistically informed objectives and/or data augmentation perform compared to larger language-agnostic models for low-resource scenarios?

Specifically, we make use of a powerful neural network based translation engine to transfer a significant portion of the SNLI data set from English to German.

The key contributions of this chapter are:

- We introduce a novel data resource for Natural Language Inference and Contradiction Detection in German.
- We present a translation-driven approach for data generation, and evaluate its effectiveness on the Natural Language Inference and Contradiction Detection tasks. Our results indicate that Machine Translation is in fact a valid method to produce data for low-resource languages, given the comparable results on German and English test data.
- We conduct a comprehensive study on different featurization methods and model architectures for NLI and CD in German, finding that RNNs are able to surpass transformer-based methods when being trained and evaluated on machine-translated data.

This research was developed and implemented by Maren Pielka, while other authors had a supervisory role in it. It was in large parts included in her Master's thesis [96], while the final evaluation with transformer models has been conducted later. The corresponding papers were mostly written by her, with small contributions and refinements by the co-authors.

This chapter is based on the following publications [18, 19]:

- Rafet Sifa, **Maren Pielka**, Rajkumar Ramamurthy, Anna Ladi, Lars Hillebrand, and Christian Bauckhage. 2019. "Towards Contradiction Detection in German: a Translation-Driven Approach." In 2019 IEEE Symposium Series on Computational Intelligence (SSCI), pages 2497-2505, Xiamen, China. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/SSCI44817.2019.9003090
- **Maren Pielka**, Rafet Sifa, Lars Hillebrand, David Biesner, Rajkumar Ramamurthy, and Anna Ladi. 2020. "Tackling Contradiction Detection in German Using Machine Translation and End-to-End Recurrent Neural Networks." In 2020 25th International Conference on Pattern Recognition (ICPR), pages 6696-6701, Milan, Italy. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/ICPR48806.2021.9413257

The chapter is structured as follows: First, we will provide an overview on existing research with respect to NLI in German and other low-resource languages. In section 6.2, we introduce the translation approach and an assessment on the translation quality. Section 6.3 describes the experimental setup and methodology, including the choice of embedding methods and model architectures, as well as pre-processing routines. We present both qualitative and quantitative results in section 6.4, followed by a conclusion and summary.

6.1 Introduction and Overview

As outlined in section 2.5.1, Contradiction Detection belongs to the research field of Natural Language Inference (NLI), whose main objective is to learn semantic correlations between two text passages, commonly called *premise* and *hypothesis*. While conventional NLI is a three-class (*entailment*, *contradiction* and *neutral*) text classificaion task, for Contradiction Detection there are only two classes (*contradiction* and *no contradiction*). Past research on NLI has mainly been investigating the three-way classification task. Also, most of the work has been conducted with respect to English text. In our work, we focus specifically on Contradiction Detection with respect to German text; one reason for this being that there is not much prior research on this topic in particular. Furthermore, it is a valid approach to treat Contradiction Detection as a problem on its own. Contradictions are considerably harder to learn and to represent by a neural language model then entailments are, as they can appear in many different forms (numeric mismatch, different polarity, antonymity ...). Also, from a linguistic point of view, contradictions arise from distinct syntactic and semantical constructions. There is reason to expect that a model being trained exclusively for the Contradiction Detection problem, would yield superior results on that task, compared to a model that has been trained on the three-way problem.

There are a variety of applications for NLI and Contradiction Detection, e.g. in solving other text mining tasks such as Question Answering, Relation Extraction and Machine Translation. It can also be a valuable tool in empirical use cases, for example detecting contradicting statements in scientific papers [97], social media, or financial documents [72].

Conneau et al. [98] attempt to learn sentence embeddings using NLI data in a supervised way. Their goal is to obtain universal representations that would generalize to other NLP tasks. They use a bi-directional LSTM to learn embeddings on a paragraph level. The features of premise and hypothesis are combined and fed to the last layer to produce a prediction output. All model weights are trained end-to-end. Their main result is that such embeddings obtained using NLI data, are in fact

Training set size	100,000 sentence pairs
Test set size	10,000 sentence pairs
Label distribution ("entailment" / "neutral" / "contradiction")	0.34 / 0.33 / 0.33
Average premise length (original data)	12.95 words
Average hypothesis length (original data)	7.42 words
Average premise length (translated data)	12.26 words
Average hypothesis length (translated data)	7.01 words

Table 6.1: Some basic statistics of our translated SNLI dataset. In this work, we created a machine translated dataset, by automatically translating sentences from English to German for the Contradiction Detection task, using the DeepL API.

capable of generalizing to other tasks and outperform unsupervised representations such as FastText or SkipThought on most benchmark problems.
A different approach is presented by Rocktaeschel et al. [99]. In their framework, premise and hypothesis are processed consecutively by two LSTMs, where the output of the first network serves as an additional input for the second. The model implements an attention mechanism as its central component, which is trained to learn weights over all pairs of words from both sentences. Those weights should indicate which words are to be aligned and assigned a higher weight with respect to the prediction output. Later work, e.g. by [100, 101] is in large parts based upon these findings, yielding very good results e.g. on the SNLI data set.
Given that, we note that all the presented approaches so far have been restricted to the English language and there exist no large data sets as well as NLI evaluation results in other languages. The main goal of this chapter is to assess whether state-of-the-art RNN methods are equally well-suited for the NLI task and specifically Contradiction Detection, when dealing with machine-translated German data. For this purpose, we re-implement the frameworks introduced by [98] and [99], respectively, and evaluate their performance on both the original and the translated data set. We also apply a pre-trained, multi-lingual transformer-based (MBERT) model, and fine-tune it on the original and translated data set, respectively. In addition, we introduce a novel method for data generation based on machine translation, and validate this method using some simple features, that can serve as a baseline for future research in the topic.

6.2 A Machine Translated Dataset for Evaluating CD Models in German

Our evaluation is based on the Stanford Natural Language Inference (SNLI) dataset, collected by Bowman et al. [13], which is one of the largest data collections for the NLI task, containing 570,000 sentence pairs in the English language that were gathered via a crowd-sourcing campaign. The dataset was collected by showing the respondees an image caption (not the image itself) and asking them to come up with another description that is entailed by the caption, one which is neither entailed nor a contradiction, and one which clearly contradicts the caption. After the first data collection step, a group of reviewers had to make a final decision about the labels. Only if three or more people agreed on a choice, a label was chosen to be a “gold label”. This policy resulted in some part of the data not

English original	DeepL machine translation	Human reference translation
“A man gliding in the sky with the sunset on the horizon in the background."	“Ein Mann, der am Himmel gleitet, mit dem Sonnenuntergang am Horizont im Hintergrund."	“Ein Mann macht einen Segelflug, mit dem Sonnenuntergang am Horizont im Hintergrund."
“Three men in white shirts dance and sing on a stage as a group of women watch from the sidelines."	“Drei Männer in weißen Hemden tanzen und singen auf einer Bühne, während eine Gruppe von Frauen von der Seitenlinie aus zusieht."	“Drei Männer in weißen Shirts tanzen und singen auf einer Bühne, während eine Gruppe Frauen von der Seite zuschaut."
“a man walking up a mountain with snow on the surface"	“ein Mann, der einen Berg hinaufgeht, mit Schnee an der Oberfläche."	“Ein Mann geht einen mit Schnee bedeckten Berg hinauf."
“A woman with a light blue t-shirt and a backpack holding some vegetables in her right hand."	“Eine Frau mit einem hellblauen T-Shirt und einem Rucksack, der etwas Gemüse in der rechten Hand hält."	“Eine Frau mit einem hellblauen T-Shirt und einem Rucksack hält etwas Gemüse in ihrer rechten Hand."

Table 6.2: Examples of sentences from the SNLI data set, machine translated by DeepL, in comparison to a human translated reference (done by a linguist with German as their mother tongue). The mismatches between the machine and human translated version are mainly due to a slightly different phrasing. However, the resulting translations preserve the overall meaning well.

being uniquely labeled. It is worth noting that, overall, there was a high level of agreement in this labeling process, resulting in more or less equal class distribution in the final dataset. Since there is no NLI corpus of comparable size available for German language, we considered automatically translating a large part of the SNLI dataset into German via the DeepL API[1], which according to the developers[2] utilizes Neural Machine Translation. This is a necessary requirement as we need enough data for a large scale assessment and training neural networks with massive numbers of parameters. Due to computational limitations, and because we found it a reasonably good representation of the whole data set, we only considered a randomly sampled subset of 100,000 sentence pairs from the training set in our experiments and kept the test data at its original size of 10,000 sentence pairs, yielding a training/test ratio of 10:1. Table 6.1 summarizes the basic statistics of our translated dataset.

For an initial verification of the translation quality, we considered the Bilingual Evaluation Understudy (BLEU) score [102] on a random sample of 100 training and 10 test sentences. The BLEU score is the standard metric to quantify the performance of a machine translation. It is defined to measure the overlap between a machine translated text and one or more of its human reference

[1] https://www.deepl.com/translator

[2] https://www.deepl.com/press.html

translations. That is, the score is based on precision and represents the reference and candidate translations (in the following referred to as *ref* and *can* respectively) as sets of n-grams:

$$\text{BLEU}(ref, can) = \frac{|ref \cap can|}{|can|}, \tag{6.1}$$

where the individual n-gram count is clipped to the maximum number of times each n-gram appears in any reference translation. This is made to ensure that the machine translation does not contain a high number of similar words (e.g. "the the the the cat"). We consider word-level n-grams with $n = 4$ for the evaluation. For each sentence, one manual translation (done by a linguist with German as their mother tongue) is used as reference. The automated translation achieves an average score of 0.6664 on the sample, which is very high given the fact that scores of 0.40 - 0.50 are already considered good (even though a perfect score would be 1, but is hardly ever achieved). For example, Bahdanau et al [103] report BLEU scores of up to 0.36 with their encoder-decoder Machine Translation framework using an attention mechanism. Later work, e.g. the multi-language framework by Johnson et al. [104], achieves scores of up to 0.45 on benchmark data sets. Given that, a BLEU score of 0.67 is extremely high for a machine translated data set. For further details on the BLEU evaluation metric, see Papineni et al. [102].

We provide a list of machine translated examples from the SNLI dataset with a reference translation in Table 6.2. It is clear from those examples that the translation engine produces mostly correct and coherent translations. The mismatches between the machine and human translated version are mainly due to a slightly different phrasing. Although the overall meaning of the translated versions is not distorted, in some cases, the machine translation does not quite capture the full extent of the meaning (e.g. the third example sentence in Table 6.2), or gets the subject of the sentence wrong (e.g. the fourth sentence in Table 6.2), which is apparently due to a lack of world knowledge. Also, there are some examples where incorrect pronouns or verb tenses are used. This can be corrected with the right choice of pre-processing (see section 6.3.2). Overall, the original content is preserved reasonably well in almost all cases we analysed. Therefore, we can conclude also from a qualitative perspective, that the translation quality is sufficient to fulfill our needs.

6.3 Experimental setup

In this section, we will describe our methodology and experimental setup for quantitatively investigating whether translating the SNLI dataset influences the performance of predicting contradictions. To this end, we will first give a brief description of our embedding approaches, and then present the implementation and training details of our models. Second, we list the pre-processing steps that we considered in our approach. Finally, we present the implementation details for our models.

6.3.1 Learning Sentence Embeddings

Learning paragraph embeddings that capture a sentence's semantics is a key task in NLI. There exist a variety of different techniques that can be used to tackle that problem, ranging from simple Bag-of-Words approaches to more sophisticated approaches using Recurrent Neural Networks (RNNs). In this work, we consider simple feature-based methods as baselines and also propose a novel method to learn paragraph embeddings using RNNs and sequence-to-sequence learning. For those baselines,

the feature extraction is done on a sentence level, for every premise and hypothesis separately. As a last step, the vectors are concatenated, such that we get a shared representation for the sentence pair, which serves as input to a classifier. We further employ methods based on the frameworks introduced by [98] and [99], in which the embeddings are learned end-to-end, with some implementation aspects being slightly simplified. A transformer-based MBERT [105] model is employed for comparison.

Utilizing State-of-the-art Embeddings

As a first baseline method, we apply a simple Bag-of-Words model by considering the Term Frequency-Inverse Document Frequency (TF-IDF) representations [106, 107] for each paragraph. In addition, we also applied three well-known unsupervised feature extraction techniques, based on semantic similarity (Doc2Vec [108]), word co-occurrence (GloVe [2]), and contextual representations (Flair [40]). In case of GloVe and Flair, we extracted textual representations from the word embeddings by considering mean-pooling (see section 6.3.3 for details on the implementation).

Using an RNN Encoder in a Sequence-to-Sequence Setup

As another approach on learning paragraph embeddings that are suitable for the NLI task, we consider a Sequence-to-Sequence bi-directional (as in the cases for [109–111]) RNN model. The network architecture consists of an encoder and a decoder, where the encoder reads the input sentence as a sequence of tokens, and the decoder tries to reconstruct the sequence from the output of the encoder. This paradigm was first introduced by Cho et al. [109]. The model can be trained for different kinds of setups, e.g. as an auto-encoder, meaning the decoder shall reconstruct the original sentence, or for a translation task, training the decoder to output a translated version of the input sentence. In our case, the idea is to use the original and translated sentences as sources and targets. Since we have the data set in both languages available, we train models for auto-encoding and translation. We can then use the last hidden state of the trained encoder as a sentence embedding for the Contradiction Detection task. For this setup, we consider the sentences individually, and do not differ between premise and hypothesis. Only in the inference step, the two embeddings are concatenated as described above.

The encoder consists of an embedding layer and a Gated Recurrent Unit (GRU) [112]. Each input word is represented as a one-hot vector of dictionary size, and mapped to its corresponding embedding, which is initialized either randomly or with a pre-trained word representation. There is the option to train the word embeddings alongside the other weights of the model, or keep them fixed. The hidden state of the GRU is updated with each token being processed, and fed back into itself as additional input. The encoder's GRU is trained in a bi-directional fashion, meaning the input sentence is read in original and reverse order. With this option being enabled, twice the number of hidden states are kept compared to the uni-directional variant. After having read the whole sentence, the last hidden states are passed to the decoder.

The decoder will read the encoder's last hidden states, and produce another sequence of words in the target language, which does not have to be of the same length as the original sequence. Its first input is always the start of sequence ⟨SOS⟩ token. The input vector is given to a Rectified Linear Unit (ReLU) and then to a GRU, which produces the next output vector, based on the input and the last hidden state. The output vector is mapped to a probability distribution over all tokens in the target language via a softmax layer, and the token with the highest probability is predicted as the next word in the sentence. In the subsequent steps, the most recent output token is fed back into the decoder as

the next input. Once the end of sentence ⟨EOS⟩ token is predicted, this process of output generation is stopped. During training, back-propagation through time is applied to adjust the weights of both the encoder and decoder.

Further, there are some configurable settings which can also be specified to the training routine. First, there is an option to not feed the decoder's GRU the previous output in each prediction step, but the corresponding target output vector. This approach is called "teacher forcing" which lets the decoder not steer too far away from the correct output sequence. When using teacher forcing, training will most likely converge faster, but the model might produce incoherent results when presented with unseen sequences. It is therefore reasonable to only apply teacher forcing with a certain probability for each output word. This probability can be given as another hyper-parameter setting. Our model also implements an early stopping criterion, which terminates the training once the change in loss between two epochs falls below a certain threshold. In addition, the dropout rate and the learning rate also need to be specified as hyper-parameters.

The intuition behind this setup is that the encoder's hidden states should preserve some kind of semantic meaning, which is expected to be helpful in identifying contradictions. Especially when trained for the translation task, the encoder needs to understand the sentence to a certain degree, in order to encode embeddings that capture enough information to re-construct it in another language. This way of generating paragraph embeddings is not only useful in the NLI task, but can also be applied to other NLP tasks involving sentence classification as studied in [98].

RNN Encoder Trained End-to-End without Attention

As a first approach on learning sentence embeddings for the Contradiction Detection task in a supervised way, we consider a variant of the model introduced by [98]. The architecture consists of a bi-directional encoder RNN and a feedforward MLP with one hidden layer and a softmax output layer for classification. We use the max-pooled last hidden states of the encoder as a paragraph embedding. The high-level architecture of the model is displayed in figure 6.1.

The main component of the model is a Gated Recurrent Unit (GRU). It is composed of a reset and an update gate, which are trained to decide how much information to keep from the current input and previous hidden state, respectively. Formally, the update of the hidden state can be described as:

$$
\begin{aligned}
z_t &= \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \\
r_t &= \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \phi_h(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)
\end{aligned}
$$

where h_t is the hidden state at time step t, x_t is the input vector, z_t and r_t are the vectors calculated by the update and the reset gate, respectively, and W, U and b are parameters of the model. σ_g is a sigmoid function, ϕ_h is the hyperbolic tangent function, and $\odot$ denotes the Hadamard product [113]. We obtain paragraph embeddings for premise and hypothesis separately, by inputting them to the encoder RNN subsequently, and re-setting the GRU's hidden states between reading the two sentences. The two embeddings are concatenated and fed to the MLP, which outputs a prediction. Backpropagation is applied to adjust the parameters of both models jointly during training.

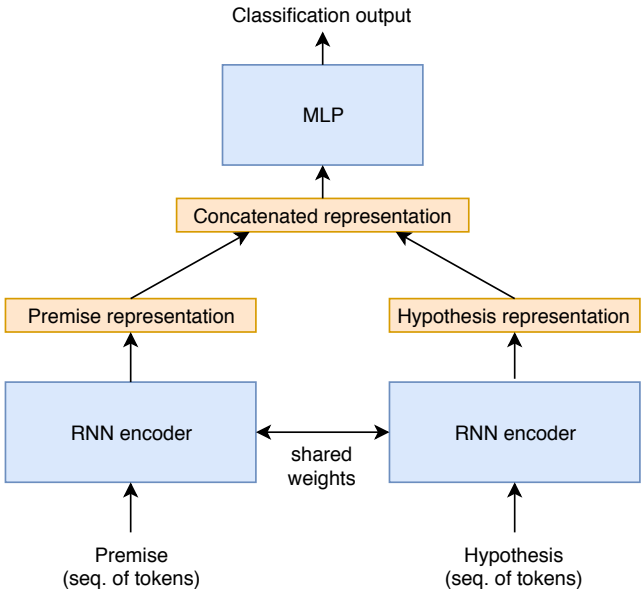


Figure 6.1: High-level architecture of the end-to-end RNN encoder model without attention. The model uses a bi-directional encoder RNN and a feedforward MLP for classification. The embeddings for premise and hypothesis are concatenated and fed to the MLP for prediction.

RNN Encoder Trained End-to-End with Attention

Another approach towards supervised embedding learning is a network architecture consisting of two bi-directional encoder RNNs, where the second one implements a word-to-word attention mechanism, and a feedforward MLP like in the previous setting. The architecture is visualized in figure 6.2. The first RNN (premise encoder) reads the premise representation as a sequence, and passes its outputs and last hidden state to the second RNN (hypothesis encoder). The hypothesis encoder reads the hypothesis, and learns attention weights over all word pairs from both sentences. The output of the hypothesis encoder is used as a joint embedding for both premise and hypothesis. This paradigm is based on the research of [99], who first proposed to use attention in the context of NLI. The attention mechanism was introduced by Bahdanau et al. [103], achieving good results on a Machine Translation task, compared to the state of the art at that time.

The architecture of the premise encoder is similar to the one used in the other paradigm. The hypothesis encoder takes three inputs: The premise encoder's last hidden state, its outputs - one for every word in the premise -, and the hypothesis as a sequence of embedded tokens. It implements a GRU layer, similar to the one used in the premise encoder, followed by a number of linear transformations and a softmax normalization:

$$\alpha_t = softmax((e_t + h_t)W_a^T)$$
$$att_t = Y\alpha_t^T$$
$$out_t = (att_t + h_t)W_o^T$$

Here, W_a are the attention weights, which map the current hypothesis token's embedding e_t,

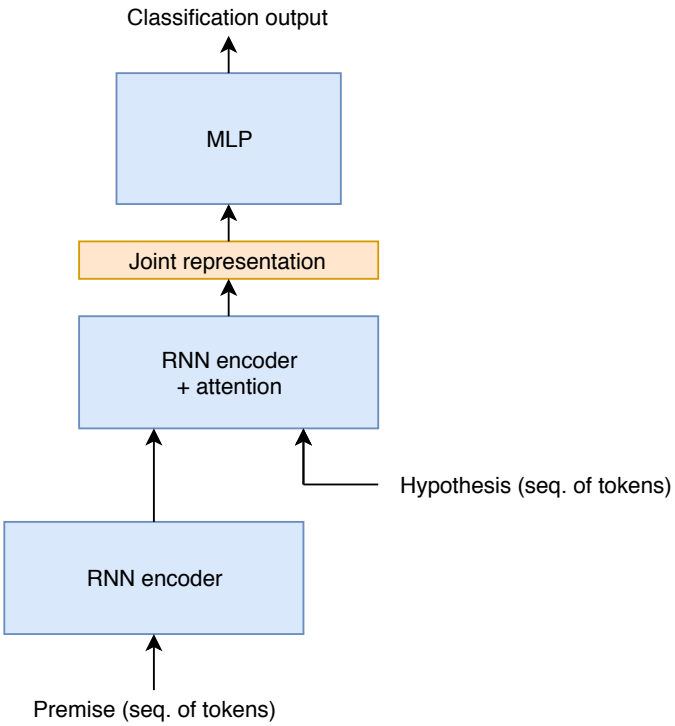


Figure 6.2: High-level architecture of the end-to-end RNN encoder model with attention.

concatenated with the max-pooled hidden states h_t at time step t, to an output vector of the maximum sentence length. The values α_t are applied to the outputs Y of the premise encoder, such that we get a weighted representation att_t for the premise. The hidden state of the hypothesis encoder is concatenated with this representation at the last time step, and mapped to the desired embedding dimensionality by a final linear transformation.

The setting allows for some introspection of the model, since we can analyze the attention weights between premise and hypothesis. Ideally, those word pairs should have a high attention weight, that are strong indicators for the presence - or absence - of a contradiction (see section 6.4.2).

6.3.2 Data Pre-processing

A number of data pre-processing steps are applied prior to training. Initially, we remove all training and test examples without a gold label, yielding a remaining training set of 99,858 and test set of 9,824 examples. The amount of text pre-processing that is meaningful for the prediction task, depends on the feature extraction method used. We consider the standard basic text pre-processing steps consisting of

- removing rare words that appear in less than 10 documents in the training corpus
- removing digits and punctuation (e.g. “3", “1000", “?", “.", ...)
- removing stop words, which in our case are the words with very high frequency and no distinct meaning (e.g. “though", “thus", “that", ...)

- lemmatizing verbs by transforming them to their corresponding infinitive form (e.g. “went" → “go", “laughing" → “laugh")
- changing the letters to lowercase.

For our baseline of TF-IDF vectorization, we employ all of the pre-processing steps listed above. In our experiments, we find that in case of English, it is sufficient to remove rare words, digits and punctuation. Applying stop word removal, lemmatization and lowercasing did not improve the prediction performance any further. However, for German text, lemmatizing and stop word removal yield some improvements while lowercasing was not effective for both languages. This can be explained by the fact that uppercase letters carry valuable information, they are e.g. indicators that a word is a name, or - in case of German - a noun.

For Doc2Vec, no lemmatization was used even when inferring the German word embeddings. This is because we do not want to lose semantic information that is contained in the conjugated verbs. Also, we did not remove stop or rare words because this would result in leaving gaps of meaning within a sentence, which is crucial for the Doc2Vec language model.

When using pre-trained GloVe, the pre-processing steps have to match those that were used for creating the GloVe model. So, since the words in the GloVe dictionary are all in lowercase, we also applied lowercasing here. Stop word and rare word removal were not used for a similar reason as in Doc2Vec: Removing words would tamper with the co-occurrence matrix and possibly yield sub-optimal embeddings.

In case of Flair, we only performed a minimal pre-processing, which is inserting white spaces to treat certain punctuation (“!", “.", “," and “?") as individual tokens. Other than that, the original text was left unchanged. The reason is that the authors of the paper [40] report top performance without any of the above measures.

For the Sequence-To-Sequence RNN, we used the optimal pre-processing steps with respect to the pre-trained word embeddings. In our experiments described below, we use pre-trained GloVe word embeddings, and thus the same pre-processing steps were applied as when using pooled GloVe (lowercasing, digit removal and punctuation removal).

In case of the end-to-end models, we only perform minimal preprocessing, which is inserting whitespaces to treat punctuation as individual tokens, and lowercasing the text. We argue that the sequential models rely heavily on context, and removing any words or suffixes would potentially result in loss of such.

6.3.3 Implementation Details

In case of Doc2Vec, we obtain two models per language; one trained on the SNLI corpus in German or English language, respectively, the other trained on a corpus of 2 million documents from Wikipedia. We train one Distributed Bag of Words (DBOW) and one Distributed Memory (DM) model per corpus with 400 dimensions each, and concatenate the vectors obtained from both, such that the resulting dimensionality of the paragraph embeddings is 800. This setting was proposed by Le et al. [108], and is widely used in literature (see example use cases in [114, 115]). For further details on the models, we refer to [108].

For GloVe, we also use models trained on larger corpora, as well as models trained only on SNLI.

There are pre-trained models available for both German[3] and English[4]. For our task, we chose models that were trained on 6 billion tokens from Wikipedia and (in case of the English model) the Gigaword 5[5] corpus. In both cases, the dimensionality of the word vectors is 300. We obtain paragraph embeddings from those by a weighted element-wise average, thus leaving the dimensionality unchanged. The weights for the word vectors are their TF-IDF values, such that more important words w.r.t. the paragraph get a higher weight.

For Flair, we also use both pre-trained and custom models. In case of the pre-trained models, the standard German and English Flair embeddings provided by the authors [40] are used, which were trained on around 1 billion tokens. We concatenate the forward and backward embeddings, as recommended in the paper, such that we obtain word embeddings of 4,096 dimensions. Paragraph embeddings are inferred from those by mean-pooling. With respect to our corpus, we also train custom models on SNLI only. For those, we use the recommended parameter settings (see [40]), but limit the dimensionality of the model to 1,024 for performance reasons. The same policy of concatenating forward and backward embeddings and mean-pooling is applied, such that we obtain 2,048-dimensional vectors.

For the sequence-to-sequence-based RNN models, we initialize the word embeddings of the encoder with pre-trained GloVe embeddings. The word embeddings are trained alongside the other weights of the model. Once training is finished, the last hidden state of the encoder is used as a sentence embedding. The dimensionality of the sentence embeddings is set to 300, to match the size of the word embeddings. The forward and backward hidden states of the bi-directional model are combined via max-pooling, such that the dimensionality is the same as when using a unidirectional model. We train the models with a maximum of 75,000 epochs. The hyper-parameters are tuned using a grid search on a reduced version of the training set prior to training, resulting in the optimal setting of the learning rate to 0.05, the teacher forcing ratio to 0.8, and the early stopping threshold to 0.1.

We do not use any attention mechanism in the translation or auto-encoder setup, because we desire that the whole information of the input sentence shall be encoded in the last hidden state. When using attention, the decoder reads the encoder's outputs from all previous steps to learn attention weights, and not only the last one. This would influence the training objective in disfavor of our objective, which is to learn meaningful sentence embeddings. We stick to a maximum sentence length of 25 words, even though there is no technical requirement for this without the attention mechanism. The reason is that the model cannot handle long sentences very well, and will produce sub-optimal embeddings. During inference, we should also presume our sentences to be of reasonably short length. Applying this measure, the training set is reduced to 95,922 sentence pairs, and the test set to 9,235 sentence pairs.

For the end-to-end RNN based models, pre-trained GloVe vectors are used as word embeddings, and the dimensionality of the sentence embeddings is set to 300. The word embedding weights are kept fixed throughout training, in order to not maintain an unnecessary high number of trainable parameters. For the attention model, the word embeddings are shared across both RNNs. The last hidden states are combined via max-pooling, to obtain a sentence representation. We initially set the learning rate to 0.001, and train for a maximum of 200 epochs. As recommended by [98], a scheduling policy is applied to reduce the learning rate by a factor of 0.1, if the validation loss does not decrease

[3] https://deepset.ai/german-word-embeddings
[4] https://nlp.stanford.edu/projects/glove/
[5] https://catalog.ldc.upenn.edu/LDC2011T07

after 10 epochs. Once the learning rate drops below 10^{-7} following this policy, training is stopped. We always use the model for evaluation, which achieved the best accuracy on the validation data (over all epochs). The feedforward MLP implements one hidden layer with 100 neurons, and a softmax output layer. A dropout probability of 0.2 is applied in the attention layers of the hypothesis encoder, and in the linear layer of the feedforward MLP. In addition, a weight decay of 0.0001 is applied for regularization. We perform batch learning with a batch size of 64. Stochastic gradient descent is used as an optimization function for all models in the pipeline.

For the MBERT-based model, we use the implementation by Lample et al. [105]. The authors provide a transformer model, that has been pre-trained on 15 languages with the objectives of Masked Language Modeling (MLM) and Translation Language Modeling (TLM), in combination with a one-layer feed-forward MLP for classification. We fine-tune the model on the downsampled English SNLI training set, and the translated German training set, respectively, using the provided script and the recommended parameter settings by [91].[4] For implementation details, we refer to [105] and [91].

6.4 Experiments and Results

In this section, the experimental results are discussed, both from a quantitative and a qualitative point of view. For the first angle, we consider common performance metrics such as accuracy and f1-score, while for the latter, we analyze the results in more detail and try to introspect our models. We focus on the two-way classification problem ("contradiction" vs. "neutral"). So in this case, the examples with the label "entailment" are treated as belonging to the "neutral" class, yielding a slightly imbalanced label distribution (32.95% "contradiction", 67.05% "neutral"). When analyzing the results, it is important to note that in case of the 3-way classification task, an accuracy of 0.33, and in case of the two-way task, an accuracy of 0.5 would indicate a random prediction performance.

6.4.1 Quantitative Evaluation

A summary of our results on both the original and the translated data are shown in Table 6.3. To represent the sentences, we consider TF-IDF, GloVe, Doc2Vec, Flair and RNN sentence embedders. In addition, we individually evaluate our predictors for GloVe, Doc2Vec and Flair embeddings that are pretrained (PTRN) on Wikipedia datasets, and embeddings that are trained on the original and translated SNLI datasets. The Sequence-To-Sequence RNN is implemented as described in section 6.3.1, evaluating both the translation and the auto-encoder setting, and initializing the encoder's word embedding layer with pre-trained GloVe word embeddings. As for the predictors, we evaluate the prediction performance of Logistic Regression (LR) and feed forward neural networks (FFNNs) with one, two and three hidden layers (respectively HL-{1,2,3}).

The end-to-end RNN-based models outperform the other approaches on the translated data set with regard to all three metrics considered (see table 6.4). Both RNN paradigms perform competititvely, while the model with attention yields a slightly better accuracy, and the model without attention achieves higher sensitivity and F1 scores.

Among the baselines, the Flair embeddings outperform the other feature extraction methods, but the gain in performance is not very significant. The TF-IDF model works almost as well as the more

[4] The MBERT model is fine-tuned on the three-way task, as training on the two-way task did not yield an improvement in this case.

Input	English (Original Dataset)		German (Translated Dataset)	
	3-Class	**2-Class**	**3-Class**	**2-Class**
Bag of Words (tf-idf)	0.6196	0.7602	0.5961	0.7422

(a) Results for the baseline tf-idf features

Input	English (Original Dataset)				German (Translated Dataset)			
	3-Class		**2-Class**		**3-Class**		**2-Class**	
	PTRN	**SNLI**	**PTRN**	**SNLI**	**PTRN**	**SNLI**	**PTRN**	**SNLI**
GloVe (mean pooling)	0.6315	0.5552	0.7613	0.7024	0.5608	0.5192	0.7257	0.7042
Doc2Vec (DM + DBOW)	0.5193	0.6150	0.7056	0.7745	0.4621	0.5999	0.6856	0.7647
Flair (mean pooling)	0.6444	0.4724	0.7703	0.6794	0.6403	0.4045	0.7807	0.6736

(b) Results for the state-of-the-art text embeddings

Input	English (Original Dataset)		German (Translated Dataset)	
	3-Class	**2-Class**	**3-Class**	**2-Class**
RNN (translation setting)	0.6375	0.7736	0.6203	0.7457
RNN (autoencoder setting)	0.6179	0.7697	0.6045	0.7597

(c) Results for the embeddings obtained from the sequence-to-sequence RNN models

Input	English (Original Dataset)		German (Translated Dataset)	
	3-Class	**2-Class**	**3-Class**	**2-Class**
RNN (end-to-end without attention)	0.6537	0.7734	**0.6816**	0.7847
RNN (end-to-end with attention)	0.6457	0.7555	0.6657	**0.7920**

(d) Results for the end-to-end RNN models

Input	English (Original Dataset)		German (Translated Dataset)	
	3-Class	**2-Class**	**3-Class**	**2-Class**
MBERT (fine-tuned on SNLI)	**0.7447**	**0.8508**	0.5873	0.7435

(e) Results for the MBERT models, fine-tuned on the downsampled (English) and machine-translated (German) training sets, respectively.

Table 6.3: A summary of our classification results, with respect to accuracy. An MLP with one hidden layer was used as predictor for the unsupervised (tf-idf, GloVe, Doc2Vec, Flair) embeddings. The best performance values per configuration (in our case language + classification task) are marked in boldface.

sophisticated methods. In case of GloVe and Flair, the pre-trained embeddings yield better results than the ones trained on SNLI, while for Doc2Vec, the opposite is the case. The Sequence-To-Sequence RNN achieves results comparable to the other baseline methods, verifying that it is a valid approach to obtain paragraph embeddings for the NLI task.

	Acc.	F1	Sens.
tf-idf	0.7422	0.5761	0.5317
Flair	0.7807	0.6188	0.5400
RNN (without att.)	0.7847	**0.6742**	**0.6759**
RNN (with att.)	**0.7920**	0.6584	0.6083
MBERT	0.7435	0.5309	0.4305

Table 6.4: Performance comparison for tf-idf, Flair, both RNN models and MBERT, with respect to accuracy, F1-Score (for the "contradiction" class) and sensitivity, evaluated on the translated SNLI data set. A classifier predicting only "no contradiction" would yield an accuracy of 0.66, which is due to the slightly inbalanced label distribution. Thus, all investigated approaches perform significantly above this baseline.

		Predicted Label	
		No contradiction	Contradiction
True label	No contradiction	0.84	0.16
	Contradiction	0.37	0.63

(a) Original data set

		Predicted Label	
		No contradiction	Contradiction
True label	No contradiction	0.90	0.10
	Contradiction	0.46	0.54

(b) Translated data set

Table 6.5: Normalized confusion matrices for the output of the MLP classifier with one hidden layer, trained on the mean-pooled pre-trained Flair embeddings of the original and translated SNLI test data, respectively.

Looking at the confusion matrices (Table 6.5), we see that the performance of the classifier is far better on the negative class ("no contradiction") than on the positive class ("contradiction"). In other words, the classifier can identify non-contradicting statements relatively well, but has trouble finding the actual contradictions. This might be due to the fact that none of the baseline features addresses the problem described by [73], that is contradicting words will most likely be mapped to similar positions in feature space. Some contradictions that are more subtle and not tied to certain words, might also not be recognized by those simple approaches (see Table 6.6 for examples). Apparently, this issue cannot be solved by the Sequence-To-Sequence model either.

Comparing the performances for the English and German data, there is no significant difference. Overall, the models perform slightly better on the English data set, which is not surprising given the artifacts of the machine translation (see Section 6.2). Still, the drop in performance is not as dramatic as one could have expected. Also, the differences in performance between the simple feature extraction methods are similar for both languages, indicating that the translated data set preserves the general properties of the original data quite well.

In case of the more advanced method, we see a slightly different picture. Even though the RNN-based approaches also outperform MBERT on the machine-translated data, this is not the case for the original English data set. In the latter case, MBERT performs significantly superior to all other approaches, achieving an accuracy of 0.85 for the two-class task. One possible explanation for this could be, that the model has been optimized to deal with English data, as there are significantly more English training resources for MLM and TLM than for any other language. Also, it might be an indication that the characteristics of the machine-translated data differ more strongly from those of a human-created data set, than we initially assumed.

6.4.2 Qualitative Evaluation

We provide some examples for the prediction results with Flair embeddings on both data sets in Table 6.6. It is worth noting that in most cases (7,959 out of 9,824 examples), the classifiers for English and German agree on the prediction. This indicates a high level of coherence between the two versions, and thus a high translation quality. Those cases where both predictions are wrong, can mostly be accounted to one of the main difficulties in NLI we already addressed earlier: It is hard for a classifier to identify a contradiction if the sentences differ just in one word (example 3), since the feature representations are most likely similar. Also, lack of world knowledge is an issue (example 4). Considering those sentence pairs where the predictions are different for German and English, the main sources of error are a slightly wrong translation (examples 5 and 6), or a typing error in the original (example 7). Interestingly, in case of example 6, the incorrect translation (“masses" → “Messen") seems to be in favor of the target, as it fits the context. Overall, the recall is higher for the German data than for English (see Table 6.5), but the precision is lower. This indicates that our classifier is less confident about predicting a contradiction than it is for English, but when it does, its probability of being right is higher.

Some examples for the predictions of the end-to-end RNN models are shown in table 6.7. The model with attention often catches contradictions that are tied to just one word (example 1), if the sentences are very similar otherwise. This is most likely due to the attention mechanism, aligning the words that should match, and spotting those that are not. On the other hand, it can also wrongly predict a contradiction if two words are actually just different expressions for the same thing (e.g. "Hausparty" and "Weihnachtsparty" in example 2). Also, the model struggles with sentences that seem completely unrelated (example 3), but labeled "contradiction", as they are supposed to refer to the same picture (see [13] for details on the data properties). Furthermore, both models fail to recognize a contradiction if world knowledge is required (example 4).

Some introspection of the models is possible, when examining the activation values of the attention layer. Ideally, those should correspond to co-importances of two words from premise and hypothesis. In figure 6.3, some examples are displayed. While most of the values are rather inconclusive, in some cases they correspond to correct alignments (i.e. "Jungen" in figure 6.3(a)). It is also interesting to see, that the model is able to focus on the relevant part of a longer sentence (figure 6.3(b)), even though it does for the most part not exactly align words that are referring to the same thing.

	Original sentence pair	Translated sentence pair	Label predicted for English	Label predicted for German	True label
1	**P:** “One tan girl with a wool hat is running and leaning over an object, while another person in a wool hat is sitting on the ground." **H:** “A boy runs into a wall"	**P:** “Ein braunes Mädchen mit Wollmütze läuft und lehnt sich über ein Objekt, während eine andere Person mit Wollmütze auf dem Boden sitzt." **H:** “Ein Junge läuft gegen eine Wand."	Contradiction	Contradiction	Contradiction
2	**P:** “A young family enjoys feeling ocean waves lap at their feet." **H:** “A family is at the beach."	**P:** “Eine junge Familie genießt es, die Meereswellen zu ihren Füßen zu spüren." **H:** “Eine Familie ist am Strand."	No contradiction	No contradiction	No contradiction
3	**P:** "A man wearing a gray ball cap walks next to a red-headed woman wearing a long-sleeved blue jean shirt." **H:** “The man is wearing a blue cap."	**P:** "Ein Mann mit einer grauen Ballkappe geht neben einer rothaarigen Frau mit einem langärmeligen blauen Jeanshemd." **H:** “Der Mann trägt eine blaue Kappe."	No contradiction	No contradiction	Contradiction
4	**P:** “A black man in a white uniform makes a spectacular reverse slam dunk to the crowd’s amazement." **H:** “the man is asian"	**P:** “Ein schwarzer Mann in weißer Uniform macht einen spektakulären Reverse-Slam Dunk zum Staunen der Menge." **H:** “der Mann ist Asiate"	No contradiction	No contradiction	Contradiction
5	**P:** “A young boy in red leaping into sand at a playground." **H:** “A child does a cannonball into a pool."	**P:** “Ein kleiner Junge in Rot springt auf einem Spielplatz in den Sand." **H:** “Ein Kind spielt eine Kanonenkugel in einen Pool."	Contradiction	No contradiction	Contradiction
6	**P:** “This church choir sings to the masses as they sing joyous songs from the book at a church." **H:** “A choir singing at a baseball game."	**P:** “Dieser Kirchenchor singt zu den Messen, während sie fröhliche Lieder aus dem Buch in einer Kirche singen." **H:** “Ein Chor, der bei einem Baseballspiel singt."	No contradiction	Contradiction	Contradiction
7	**P:** "A blond-haired doctor and her African american assistant looking threw new medical manuals." **H:** “A doctor is looking at a book"	**P:** “Eine blonde Ärztin und ihre afroamerikanische Assistentin, die auf der Suche nach neuen medizinischen Handbüchern warf." **H:** “Ein Arzt schaut sich ein Buch an."	No contradiction	Contradiction	No contradiction

Table 6.6: Prediction examples for English and German, in comparison. An MLP classifier with one hidden layer, trained on mean-pooled Flair embeddings was used to obtain the results for both languages.

	Translated / original sentence pair	Prediction model 1	Prediction model 2	True label
1	**P:** "Der Mann trägt ein orange-schwarzes Poloshirt und kniet mit seiner Lunchbox in der einen Hand und hält eine Banane in der anderen Hand." - *"The man is wearing an orange and black polo shirt and is kneeling with his lunch box in one hand while holding a banana in his other hand."* **H:** "Ein Mann trägt ein grünes T-Shirt und hält eine Banane." - *"A man is wearing a green t shirt while holding a banana."*	Contradiction	No contradiction	Contradiction
2	**P:** "Hier ist ein Bild von Leuten, die sich auf einer Hausparty betrinken." - *"Here is a picture of people getting drunk at a house party."* **H:** "Die Leute feierten auf einer Weihnachtsparty." - *"People were celebrating at a Christmas themed party."*	Contradiction	No contradiction	No contradiction
3	**P:** "Stadtlandschaft mit vier Männern auf dem Fahrrad, die als Mittelpunkt des Bildes dient." - *"Cityscape with four men on bikes serving as the focal point of picture."* **H:** "Lokale Räuber fliehen, nachdem sie einen Otter gestohlen haben." - *"Local robbers escape after stealing an otter"*	No contradiction	Contradiction	Contradiction
4	**P:** "Ein schwarzer Mann in weißer Uniform macht einen spektakulären Reverse-Slam Dunk zum Staunen der Menge." - *"A black man in a white uniform makes a spectacular reverse slam dunk to the crowd's amazement."* **H:** "der Mann ist Asiate" - *"the man is Asian"*	No contradiction	No contradiction	Contradiction

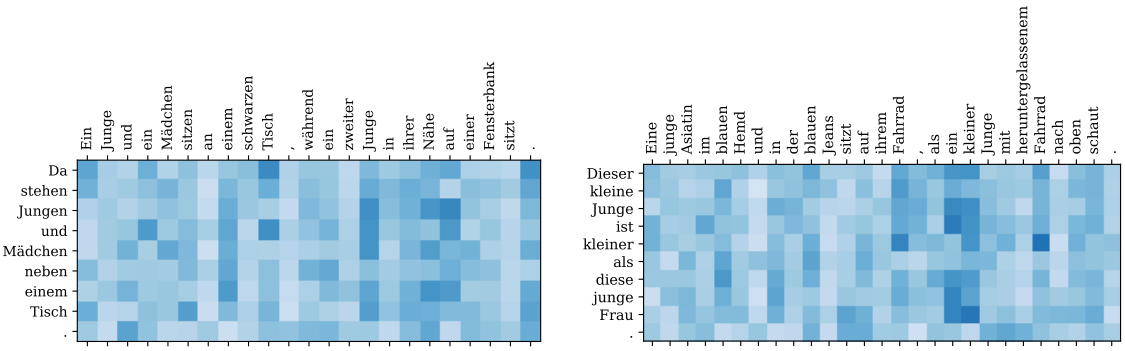
Table 6.7: Prediction examples for the end-to-end models with (model 1) and without (model 2) attention, in comparison. The attention-based model can identify contradictions that are linked to specific words (example 1), but it also returns wrong predictions for examples where the alignment between the sentences is not helpful for the classification (examples 2 and 3). Both approaches cannot cope with sentences where world knowledge is involved (example 4).

6.5 Conclusion and Summary

In this chapter we presented a Machine Translation based method to generate data for the CD/NLI task. Furthermore, we evaluated several Neural Network based approaches to investigate the effect of the translation on data and prediction quality.

We can conclude that neural Machine Translation is capable of producing large amounts of high-quality data. This suggests that the approach could be successfully applied to other data sets and different text mining tasks, overcoming the limitation to the English language. We could verify the superior performance of the end-to-end models, compared to models trained on previously extracted embeddings. Our RNN approaches also outperform the state-of-the-art MBERT transformers on the translated data set. They are especially able to correctly classify sentences that contradict based on one or just a few words, but are otherwise very similar. Still, they struggle with other phenomena such as hyponymity and world knowledge.

There are no essential distinctions to be made between the two languages throughout our NLI pipeline, except for a slightly different pre-processing policy. The approach of using the encoder RNN's hidden states as paragraph embeddings also shows good results, and could possibly be further



(a) (b)

Figure 6.3: Attention matrices of the model after inputting two sentence pairs from the translated SNLI test data set. The x-axis corresponds to the premise, and the y-axis corresponds to the hypothesis.

improved with more time invested into model design and parameter tuning.

For future work, it would be important to assess the generalizability of the models that were trained on machine-translated data. It could be worth investing in the collection of additional training and evaluation resources for the German language, e.g. by utilizing the MultiNLI [116] and XNLI [91] corpora. We also plan to collect new data sets for the Contradiction Detection task from online sources in German language.

A main limitation of our approach is the lack of world knowledge. All of our trained models fail to recognize contradictions that assume such background, which is not explicitly encoded in the training data. In order to address this problem, an NLI model would need to incorporate external knowledge into its learning process. This could be done e.g. using informed Machine Learning approaches as suggested by [9] or [73] (see chapter 4). Another option would be to involve user input during training.

The next chapter will explore a different technique for obtaining data in low-resource scenarios, namely augmentation using generative transformer models and linguistic knowledge.

CHAPTER 7

Data Augmentation Using Generative AI and Linguistic Rules

We continue our investigation on data generation for LM training, by introducing a hybrid approach for data set construction using linguistic features and generative AI. Here, we turn our focus back to answering RQ1:

Research Question 1 (RQ1)

Can training with linguistically informed objectives and/or data augmentation improve language model efficiency in terms of training set size, model size and/or training time, while maintaining downstream performance?

In particular, we investigate four approaches that utilize semantic and syntactic features, as well as state-of-the-art generative AI to obtain a concise, prototypical data set for LM training on the Contradiction Detection task. We hypothesize that using this data set can help reducing the resources needed for training, as it will provide the models with condensed knowledge about the essential features of contradictions.

The key contributions of this chapter are:

- We present four different approaches for rule-based, hybrid and LLM-based data generation, opening a new avenue of research in the context of informed NLU.
- We provide insights into the linguistic features of contradictions, building upon and extending the work by [9].
- We conduct an experimental validation of our proposed data generation approaches, augmenting reduced versions of the SNLI data set with prototypical samples. Our results show that adding those prototypes to the training data helps maintaining baseline performance, while significantly reducing the data set size.

The initial idea for this research direction was developed by Maren Pielka, who also conducted part of the data augmentation pipeline and supervised the research work. Svetlana Schmidt did most of

the implementation, both with respect to data augmentation and model training. She also wrote her Master's thesis [117] about this topic, which was supervised by Maren Pielka and Ralf Klabunde. The data augmentation approach using Named Entities and generative AI for paraphrasing was developed and implemented by Marie-Christin Freischlad. The papers were written by Maren Pielka, Svetlana Schmidt and Marie-Christin Freischlad, while the contribution was equally shared among Maren Pielka and Svetlana Schmidt for the first one [10], and among Maren Pielka and Marie-Christin Freischlad for the second one [118].

This chapter is based on the following publications [10, 118]:

- **Maren Pielka**, Svetlana Schmidt, and Rafet Sifa. 2023. "Generating Prototypes for Contradiction Detection Using Large Language Models and Linguistic Rules." In 2023 IEEE International Conference on Big Data (BigData), pages 4684-4692, Sorrento, Italy. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/BigData59044.2023.10386499
- **Maren Pielka**, Marie-Christin Freischlad, Svetlana Schmidt, and Rafet Sifa. 2025. "Improving Language Model Performance by Training on Prototypical Contradictions." In Advances in Information Retrieval (ECIR 2025), pages 148-155, Lucca, Italy. Springer. DOI: https://doi.org/10.1007/978-3-031-88714-7_12

After a brief introduction, we will introduce our data generation approaches in section 7.4, and conduct a qualitative evaluation of the generated samples (section 7.5) This initial investigation is followed by a quantitative assesment using the new data for training an ML model, see section 7.6. Some limitations of the approach are discussed in section 7.7.

7.1 Introduction

Detecting contradictions in text is one of the hardest tasks for a language model to comprehend. This is due to the complex semantic nature of contradictions, and the variety of contexts in which they can occur. For this reason, a multitude of data sets and models have been developed to solve this task. Meanwhile, the recent onset of large generative language models has given rise to new possibilities for problem solving as well as data augmentation, which we aim to explore in this work.

Contradiction Detection (CD) is a subtask of Natural Language Inference (NLI), but has also been investigated independently in recent years. The goal is, given two pieces of text (premise and hypothesis) that are assumed to refer to the same fact or event, to predict whether there is a contradiction between those. To this end, a contradiction is defined as a mismatch between two statements, such that they cannot possibly be true at the same time. There is of course some subjectivity involved in judging whether two statements are strictly contradicting, or just slightly deviating. Also, the context in which the statements occur can play a crucial role.

Our goal is to build a data set for Contradiction Detection (CD) that conveys different prototypes of contradictions. The idea is to "condense" the essential linguistic properties of contradictions into a relatively small data collection, thereby reducing the computational resources needed to train models for solving this task. Thus, a comparatively small (< 1B parameters) language model could be fine-tuned on such a data set and achieve competitive results to a larger model, which has seen a lot more data. In order to create the data, one first has to understand the semantic and linguistic

phenomena that contradictions arise from. Those can be very diverse, comprising simple cases of antonymy and negation, as well as more complex ones such as mismatch in syntax, stated facts or deviation in the context of background knowledge. To this end, a side-goal of our work is to extend the typology by [9] and to include more fine-grained contradiction types. We employ an automated method for generating contradictions using rules and large language models (LLMs), which is easily scalable. Our approach for generating the data set is three-fold:

1. We generate samples in a rule-based manner by exploiting semantic knowledge graphs and syntactic parsing.
2. We instruct a large generative language model to produce contradicting hypotheses, given premises from the standard NLI corpus, SNLI [13], both using a hybrid (based on Named Entity Recognition) and a purely generative approach.
3. We instruct a large generative language model to produce completely new contradicting statements (both premise and hypothesis), as well as new types of contradictions.

The intuition is to use linguistic and factual rules where this is applicable, i.e. for contradictions based on antonymy, negations and numeric mismatch. For more complex relations such as factive or structural contradictions, we instruct a generative model to produce new samples, either based on given premises or on the type description alone. The resulting data set is then being used to augment existing ones such as SNLI [13], while drastically reducing the overall data set size. We want to show that models trained on this reduced prototypical data set can perform competitively with those trained on much more data, thereby significantly reducing the cost in terms of energy consumption and storage space needed. Furthermore, our vision is to provide a method to generate more data without much effort, and to broaden the understanding of the complex linguistic nature of contradictions. The code for this chapter has been published on Github[1]. To our knowledge, this is the first work implementing such a hybrid data generation method with respect to NLI.

7.2 Related Work

The Stanford Natural Language Inference (SNLI) corpus is a freely available data set which contains 570K sentence pairs [13]. The labeled sentence pairs were written by humans based on image capturing. The data was collected with help of the Amazon Mechanical Turk[2]. The human workers were asked to provide a hypothesis for a premise scene description. The hypotheses should entail, contradict, or be neutral toward the preexisting premise [13].

One approach for creating more realistic data for NLI was presented by [119]. The texts for premises were collected from news articles. The hypotheses were generated in several ways, using information and relation extraction, question answering, and summarization systems.

A more fine-grained system for detection of different types of textual contradictions was proposed by [9]. Following the reasoning in [9], there are two main categories of contradictions: 1) those which arise from antonymy, negation, and numerical mismatch, and 2) contradictions occurring from subtle lexical differences, contrasting structure of the sentences, factive mismatch, and contrast in the world

[1] https://github.com/fraunhofer-iais/informed_nlu/
[2] https://www.mturk.com/

knowledge (WK). It is quite difficult to automatically detect the contradictions arising from the second category. The understanding of such types requires the understanding of the sentence meaning.

Large (generative) language models (LLMs) have sparked great interest in recent years. Especially the Generative Pretrained Transformer (GPT) framework [4, 5, 120] from OpenAI has gained significant popularity - even outside the AI community - since the release of the ChatGPT conversational interface in late 2022. Their latest model GPT-4 [6] has set a new state of the art for many language understanding tasks, showing the capability to solve a broad range of real-world tasks such as academic exams on par with human performance. Nevertheless, there are still some shortcomings with respect to those models, for example the fact that they tend to produce incorrect output when asked about complicated or unknown events. Also, they require an extensive amount of data for pre-training, as well as powerful computing resources both for training and inference. There has been some work on utilizing LLMs to create new training data and problem descriptions. Wang et al. [121] introduce "Self-Instruct", a framework which can be used to extend the language understanding capabilities of LLMs by having it produce instructions and training instances for language understanding tasks. Those generated instances can then be used to train the LLM further. Our approach is inspired by that idea, but we focus on the more specific task of detecting contradictions using a linguistic typology.

The idea of training language models with prototypical knowledge is inspired by [64], who suggested to use a similar approach in image classification. They argue that the condensed knowledge of a target domain can be represented by a relatively small data set, which contains training samples that are typical manifestations of the task at hand.

There has been some previous work regarding the analysis of the linguistic nature of contradictions, as well as methods to make use of those features in a language modeling setup. [20] examine some semantic aspects that are hard to comprehend for Machine Learning models, such as difference in local prepositions or metaphors. [21] build upon those findings by introducing a linguistically informed pre-training regime for encoder-based transformer models, utilizing information about part of speech (POS) tags, synsets and syntactic dependencies. We aim to extend this work by presenting an informed data generation approach which can be used to efficiently fine-tune language models for Contradiction Detection.

7.3 Data Acquisition

The data augmentation is mainly based on the Stanford Natural Language Inference (SNLI) [13] corpus, which contains 570K sentence pairs that are annotated with the labels “entailment”, “neutral” and “contradiction”. Only the premises from SNLI are used to generate contradicting hypotheses (see section 7.4). In addition, we utilize news headlines from the Deutsche Welle website[3] for creating world knowledge contradictions. For our NER-based approach, we use five datasets of BBC News articles, available on Huggingface.co.[4] The datasets include BBC News from March to July 2023 and comprise a total of 9,389 samples. For our purposes, only the columns “description” and “section”

[3] https://www.dw.com/en/top-stories/s-9097, accessed at 24. 8. 2024

[4] https://huggingface.co/datasets/RealTimeData/bbc_news_march_2023,
https://huggingface.co/datasets/RealTimeData/bbc_news_april_2023,
https://huggingface.co/datasets/RealTimeData/bbc_news_may_2023,
https://huggingface.co/datasets/RealTimeData/bbc_news_june_2023,
https://huggingface.co/datasets/RealTimeData/bbc_news_july_2023

are relevant, as they are used as text base for contradiction generation and data pre-sorting.

7.4 Data Augmentation

We argue that the data set for the CD task should include contradictions created in different ways. The generation of contradictions is based on the idea that they arise from different lexical features [9]. We generate contradictions based on antonymy, negation, and numerical mismatch using a rule-based approach. The more complex types of contradictions, as described by [9], are generated via the GPT frameworks. The general approach is to manipulate a given premise statement in order to obtain a contradictory hypothesis.

7.4.1 Generating Contradictions Based on SNLI, Using Linguistic Rules

For the construction of the hypotheses via linguistic rules, the syntactic and semantic features are extracted from the premises of the SNLI data set. The stanza[5] Dependency Parser is utilized for extraction of dependencies, POS-tags and morphological features of each sentence.

For the generation of the antonymy based contradictions we take advantage of the WordNet [78] framework. The content words in WordNet are grouped into synsets (synonym sets) based on their semantic similarity. One meaning of a word is represented by a specific synset.

Contradictions based on antonymy arise when the hypotheses contain antonyms of the aligned words of the premise [9]. We extract the antonyms of the adjectives and nouns in the premise from WordNet. The hypothesis is then created by replacing the objects and the adjectives of the premise with their antonyms. The meaning of each premise first has to be disambiguated. The disambiguation includes defining one distinct synset for each word in a sentence if available. The SyntagNet API[6] is utilized for extracting one meaning of each word in a sentence. This step is necessary since there could be several possible synsets for one word.

The POS-tags, dependencies and morphological features are used for creating the contradictions based on numerical and polarity mismatch. For the contradiction type *negation*, the contradictory hypotheses are created by negating the verbal phrases of the premises. The morphological features are utilized for exerting the right type of the negation, for example the verb phrase in singular and present tense yields the negative particle *not* and the modal verb *do*.

In order to create numerical mismatches, we make use of the dependency *nummod*[7] (numerical modifier). The idea is to create the hypothesis containing numbers bigger or smaller than the ones in the premise.

7.4.2 Generating Contradictions Based on SNLI, Using LLMs

The nature of other contradiction types is more complicated. Generating lexical, structural and factive contradictions, as well as those arising from world knowledge (WK) contrasts, requires more than just changing a pair of words in the hypothesis. The data set collected by [9] RTE3[8] consists of "real−life"

[5] https://stanfordnlp.github.io/stanza/depparse.html
[6] http://syntagnet.org/api-documentation
[7] https://universaldependencies.org/u/dep/nummod.html
[8] https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

contradictions which were additionally annotated for their type, and the typology described in their paper is the base for defining the important features of the contradiction types in this work. The semantics of the verb embedding influence the meaning of the whole sentence, and can serve as a basis for entailment or contradiction [122]. According to [9], factive contradictions arise from the context in which the verb phrase is embedded, for example *Sudan was ready to accept U.N. troops in Darfur* contradicts *Sudan refused to accept U.N. troops in Darfur*[9]. The opposite meaning of the verb phrase in the hypothesis also creates the factive contradiction:

P: Sudan refused to allow U.N. troops in Darfur.
H: Sudan will grant permission for United Nations peacekeeping forces to take up station in Darfur.[10]

Structural contradictions arise from the mismatch between the syntactical structures of the premise and hypothesis. This contrast occurs from replacing the object of the verb with the subject from the premise or with a new entity. For example, replacing *parents* in *The children are smiling and waving to their parents* with *friends*: *The children are smiling and waving to their friends* creates a contradiction.

As specified by [9], lexical contradictions are the type of contradictions which can arise from the distinctive views on the identical event as it is shown in the following example.

P: Two women who just had lunch hugging and saying goodbye.
H: The two women who just ate lunch ignored each other and left without saying a word.

The WK contradictions arise from a mismatch in the information regarding one unique event or well-known person [9]. The example from [9] illustrates this kind of contradiction:

P: President Kennedy was assassinated in Texas.
H: Kennedy's murder occurred in Washington.

Utilizing LLMs allows us to produce a large amount of samples. We use the GPT-4 model with the maximum number of generated tokens set to 512, and the temperature parameter set to 1 for obtaining diverse output. We instruct the GPT-4 model to generate contradictions of each type for every premise and additionally provide it with the descriptions of the different contradiction types and some examples for those contradiction types. The prompt and complete list of the contradiction types' descriptions can be found in the Appendix, sections A.1 and A.2.

7.4.3 Generating Contradictions Using Named-Entity Recognition and GPT-4 for Paraphrasing

Structural contradictions arise when "the syntactic structures of premise and hypotheses create contradictory statements." [9] In order to create structural contradictions, we employ an approach based on named-entity recognition (NER). We use BBC News descriptions as text base to generate samples with real-world relevance. The ones mentioned in section 7.3 are particularly suitable because all texts are thematically presorted by their publication section. Additionally, the description of each article is well-sized for our task and does not require further cleaning of unwanted text artifacts. As

[9] https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml
[10] https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

some sections only contain few articles, or even just one, we changed their section labels to gather them into larger section groups. The processing algorithm is applied individually to each publication section and contains the following steps:

- In a first iteration all sentences are tokenized and annotated using the SpaCy transformer model[11] and the default SpaCy pipeline, which contains part of speech tagging, dependency parsing and named-entity recognition.
- In a further step we create an entity pool: We exclude all SpaCy NER categories of mixed entities, as they cannot be used interchangeably.[12] Instead we import a database of entities from closed categories such as “countries” or “nationalities” as reference objects. Moreover, we added a column with fictitious personal names to make use of the SpaCy NER category "person" as well.
- For generating a contradictory hypothesis pair, each sentence is then processed again by replacing the first entity in the object position with another entity from the same category. Subsequently, the sentence is paraphrased by GPT-4o mini[13], explicitly retaining the exchanged entity.
- For non-contradictory samples, we simply instruct GPT-4o-mini to rephrase the original statement without object alteration.

In this manner, we create both contradictory and non-contradictory pairs of premises and hypotheses and add them to our dataset. We validate the generated contradictions by prompting GPT-4o[14] and manually check a 10% sub-sample to ensure that our judgment mainly aligns with the model’s findings. Due to time constraints, it was not possible to validate the whole dataset manually.

7.4.4 Generating Contradiction Types and Instances Based only on Instructions

As a fourth approach, we instruct an LLM to generate new instances of specific contradiction types, as well as new typologies. We do not provide the model with any external data, except the descriptions for pre-defined contradiction types, following the typology by [9]. This approach is inspired by the idea of [121], who proposed to utilize LLMs to jointly generate instructions and instances for language understanding tasks.

The generation process is structured as follows: In every iteration, a fixed number of contradiction samples is being generated for each contradiction type. Additionally, a new type description is being generated, given three randomly sampled descriptions of existing types as examples. The new description is then added to the pool and being used in later iterations for generating both new instances and new types. This policy is depicted in figure 7.2.

We initially start with a pool of contradiction types that contains the classes of contradictions based on structure, lexicality, facts and embedding context as well as verbal antonymy. The prompt templates

[11] https://spacy.io/models/en#en_core_web_trf

[12] For instance, “the Pacific Ocean” and “London” both belong to the category “Location”. However, the statement “President Biden pays visit to the Pacific Ocean” holds no proper meaning.

[13] https://platform.openai.com/docs/models/gpt-4o-mini

[14] https://platform.openai.com/docs/models/gpt-4o

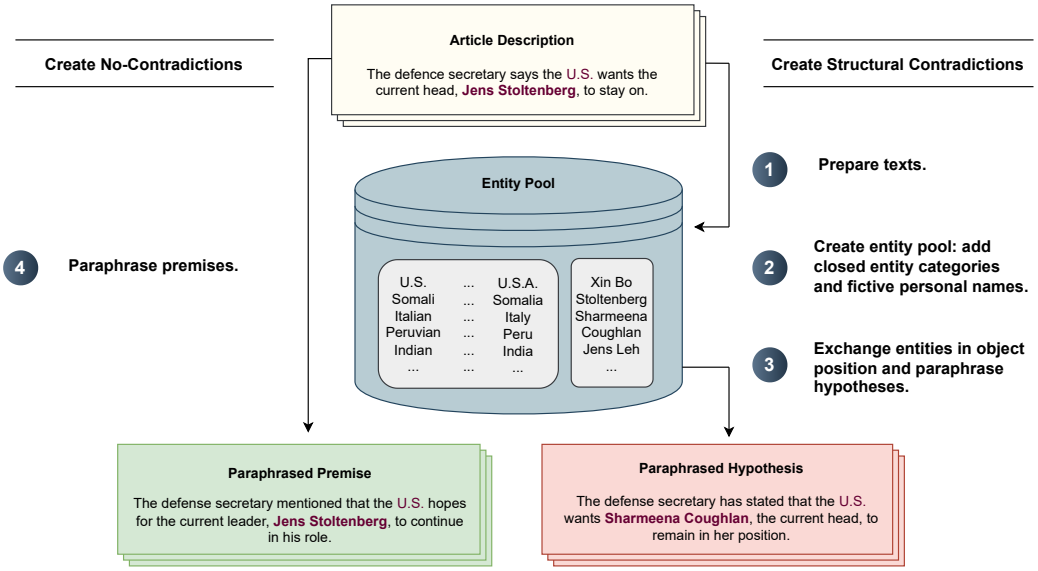


Figure 7.1: Illustration of the structural contradiction dataset generation approach.

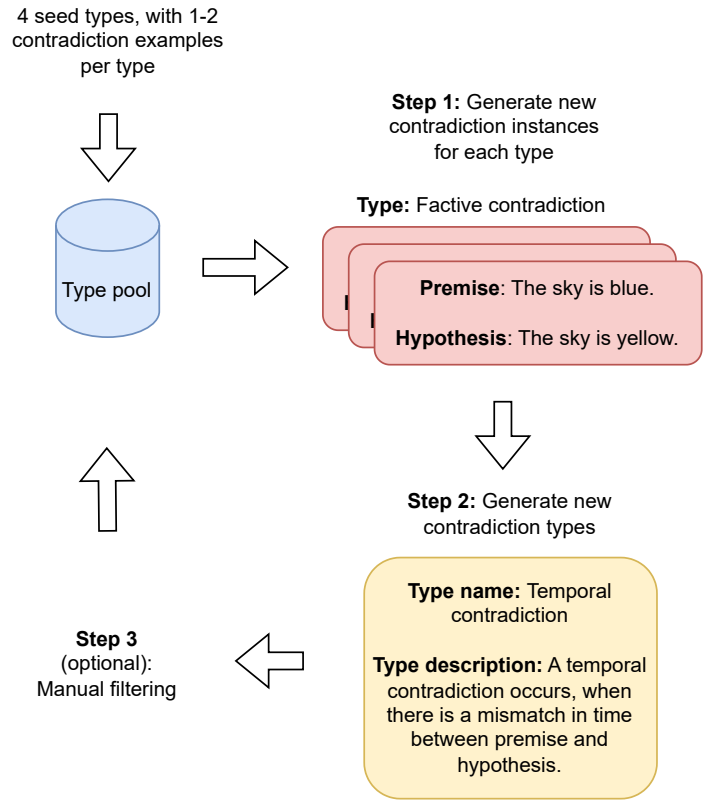


Figure 7.2: Illustration of the multi-step generation approach for contradiction types and samples.

	Method 1 (rule-based)			Method 2 (GPT with SNLI)				Method 3 (GPT only)				
	Antonymy	Numerical	Negation	Factive	Structure	Lexical	WK	Factive	Structure	Lexical	WK	Other
Number examples	170	165	165	125	125	125	125	50	50	50	50	225

Table 7.1: Number of generated examples for the three generation methods, per contradiction type. "Other" stands for all new contradiction types generated by GPT (see Appendix A).

for generating those contradictions can be found in the Appendix A.1. For the generation of new contradiction types, GPT-4 is being used. For generating new instances, we experiment with both GPT-3.5 and GPT-4. The parameters of the API call are the same as specified above for the second method.

7.5 Theoretical Assumptions and Qualitative Results

Our main assumption is that a data collection which includes most features of contradictions can help to improve the performance of smaller transformer models such as BERT [41] in Contradiction Detection. For an initial qualitative evaluation, we collect around 1500 samples in total. Among those, 1000 sentence pairs are generated with rules and GPT using SNLI as shown in 7.1 (methods 1 and 2), and 973 using NER and GPT for paraphrasing (method 3). The number of samples generated using method 4 with GPT only is about 500 sentence pairs (see 7.1).

7.5.1 Method 1: Contradictions Generated Based on SNLI, Using Linguistic Rules

The following examples illustrate the contradictions generated with the rule-based approach:

Antonymy:

P: Women exercising one *woman* has a green mat and black outfit on.
H: Women exercising one *man* has a green mat and black outfit on.

P: Two *blond* women are hugging one another.
H: Two *brunet* women are hugging one another.

Numerical:

P: *Two* blond women are hugging one another.
H: *Three* blond women are hugging one another.

Negation:

P: Two blond women *are hugging* one another.
H: Two blond women *are not hugging* one another.

As it can be seen from the following pair of sentences, the hypothesis created with the rule-based approach can contain semantic and grammatical errors:

P: A **young** girl sitting at a table with a bowl on her head.
H: A **old** girl sitting at a table with a bowl on her head.

Thus far, the rules which were used generating contradictory hypotheses are simple. They do not include the semantics of the sentence, and adjustment of the grammatical forms. Nevertheless, the data could still be useful for training a language model.

7.5.2 Method 2: Contradictions Generated Based on SNLI, Using LLMs

Here are some examples for hypotheses which have been generated using GPT-4, given the respective premise:

Factive:

P: Children are smiling and waving at the camera.
H: Children are crying and ignoring the camera.

P: Children are smiling and waving at the camera.
H: Children forgot to smile and wave at the camera.

Structure:

P: A couple is playing with a little boy on the beach.
H: A couple is playing with a dog on the beach.

Lexical:

P: A boy is jumping on skateboard in the middle of a red bridge. H: The kid is sitting while riding his bike at the end of a green passage.

World Knowledge:

P: A person on a horse jumps over a broken down airplane.
H: Airplanes are too large and tall for a horse to jump over.

One of the difficulties of the contradiction generation with the GPT-4 model is the limited number of sentences it can generate at one time. Another complexity in generating the contradictions is that the GPT-4 model can produce sentences with semantic errors or contradicting to world knowledge, as it is illustrated in the following example:

P: A person on a horse jumps over a broken down airplane.
H: The broken down airplane overleaps the person on a horse.

7.5.3 Method 3: Contradictions Generated Using NER and LLMs for Paraphrasing

The method based on named-entity recognition allows generating some sort of structural contradictions, e.g.

P: The late Queen singer left his house and its contents, including lyrics and costumes, to Mary Austin.
H: The late Queen vocalist bequeathed his home and its belongings, which comprise lyrics and costumes, to Jodie Smith.

Since it seems unlikely, under real conditions, that a statement would appear with the same wording and sentence structure, except for the object entity, we decided to paraphrase the sentence after replacing the object entity. This approach aims to increase the complexity of Contradiction Detection. It also helps make the process more aligned with real-world situations, even though it may introduce more noise into the dataset. However, there is still a relevant number of sentence pairs that do not meet the established contradiction criteria, especially due to missing world knowledge and the difficulties that have already been described by [9]. For instance, the method systematically creates a meaningful hypothesis from the premise "*Rishi Sunak has presented Dáithí Mac Gabhann with an award while in Belfast to see Joe Biden*". However, the hypothesis "*Rishi Sunak awarded Rory Gray during his visit to Belfast to meet Joe Biden.*" does not necessarily contradict the premise in our understanding, as the former British Prime Minister could have honored both individuals at the same event. In other cases, hypotheses emerge that also present contradictions within themselves due to missing world knowledge, as seen with "*Tunisia*" and "*Africa*" in the hypothesis "*Numerous migrants embark on boats from Tunisia in their pursuit of reaching Africa, yet the outcomes can often be devastating*". Consequently, all contradictions must be validated afterwards, for which even high-performance LLMs are still only partially suitable, resulting in remaining noise in the dataset. Further manual validation or alternative quality control mechanisms could improve the robustness of the dataset and positively influence the results.

7.5.4 Method 4: Contradictions Generated Using LLMs only

With respect to the fourth method, where we instruct the LLM to produce both premise and hypothesis given a contradiction type description, there are significant differences in quality between the contradiction types. For some cases the generation works reasonably well, as can be seen in this example of a lexical contradiction:

P: The cat is sleeping peacefully on the couch.
H: The cat is wide awake and running around the room.

In other cases - specifically for structural contradictions - the approach does not work well at all, as the language model produces grammatically correct, but semantically meaningless hypotheses:

P: He cooked delicious pasta for dinner.
H: The pasta cooked him deliciously for dinner.

Surprisingly, switching from GPT-3.5 to GPT-4 for the instance generation does not change the quality of the results much.

As described in section 7.5, we also instruct GPT-4 to generate completely new types of contradictions. This works surprisingly well, as can be seen in this example (both the type description, as well as the instance have been generated by GPT-4):

Temporal mismatch

Description: This contradiction arises when there's inconsistency between the time frames or chronological events presented in two statements. Hypothetically, if one statement indicates an event happening before another, and the contradictory statement implies the opposite sequence or suggests the events are simultaneous, a temporal mismatch is present.

Example:
P: The movie was released two months ago.
H: The actors are currently filming the sequel.

Most of the types produced by GPT are logically coherent and semantically meaningful. The LLM also manages to generate reasonable instances for each of those types. We observe that in some cases the same type of contradiction is effectively generated multiple times, even though the description varies slightly (e.g. for "Temporal mismatch"). Nevertheless, those new types of contradictions could possibly contribute to better understanding the underlying semantics, and offer a more fine-grained typology. A complete list of all newly generated types (after duplication removal) is provided in the Appendix A.3. The contradictions generated in that way were not included to the following quantitative evaluation, as we found it hard to control for data quality here.

7.6 Experiments and Results

For our experimental setup, we incrementally reduce the SNLI training set to 50, 30 and 5% of its original size, and replace part of the samples with the prototypes we generated using the methods described in section 7.4, s.t. the size of the resulting dataset does not change. So, the training sets in each experiment configuration (i.e. SNLI being reduced to 50, 30 and 5%, respectively) are equal in size. We are including the reduced SNLI dataset without prototypes as a baseline for each configuration, to show that adding the prototypes yields a performance improvement over the SNLI data of the same size. Detailed results are displayed in table 7.2. XLMRoBERTa[15] [22] with an Adam optimizer and cross-entropy loss is used for all experiments. The model is being trained for 10 epochs with a learning rate of 5^{-6}. We include the performance on the original (not reduced) SNLI training set for comparison.

It is evident from those results, that adding prototypical contradictions enables us to significantly reduce the number of training examples, while almost maintaining baseline performance. For all three experimental setups, the model trained on reduced SNLI and added prototypes performs better than the base model which was trained on the reduced SNLI data of the same size. Especially for the third experiment, where SNLI is reduced to 5%, this effect is noticeable. Comparing the different methods for generating prototypes, we see that while the first approach already leads to an improvement over the baseline, using both methods ("SNLI+Prototypes+Struct") performs best in most cases. The standard deviation of the models is overall rather low, indicating stable and reproducable results, except for the last experiment with "SNLI+Prototypes+Struct.". This might be because of remaining noise in the

[15] https://huggingface.co/docs/transformers/model_doc/xlm-roberta

	Accuracy	F1(Cont.)	F1(No Cont.)
not reduced SNLI	**96.34** (0.07)	**94.49** (0.1)	**97.26** (0.05)
Reduced size: 50%			
SNLI base	95.74 (0.1)	93.61 (0.7)	96.81 (0.9)
SNLI+Prototypes	95.86 (0.7)	93.75 (0.1)	96.91 (0.05)
SNLI+Prototypes+Struct.	**95.93** (0.09)	**93.88** (0.1)	**96.95** (0.07)
Reduced size: 30%			
SNLI base	95.58 (0.2)	**93.36** (0.3)	96.68 (0.2)
SNLI+Prototypes	95.38 (0.1)	93.05 (0.1)	96.54 (0.1)
SNLI+Prototypes+Struct.	**95.59** (0.1)	**93.36** (0.2)	**96.7** (0.1)
Reduced size 5%			
SNLI base	93.85 (0.1)	90.77 (0.2)	95.39 (0.1)
SNLI+Prototypes	**94.05** (0.1)	**91.06** (0.1)	**95.54** (0.09)
SNLI+Prototypes+Struct.	89.45 (9)	83.7 (14)	92.19 (6)

Table 7.2: Performance comparison for datasets of different sizes using XLMRoBERTa. Evaluation is done on the SNLI test split. "SNLI base" refers to the SNLI training set that was reduced to the respective size, without adding prototypes. "SNLI+Prototypes" is the reduced SNLI data with prototypes added that were generated according to the methods described in sections 7.4.1 and 7.4.2 . "SNLI+Prototypes+Struct." is the same data with additional structural contradictions that were obtained using the method described in section 7.4.3. All experiments were repeated five times, and mean results are reported (standard deviation in brackets). The best results per experiment are bold. We report accuracy, as well as the f1-score for the two classes ("contradiction" and "no contradiction"). All values are in percent.

dataset of structural contradictions (see section 7.5.4). Also, the fraction of SNLI data is lowest here compared to the other setups, which could cause this issue to become noticable. Nevertheless, we chose to include those results, as we can already achieve some improvement by adding this data.

7.7 Limitations

Clearly, the fact that GPT-4o was used for data generation makes it unusable in many application scenarios due to legal concerns. For a real-world use case, one would rather employ an open-source LLM such as LLAMA3 [123] or Phi [59] that can be used more freely. This work is merely a first showcase to provide evidence for our hypothesis that training on prototypical text data helps increase model efficiency, which is to be followed up with more experiments also using other models.

As stated in section 7.5, some samples display semantic and grammatical errors. One possible solution could be manual filtering, meaning the generated data could be additionally validated and possibly refined by human annotators.

From a future perspective, it might be worthwhile to know which semantic and syntactic conditions are present in both successful and unsuccessful contradiction generation. The proposed NER-based approach also represents only one way to generate structural contradictions. Simpler approaches, such as rotating entities of the same type in different positions, might be worth exploring.

7.8 Conclusion and Summary

To conclude this chapter, we presented an informed approach to generate prototypical training data for Contradiction Detection, which we evaluated by combining the synthetic data with different proportions of the SNLI dataset and training a language model on this combined data. Our results show that adding prototypical contradictions to the training helps maintaining baseline performance, while using considerably less data. This is a promising prospect with respect to the employment of small, efficient models that could replace large generative ones to some extent. Qualitative analysis shows that LLMs can comprehend the instructions and create meaningful contradictions according to specific descriptions. The advantage of this approach is that the contradictions created in this way contain the major features of the contradiction types, e.g. the aligned pair of words being antonyms. Still, a few issues with respect to data quality would need to be addressed (see section 7.7).

Future work includes extending our approach to more downstream tasks, such as (e.g.) Causality Extraction or Critical Error Detection. It should generally be possible to also apply the approach to languages other than English with little to no adaptions, given that the most powerful generative LMs are nowadays almost language-agnostic. The rule- and NER-based methods would need to be refined for other languages, but this should be relatively straightforward, given that there are well-performing approaches at least for most Indo-European languages. There are also several industry use cases, e.g. from the financial or legal domain, that might benefit from such an approach, especially since privacy and computing restrictions are usually harsh in these contexts. Working closely together with domain experts, it should be possible to derive custom prototypical training sets that represent the intricate features of those applications.

In the following chapter, we will delve into real-world applications for the theoretical approaches we have discussed so far.

CHAPTER 8

Applications - Low-Resource Strategies in Specialized Domains

In this chapter, we will explore three NLU applications with real-world relevance from the financial domain, and apply different methods for informed LM training that have been introduced in the previous chapters. Our goal is to show that those actually contribute to improving performance in low-resource, specialized application domains. Thus, we are aiming to answer the following research question:

Research Question 3 (RQ3)

How do linguistically informed methods influence language modeling and understanding performance in real-world industry use cases?

In section 8.1, we will have a look at detecting causal statements in financial news text, both using conventional token-level classification as well as generative models. The second use case is about detecting critical errors in translations of financial reports, which we tackle using encoder-based transformer models and generative AI for data augmentation, similar to the approach that was introduced in the previous chapter (section 8.2). Finally, in section 8.3, we present an application for a task that has been examined in-depth already throughout this thesis, but which will now be placed in a specific application context, namely Contradiction Detection on financial reports. Here, we apply our methods for informed pre-training, which have been introduced in chapter 4.

The relevant papers and key contributions will be discussed in the respective sections of this chapter.

8.1 Causality Extraction in Financial News Text Using Sequence Tagging and Generative AI

This section introduces the task of Causality Extraction in financial documents, both using traditional ML methods as well as generative AI with informed prompting.

The key contributions are:

- We present our solution to the 2020 Fincausal [15] challenge, consisting of an ensemble method

for detecting causality, and a sequence tagger with informed post-processing for extracting causal statements from text.

- We introduce an informed prompting approach for financial Causality Detection, leveraging our insights on linguistic intricacies of this phenomenon. This opens up many possibilities for meaningful use of generative AI in this domain.

The first part of this section was a joint work, where Maren Pielka did most of the planning and coordination. The Causality Detection approach with ensemble methods was implemented by Clayton Chapman and supervised by Anna Ladi and Maren Pielka. The Causality Extraction part was implemented by Maren Pielka and Rajkumar Ramamurthy. The paper writing was done by all beforementioned authors, while Maren Pielka had the most significant contribution. The other authors of this paper had supervisory roles. The work on the second part (Causality Extraction with generative models) was conducted only by Maren Pielka, including conceptualization, implementation and writing.

The section is based on the following publications [26, 28]:

- **Maren Pielka**, Rajkumar Ramamurthy, Anna Ladi, Eduardo Brito, Clayton Chapman, Paul Mayer, and Rafet Sifa. 2020. "Fraunhofer IAIS at FinCausal 2020, Tasks 1 & 2: Using Ensemble Methods and Sequence Tagging to Detect Causality in Financial Documents." In Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 64-68, Barcelona, Spain. Association for Computational Linguistics.

- **Maren Pielka** and Rafet Sifa. 2024. "Insights About Causalities in Financial Text - Towards an Informed Approach." In 2024 IEEE International Conference on Big Data (BigData), pages 8801-8804, Washington, USA. DOI: https://doi.org/10.1109/BigData62323.2024.10825863,

The section is split into two parts, where the first one presents our contribution to the 2020 challenge, and the second one covers the informed prompting approach. The methodologies and results for those two parts will be discussed separately, as they are dealing with different versions of the data set and follow distinct research objectives. We will conclude the section with a summary that consolidates the findings from both parts and outlines the prospects of future work.

8.1.1 Introduction and Related Work

Despite the many advances in the field of Natural Language Processing (NLP) in recent years, many challenges dealing with contextual relationships still persist, one of which being causality. Causality, as conveyed in written English, relates a textual description of a cause to a description of its effect(s). While the identification of subordinating conjunctions - such as "because", "since" and "if" - can help in identifying relevant pieces of text forming causal relationships, often richer contextual information is needed.

Being able to correctly detect causal statements is a key skill when it comes to in-depth understanding of financial text, as those statements usually are the building blocks of reasoning and argumentation. It is especially relevant for auditors who have to review financial reports in order to make sure that all

stated facts align with each other, as well as with external knowledge, e.g. from news articles. This task can be especially hard due to the intricate nature of financial language, which is why it requires specific training. On the other hand, the workload for auditors is very high, due to labor shortage as well as the recent introduction of many new rules and regulations in the context of sustainability reporting.

Given this situation, the demand for automated or partially automated solutions is high. There are a number of existing machine learning (ML) systems for auditing, such as recommender tools [72], key performance indicator extractors [124] and Contradiction Detection methods [27], which are being used in practice. At the same time, to our best knowledge, there are no existing productive solutions for causality detection in financial text.

The Fincausal shared task [15–17] was designed to direct more attention towards this topic, and inspire researchers to come up with creative solutions. Its goal is to explore causal relationships from a financial and economic context, and to determine the likelihood of whether or not a given text contains a causal relationship. While there have been a number of interesting approaches, none of those is actually being used in a real-world application setting yet - this being due, presumably, at least in part to the fact that all approaches achieved a rather mediocre performance on the provided test sets.

With the recent advance of generative pre-trained transformers (GPT) [4, 6, 120], the task demands re-investigation. Those models show promising capabilities in the context of complex text generation tasks, and can be used to tackle all kinds of problems from the Natural Language Processing domain. Nevertheless, they sometimes tend to produce incorrect or misleading output, and are very sensitive to the correct prompting. Some techniques have been developed to alleviate these problems and maximize the quality of the output, such as few-shot and chain-of-thought prompting. Based on these ideas, we want to investigate whether providing GPT models with additional knowledge about the nature of financial causality can help guiding them to find the correct cause and effect statements in the text.

In this chapter, two different approaches are being explored. The first one focuses on solving the causality detection and extraction tasks from the Fincausal challenge in 2020 [15], using ensemble methods and sequence tagging. For the second part, we collect our own data set out of the Fincausal training and practice splits from different years, followed by a linguistic analysis and an informed prompting approach.

8.1.2 Part 1: Using Ensemble Methods and Sequence Tagging to Detect Causality in Financial Documents

In the first part of this chapter, we focus on the 2020 Fincausal shared task. To this end, we apply ensemble-based and sequence tagging methods for identifying causality, and extracting causal subsequences.

Data

The data sets are composed of 23808 paragraphs from financial news.[1] They contain paragraphs, with an average size of 214 (+/- 161) characters, each with annotations relevant to the corresponding task. For task 1, the size of the complete data is 22058 paragraphs, each of which is annotated with a "1" (a

[1] The data set both for task 1 and task 2 is split into a "trial" and "practice" test. To account for any differences in the distribution of the data in the two subsets, we decided to join the two sub-datasets.

causal relationship exists in this paragraph - 7.2% of the paragraphs), or "0" (no causal relationship in the paragraph - 92.8% of the paragraphs). The data set for task 2 has a size of 1750 paragraphs, with every paragraph containing exactly one cause and one effect character sequence. These are annotated as character ranges. Since we treat task 2 as a sequence tagging task, we transform these labelss to token level annotations, so that every token in the paragraph gets the label CAUSE, EFFECT or, 0 (for all tokens which belong neither to the cause nor the effect part). This yields a label distribution of 40.8% CAUSE, 40.8% EFFECT, and 18.4% "0". For the experiments and the model evaluation for task 1 and task 2, a 70% training - 30% validation split of the respective data-set (joined practice and trial) was used.

System

Tasks 1 and 2 were treated independently. The former was modeled as a document classification task (a document being a paragraph in this case) and the latter as a sequence tagging task.

Task 1: Causality Detection The goal of task 1 is to identify whether a paragraph contains a causal relation or not. The labels are binary, with a 1 indicating the presence of (a) causal statement(s), and a 0 otherwise. For the best performing models, no data pre-processing was used, other than ignoring punctuation and single-character tokens. We explore two approaches: The first is a paragraph embedding paired with shallow Machine Learning (ML) models, while the second is embedding the individual tokens and using a one-dimensional Convolutional Neural Network (CNN) as a classifier. For the first strategy, the text data is transformed into a feature matrix using the scikit-learn [125], version 0.23.1 implementation of TF-IDF. This matrix is then given as input to by several classical ML models, including SVMs, Logistic Regression and Random Forests. The best performing models were found to be an SVM Classifier [2] and an XGBoost model. XGBoost, or eXtreme Gradient Boosting, is a fairly recent algorithm based on gradient boosting techniques [126]. Additionally, a voting classifier was constructed using these two models, and programmed to predict the class label based on the argmax of the sums of the prediction confidence values from the SVM classifier and the XGBoost models. We adjust the parameters of our models using the RandomizedSearchCV class from scikit-learn[3]. The models were trained on an Intel i7-8750H CPU. After training, a Voting Ensemble classifier from scikit-learn was built using the trained models as parameters. This ensemble used "soft" voting, that is, the probabilities output by the SGD and XGBoost were averaged and the result was the output of the ensemble.

The second strategy is based on the idea of using CNNs for NLP [127]. Adjusting the filter size on a one-dimensional CNN also acts similarly to an N-gram, and CNNs are generally very fast at analyzing problems with large feature matrices. This model was implemented using Tensorflow [128] and Keras [129], versions 2.1.0 and 2.3.1 respectively. The data was processed using the Keras tokenizer and then fed into a one-dimensional CNN. The network consists of an embedding layer that uses the 200-dimensional Word2Vec [108] embedding from GoogleNews, a convolutional layer, and 2 dense

[2] For the SVM classifier, the scikit-learn implementation was used.

[3] For the TFIDF-SVM the adjusted parameters are: ngram_range = $(1, 2)$ and smooth_idf = False for TFIDF, and alpha = 1^{-6}, loss = log for SVM. For TFIDF-XGBoost: ngram_range = $(1, 1)$ for TFIDF, and booster = gbtree, learning_rate = 0.3, max_depth = 6, and n_estimators = 100 for XGBoost.

layers. The models were optimized using manual parameter tuning[4]. This model was trained using an Nvidia GTX 1060 Max-Q with 6GB of GPU memory.

Task 2: Causality Extraction The goal of task 2 is to identify the parts of a sentence, that correspond to cause and effect, respectively. It is thus a sequence tagging task in which the tokens of a sentence must be assigned either a CAUSE, EFFECT, or 0 tag. One approach to tackle sequence tagging tasks is the use of sequential models such as a Recurrent Neural Network (RNN). We employ the Flair sequence tagger [130], using ElMo [39] and fine-tuned BERT embeddings. For BERT, the transformers library by [131], version 3.0.2 is used.

We utilize the Flair Sequence tagger [130] to produce token-level predictions for the causality extraction task. The framework consists of a recurrent neural model with a Conditional Random Field (CRF) and a Long Short Term Memory (LSTM) layer trained for token classification. Learning rate scheduling is applied during training, meaning that the learning rate will be reduced whenever the validation loss does not decrease after 3 epochs. Once the learning rate falls below 0.0001 following this policy, training is stopped.

As a first fine-tuning step, we evaluate different pre-trained word embeddings (ElMo, BERT, Flair, GloVe, and FastText) that are integrated in the Flair framework. We find that ElMo embeddings obtained from the full-sized model (see [39] for implementation details) yield the best results on our data, so we use them as word embeddings in the following evaluation steps. The model is being further improved by optimizing the hyperparameters, yielding a batch size of 32, an initial learning rate of 0.1, and a hidden size of 500 neurons. In addition, we adjust the class weights of the model to compensate for the imbalanced label distribution (see section 8.1.2). Thus, the weights for the CAUSE and EFFECT classes are decreased to 0.25, and the weight for the 0 class is increased to 0.5. As an alternative embedding method, we also incorporate the fine-tuned BERT model which was provided as a baseline for task 1 by the challenge organizers[5] to the Flair Sequence Tagger. The models were trained on an Nvidia Tesla V100-SXM2 with 32 GB of GPU memory.

In addition to the token classification by the Flair sequence tagger, we apply some post-processing to the output in order to further improve the results. Our first approach is based on the observation, that the classifier sometimes correctly recognizes large parts of a CAUSE or EFFECT sequence, but still predicts 0 for some single tokens in between. Since in most of the cases, every CAUSE or EFFECT is a coherent sequence, it makes sense to account for that using rule-based post-processing. This is done by filling in every "hole" of up to three tokens in a consequent sequence of CAUSE or EFFECT predictions with the surrounding label. An alternative, less strict post-processing approach that was tested, included smoothing the output class probabilities over consecutive tokens, using an average filter with a window size of 3 tokens.

[4] The best configuration was: a tokenizer that ignored all punctuation and symbols, and set all words to lowercase, and for the CNN: a 1D convolutional layer of 64x3 with relu, a dropout layer with $rate = 0.1$, a global 1D maxpooling layer, a dense layer with an output of 16 with relu, another dropout layer with $rate = 0.1$, and a final dense layer with an output of 1 with sigmoid activation.

[5] https://github.com/yseop/YseopLab/tree/develop/FNP_2020_FinCausal/baseline/task1

Model	F1 score	Recall	Precision
XGBoost	0.942374	0.948957	0.943619
SVM classifier	**0.942909**	0.947604	0.942031
Ensemble	0.937722	**0.949093**	**0.950175**
CNN	0.942066	0.945979	0.940644

Table 8.1: Results for task 1 (Causality Detection).

Results

The reported results were evaluated on the holdout test set, which was not included in the validation set. The evaluation metrics are reported as defined by the shared task organizers. [6]

Task 1 For task 1, the best results in terms of F1 score were achieved by the SVM classifier, while the ensemble of the SVM classifier and the XGBoost classifier performed best in terms of precision and recall (see Table 8.1). The CNN model performed not significantly different than the less complex ML models.

Task 2 For task 2, the best scores with respect to the three common metrics are achieved by the Flair sequence tagger model using fine-tuned BERT embeddings, balanced class weights, and applying the first post-processing approach (filling in "holes" of predicted sequences with the surrounding label, see section 8.1.2). An overview of the results is displayed in table 8.3. Interestingly, our baseline model (ElMo) outperforms the fine-tuned models on ExactMatch. Especially adding balanced class weights seems to cause a drop in the ExactMatch metric. This could potentially indicate some overfitting effect in the simpler model. Table 8.2 shows two examples that illustrate this behavior. In example 1 (Table 8.2(a)), while the simpler model predicts the sequence exactly right, the model with fine-tuning added makes one mistake in between. Example 2 (Table 8.2(b)), however, shows that the second model is better at separating cause and effect in a non-straightforward formulated sentence, even though it does not get the labels exactly right. Generally, the fine-tuned model tends to leave larger "holes" between "cause" and "effect" sequences, than the simpler model does.

Discussion

Our present system relies heavily on the choice of embeddings, as well as on large pre-trained language models. This could possibly be changed if we had a more extensive and consistent data set for training. One of our observations in this regard was, that the provided data included a number of irregularly formatted paragraphs (for example headlines or bullet points). We believe that our algorithms would benefit from further analysis and possible cleaning of such instances. Additionally, the current data does not allow us to explore other approaches that are based on syntactic features.

Regarding task 1, the results of the CNN could be potentially improved by additional hyperparameter tuning. The CNN model showed good results already in early epochs, but was not able to outperform

[6] See website of shared task: https://competitions.codalab.org/competitions/23748

Sentence	They	fell	(...)	to	4p	on	Wednesday	as	analysts	lowered	price	targets	and	cut	forecasts.
Model 1 prediction	E	E	(...)	E	E	E	E	0	C	C	C	C	C	C	C
Model 2 prediction	E	E	(...)	E	E	E	0	0	C	C	C	C	C	C	C
True label	E	E	(...)	E	E	E	E	0	C	C	C	C	C	C	C

(a) Example 1

Sentence	Suppose	(...)	they	could	borrow	at	-0.25%	NLY	would	be	getting	paid	$262	million.
Model 1 prediction	E	(...)	E	E	E	E	E	E	E	E	E	E	E	E
Model 2 prediction	C	(...)	0	C	C	C	E	E	E	E	E	E	E	E
True label	C	(...)	C	C	C	C	0	E	E	E	E	E	E	E

(b) Example 2

Table 8.2: Example predictions of model 1 (ElMo) and model 2 (fine-tuned BERT, balanced, post-processing), on two sentences from our validation split (part of the practice data set) for task 2 (Causality Extraction). "C" stands for "cause", "E" for "effect". Due to space constraints, only the relevant part of the sentence is displayed.

Model	F1 score	Recall	Precision	ExactMatch
ElMo	0.740380	0.745353	0.741605	**0.231975**
ElMo, balanced	0.726232	0.71307	0.777991	0.184953
ElMo, balanced, post-processing	0.743371	0.734344	0.769642	0.210031
ElMo, balanced, post-processing (probability smoothing)	0.741773	0.734672	0.759757	0.152038
ElMo, task1 data augmentation	0.731066	0.733866	0.729065	0.188088
ElMo, task1 data augmentation , post-processing	0.737836	0.748546	0.735892	0.191223
fine-tuned BERT, balanced, post-processing	**0.759982**	**0.748933**	**0.799503**	0.191223

Table 8.3: Results for task 2 (Causality Extraction). F1-Score, recall and precision refer to the micro averaged scores over the “cause” and “effect” classes, as defined by the challenge organizers. ExactMatch is a custom metric that quantifies the fraction of samples where the annotated cause and effect sequences were matched exactly by the model predictions. The Flair sequence tagger model was used to produce all of those results.

the less complex ML models, which were extensively tuned. With respect to task 2, it would have been interesting to test more combinations of embeddings, data augmentation, and post-processing, which was not possible due to limited time and computing resources.

8.1.3 Part 2: Insights About Causality Extraction in Financial Text - Towards an Informed Approach

In the second part of this chapter, we perform a linguistic investigation of causality in financial reports, and apply our findings by prompting GPT-4o with the acquired knowledge to improve its predictive capabilities.

Methodology

To tackle the problem of Causality Extraction in financial text, we first assemble a data set out of the existing training and practice sets from Fincausal. We perform a qualitative analysis of this data and derive some findings about the linguistic nature of financial causalities. Based on those findings, we

devise an informed prompt that helps the LLM pay attention to specific aspects during classification.

Data Collection Our data set consists of the training and practice splits[7] from the Fincausal challenges in 2020, 2022 and 2023. Those are all originally from financial news websites, one sample corresponding to a paragraph from an article. For more detail on the initial data collection, please refer to [15–17].

We find that there is some overlap especially between the data sets from 2022 and 2023, as well as a few duplicates, which we remove. Our resulting data set contains 3422 examples, out of which 742 are complex samples, meaning that there is more than one possible cause and/or effect statement annotated. We perform a random training-development-test split on this data. Only the test set, containing 685 samples, is used for the evaluation in this part of the chapter.

Intricacies of Financial Causality We investigate the training data split manually, in order to gain a better understanding of the linguistic nature of causality in financial documents. To this end, around 100 samples are inspected and analyzed in detail. Generally, it is not trivial to clearly make out the cause and effect parts in each sentence, especially as a financial non-expert. Some of the annotations are very subtle, and the cause and effect parts can also be ambiguous. We still identify some interesting phenomena that could be relevant for better understanding those statements.

One frequently observed pattern is the cause and effect being implicitly defined by a temporal relationship. Consider this example:

> **Statement**: Post 2008 financial crisis, lending policies were tightened with interest-only mortgages no longer able to be insured, a reduction of maximum insurable amortization terms from 40 to 35 years, and the minimum down payment raised from 0% to 5%.
>
> **Cause:** Post 2008 financial crisis
>
> **Effect**: lending policies were tightened with interest-only mortgages no longer able to be insured, a reduction of maximum insurable amortization terms from 40 to 35 years, and the minimum down payment raised from 0% to 5%.

It is not immediately clear only from the text of this paragraph, that the financial crisis is in fact the cause of all subsequently described events. Some degree of world knowledge and insight is necessary to draw this conclusion.

A similar case can be observed with the following sample:

> **Statement**: Things got worse when the Wall came down. GDP fell 20% between 1988 and 1993. There were suddenly hundreds of thousands of unemployed in a country that, under Communism, had had full employment.
>
> **Cause:** Things got worse when the Wall came down.
>
> **Effect**: GDP fell 20% between 1988 and 1993.

Here we also have a temporal relationship that does not per se carry a causation. Moreover, the cause could be seen as a high-level description of the effect. Only with some background knowledge it

[7] We cannot use the test splits here, because labels are not provided.

is possible to see the causal connection between the collapse of the Berlin Wall and subsequent GDP drop in East Germany. Also, correctly guessing what historic events are referred here requires filling in missing information from the text with known facts.

> **Statement**: Transatlantic passenger traffic - key in the late-spring and summer months when sun-destination trips drop off - rose more than four per cent year over year last quarter to help boost revenue to $698.9 million, up from $664.6 million a year earlier.
>
> **Cause:** help boost revenue to $698.9 million, up from $664.6 million a year earlier.
>
> **Effect**: Transatlantic passenger traffic - key in the late-spring and summer months when sun-destination trips drop off - rose more than four per cent year over year last quarter

In the above example, the cause is also not easily identifyable by a non-expert. One could even think that cause and effect might be reversed here (since the increase in traffic led to the rise in revenue). The reporter probably takes a retrospective point of view, seeing the cause of traffic increase being the goal that was set by the airline the year before.

Those exemplary statements show, that detecting causes and effects in financial text is not at all trivial and in some cases very subjective. Also, a high amount of world knowledge, as well as logical reasoning is required, rendering this a particularly hard problem for a machine learning model.

Prompting GPT-4 We utilize our insights from the previous section to derive informed instructions for prompting an LLM. Firstly, we design a very simple system prompt that only contains some basic background information, as a baseline:

> **System:** You are an expert on finance and linguistics, with a profound knowledge in financial document analysis. You are especially trained in detecting causal statements in financial text.

Building upon this, we add four key findings from our analysis of the training data to the basic prompt, resulting in this more sophisticated version:

> **System:** You are an expert on finance and linguistics, with a profound knowledge in financial document analysis. You are especially trained in detecting causal statements in financial text. Please also consider the following:
>
> 1. Causes and effects in financial text might be very subtle and subject to ambiguity.
> 2. Causes could also be formulated implictly, e.g. as a temporal relationship ('... after X ...').
> 3. The cause may be a different or more high-level description of the effect.
> 4. Both cause and effect are assumed to be contiguous parts of the input paragraph, respectively.

Two versions of this prompt are used: One containing only the above text, another with few-shot examples from the training set added after points (2) and (3).

Finally, we devise a user prompt which is given to the model together with the respective data sample (represented by the placeholder <SAMPLE> here):

User: Please determine, given the following statement, the parts in the text that refer to cause and effect. Only return those exact statements and nothing else. If there are multiple possible cause-effect tuples, only return one of them. Answer in the following JSON format: {{'cause': <Cause>, 'effect': <Effect>}}
Sentence: <SAMPLE>

The user prompt is the same for both approaches. We use GPT-4o and GPT-4o-mini for our experiments, as those are the most performant models to date.

Results

We instruct GPT-4o and GPT-4o-mini with the prompts defined above, and evaluate the results. We enforce a JSON formatted output by setting the "response_format" parameter of the OpenAI API accordingly. The task is treated as a 3-class token-level classification problem, using the classes "cause" (for tokens that are part of the cause statement), "effect" (for tokens that are part of the effect statement) and "0" (for tokens that are part of neither cause nor effect). For simplicity, tokenization based on whitespaces is used. Complex samples (more than one possible cause and/or effect statement) are evaluated as follows: We compare the output of the model with all cause and effect annotations, and choose the one with highest similarity (based on the difflib implementation[8] of Gestalt pattern matching [132] on token basis) as ground truth. In a few cases, the output did not adhere to the defined json schema; we treat those cases as if the model predicted only "0" for all tokens in the respective samples. Same goes for examples where the model produced text that did not exactly match the original sample.

Model	Recall	Precision	F1-Score	ExactMatch
GPT-4o-mini (simple prompt)	0.518	0.657	0.524	0.117
GPT-4o (simple prompt)	0.523	0.658	0.530	0.105
GPT-4o-mini (informed prompt)	0.557	0.671	0.566	0.142
GPT-4o (informed prompt)	0.551	0.665	0.559	0.121
GPT-4o-mini (few-shot prompt)	**0.615**	**0.690**	**0.623**	**0.237**
GPT-4o (few-shot prompt)	0.598	0.684	0.606	0.191

Table 8.4: Model performance on our custom test set. We report recall, precision and F1-score averaged over all three classes (0, cause, effect) and weighted by the number of samples in each class. Best performances are bold.

Quantitative results are displayed in table 8.4. We can see that while both models benefit significantly from using the informed prompts, the effect is especially evident for the more light-weight GPT-4o-mini. It even outperforms GPT-4o with all three configurations. This could hint to the fact that the sheer number of parameters does not necessarily determine the reasoning capabilities of a model, and that smaller models can benefit from an informed approach even more.

It is also worth noting, that while using the plain informed prompt already improves the models' prediction capabilities, the most significant performance boost comes from adding the few-shot examples. Again, this effect is even more evident for GPT-4o-mini. This provides some valuable

[8] https://docs.python.org/3/library/difflib.html

insight, namely that LLMs are especially good at learning from examples, compared to learning from abstract knowledge formulations.

8.1.4 Conclusion and Summary

In this section, we presented different methods for identifying and extracting causal information in financial text. Our findings suggest that while causality is among the hardest problems to solve for a language model, providing the LM with information about the nature of causalities helps improving its performance considerably. We can see this effect both in part 1 of this section, where we showed that intelligent post-processing helps boosting performance, and in part 2, where especially the smaller GPT-4o-mini model benefits from an informed prompting approach.

Future work should be focused on further refining the approaches, i.e. by adding more few-shot examples to the prompt. Another idea would be the generation of prototypical training data, similar to the work presented in chapter 7 and by [10]. In order to accomplish this, a typology of causality types could be derived from the available data, and an LLM would be instructed to create new data points according to the type descriptions. The resulting data set could be used to train considerably smaller, encoder-based models for this task, alleviating the need for generative AI during inference.

On a more general note, identifying causality in text can be a helpful tool for many empirical use cases. In the context of financial document analysis, it could be used to identify important facts and developments in the annual report of a company. We can potentially extend our work on various downstream-tasks by incorporating causality, for instance for Key Performance Indicator Extraction [133], Contradiction Detection [27], or content-based text classification and consistency checks [72]. It is also possible to exploit Causality Extraction in the context of text summarization. Causal sentences may indicate content richness, which is useful not only to extract the most relevant sentences of an original text within an extractive summarization setting [134], but also when we evaluate the quality of a generated summary from a wide range of features [135]. We are planning to address such applications, building on our experiments and insights from this work.

The ultimate goal would be to devise an automated system that helps auditors and financial professionals to detect causal statements in reports and news articles, in order to unravel reasoning chains and possibly detect errors or irregularities. The solution could be deployed in a recommender-like setup, by giving the auditors suggestions for where to look more closely in a text. The professionals could then accept or decline the suggestions, thereby providing training data points for refining the system further.

8.2 Automating Translation Checks of Financial Documents Using Large Language Models

In this section we present a solution for automatically detecting translation errors in financial reports, using encoder-based transformer models for classification and large generative models for data generation.

The key contributions are:

- We propose an industry-ready, end-to-end solution for the problem of identifying translation errors in financial reports, a task that is relatively under-researched compared to other text-related use cases.

- We showcase how state-of-the-art generative LMs can be used to facilitate the creation of high-quality data sets, when provided with condensed knowledge about a specific task.

The idea for this research direction was developed by Maren Pielka, based on the insights from a customer project with a similar use case. The error categories were derived by Maren Pielka and Maria Chiara Talarico. The data augmentation pipeline was designed by Maren Pielka and Tobias Deußer, and implemented by Cong Zhao. Daniel Uedelhoven implemented the sentence matching algorithm, and Max Hahnbück conducted the model training and evaluation. The paper was written by Maren Pielka (Introduction, Related Work, conceptual part and Conclusion) and Max Hahnbück (experimental part). The other authors worked on the backend and frontend implementation of the tool, and/or were involved in project management.

The section is based on the following publication [29]:

- **Maren Pielka**, Max Hahnbück, Tobias Deußer, Daniel Uedelhoven, Moinam Chatterjee, Vijul Shah, Osama Soliman, Jannis von der Bank, Writwick Das, Maria Chiara Talarico, Cong Zhao, Carolina Held Celis, Christian Temath, and Rafet Sifa. 2025. "Automating Translation Checks of Financial Documents Using Large Language Models." In Language Resources and Evaluation (2025). Springer. DOI: https://doi.org/10.1007/s10579-025-09862-z

In the following, we will first give an introduction to the problem and explain our data augmentation approach, as well as the pipeline of our translation check tool. Then we will discuss experiments and results, and draw a conclusion with respect to the overall goals of this chapter.

8.2.1 Introduction

Assessing the correctness of manual translations is a task that regularly occurs in the context of financial reporting. Many multinational companies face the challenge of having to compile their annual financial statements in multiple languages. This process is very delicate and prone to errors, as all terms and formulations have to be translated correctly, and small deviations can amount to grave differences in meaning. For this reason, the reports have to be checked for mistakes thoroughly after they have been translated.

We propose a machine learning based solution to automatically identify translation mistakes in German-English financial reports. To this end, we use a heuristic matching algorithm to find pairs of corresponding sentences in both languages, followed by a deep learning approach to identify any deviations in the translated text. Additionally, we leverage the power of large generative language models (LMs) such as GPT-4o to generate training data. This approach alleviates the need to collect large amounts of annotations from domain experts, and allows us to transform knowledge about the main characteristics of common mistakes directly into a data representation.

We are making our solution publicly available on a German AI platform[9], where users will be able to upload their own reports for testing the functionality.

8.2.2 Related Work

While machine translation is a well-researched and developed topic, there is limited previous work on the subject of automated translation checks. In previous years, the WMT conference hosted a

[9] https://translation-check.ki.nrw/

shared task on Machine Translation Quality Estimation (MTQE, or just QE) and specifically Critical Error Detection (CED)[10] [14, 136], which is about identifying critical errors in machine translations. We found this to be a good approximation of our industry use case, although it is not exactly the same problem formulation (i.e., we are especially interested in errors that would occur in financial text, and thus are not necessarily to be considered critical in a classical sense). The top performing participants in the 2022 challenge used multi-task, multilingual large pre-trained transformer models such as XLM-RoBERTa [22]. Still, the results especially in the constrained CED setup, where no external training data was to be used, were far from optimal, underlining that it is still a hard and under-researched task. Integrating explainability to QE systems also seems to be a promising research direction in order to improve performance and user acceptance. An interesting approach in this regard was introduced by [62], who employed linguistically informed data augmentation to produce synthetic training examples for CED, thereby achieving competitive results on the shared task while using only a very small amount of data compared to the other participants.

The transformer architecture, which was first introduced by [3], is the basis for most text mining methods used today. Encoder-only transformer models such as BERT [41], DistilBERT [137] and XLM-RoBERTa [22] are the state of the art for many text classification problems. Even though large generative language models such as GPT-4o [6], LLama3 [123] and Mixtral[11] achieve impressing zero-shot results on those tasks, it is still favorable in many application scenarios to use a much smaller, task-specific model.

In the context of financial document analysis, there are a number of related use cases w.r.t. our application. For instance, [72] introduce an AI-based system for automated auditing, which enables the user to pre-filter relevant passages from a financial report according to legal criteria. Building on this framework, [138] and [124] develop a sophisticated, multi-step approach to identify numerical inconsistencies among a report, by automatically finding and matching key performance indicators in text and tables. Regarding semantic inconsistencies, [27] introduce an approach for detecting contradictions in financial text, using transformer models and linguistically informed pre-training (see also section 8.3 of this chapter). While being closely related to the previously mentioned consistency checking methods, our approach covers a novel use case in this context. To our knowledge, there is no directly comparable previous work on this specific task (translation checking/critical error detection on financial documents in German-English).

8.2.3 Methodology

We employ a multi-step approach for identifying semantic translation mistakes in financial reports. The high-level pipeline is being illustrated in figure 8.1. In the first step, the reports are processed by a custom PDF parsing solution that employs the identification of specific text classes (e.g. paragraph, header/footer, table). The segmented documents are further processed, i.e. irrelevant elements such as headers, page numbers, tables and images are filtered out. On the remaining paragraphs, we apply a heuristic algorithm to identify matching pairs of sentences. Those pairs are then given as input to an ML model that determines whether there is a potential translation mistake. For training the classifier, we use synthetic data generated by state-of-the art LLMs, a process that is further being described under 8.2.6.

[10] https://wmt-qe-task.github.io/wmt-qe-2022//subtasks/task3/

[11] https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO

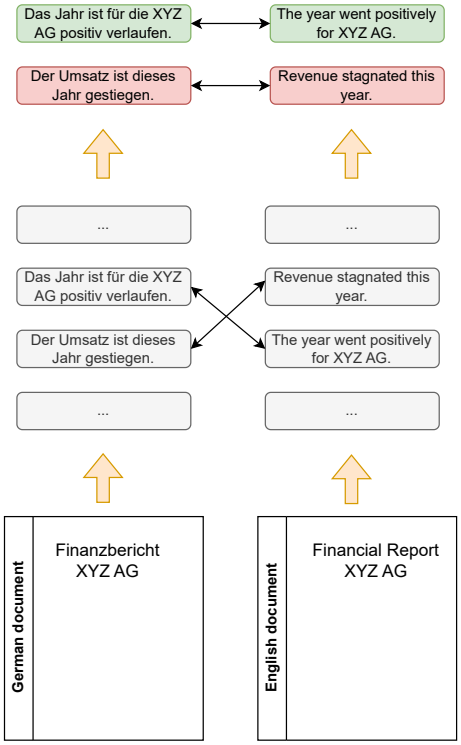


Figure 8.1: Illustration of the multi-step translation check approach. First, the financial reports are analyzed by a PDF parser and segmented into sentences. Those segments are subsequently being matched by the Gale-Church algorithm, and potential mistakes are being identified among the matched pairs using a deep learning approach.

8.2.4 Sentence Splitting and Matching

In order to identify translation mistakes, we need to find a mapping between sentences in the German and the English document.[12] To achieve this, the parsed documents are first split into sentences using the nltk toolkit [139]. We argue that it is easier to operate on the sentence level compared to paragraph level, because of the finer granularity, and in order to alleviate parsing issues. We then employ the Gale-Church algorithm [140] to find matching pairs of German and English sentences. It is a heuristic approach that applies a probabilistic method based on the differences in length of the sentence pairs, in order to find the optimal matches. It operates on the assumption that longer sentences in one language are most likely to correspond to longer sentences in the other language, and it is to some extent robust w.r.t. permutations and missing sentences. The resulting matches might not be 1:1, so a German sentence could be matched to two consecutive English sentences, or vice versa.

We find this approach to work reasonably well with the given setup and data. It is to be mentioned, that some false positives (sentence pairs being wrongly labeled mistakes) may result from incorrect matches. In our framework, this is to some degree acceptable, as our focus is on finding any potential

[12] This is under the assumption, that such a mapping exists, which might not be the case if there is additional information in one of the documents - a scenario that we explicitly exclude here.

translation mistakes, and a small amount of incorrect hits is better than missing a possible mistake. Also, the models are explicitly trained to recognize those wrongly matched sentences, by including examples into the data generation process (see 8.2.6). This approach is not part of our evaluation in section 8.2.7.

8.2.5 Critical Error Detection

Critical Error Detection (CED) is the task to identify critical errors in (machine) translations. It is a sub-task of QE and a relatively new research objective. CED was first introduced as a part of the 2021 WMT QE shared task [14]. In general, it is defined as the objective to find translation mistakes that could have a critical impact, i.e. in terms of health and safetly risks, toxicity or sentiment. Critical errors can e.g. occur in the form of additions, deletions, deviations in named entities, meaning and/or numbers. In the scope of this work, we mainly consider negations, additions and significant changes in meaning, as we find them to be especially relevant and comprehensive w.r.t. financial text, and it is also easy to artificially produce those without deeper knowledge on the subject of the text.

8.2.6 Data Generation and Augmentation

There is little public data available for the CED task, and no existing data sets for this problem in the financial context. Therefore, we train and evaluate our model on a custom created data set. The data set originates from two publicly available financial reports. Each of the reports has recently been published both in German and English. Half of the English sentences is translated to German, once correctly and once with adding different types of mistakes. In order to get reasonable results, we first perform a linguistic analysis of similar reports, with the goal to obtain an understanding of probable translation mistakes that could happen when processing those reports. Having done this, we define three types of mistakes that are being applied while translating:

1. Negation: Adding words such as "not" or "no" to negate the meaning of the sentence,
2. Addition: Adding more information not included in the original sentence,
3. General Error: Significantly change the meaning of the sentence.

The same is done to the other German half. Using this method, 5000 correctly translated and 5000 incorrectly translated sample are created. For this task we use GPT-3.5-Turbo by OpenAI [6]. The reason for using this model variant, compared to the more recent and potent GPT-4o, is that we wanted to keep the costs for generating the data relatively low, and we found that GPT-3.5 is already capable of producing high quality data.

Additionally German and English sentences from these reports are randomly matched. Also for both languages a few sentences are paired with an empty string. For each of these methods 1000 samples are created. The first method is done so the model learns to detect if the provided sentences are too different from each other and therefore can not be a correct translation. The latter one has a similar goal: to catch any sentences where there is an empty sentence matched which has not be detected by other means before, thereby improving robustness of the approach. This results in a slightly imbalanced class distribution of 7000 positive (incorrectly translated) samples to 5000 negative (correctly translated) samples. An overview on the resulting data distribution can be found

Data generation method	Num. samples
Correctly translated from German to English (GPT)	2500
Correctly translated from English to German (GPT)	2500
Incorrectly translated from German to English (GPT)	2500
Incorrectly translated from English to German (GPT)	2500
Randomly matching English and German Sentences	1000
Matching sentences with empty string	1000
Total	12000

Table 8.5: Overview on the custom data set created using GPT-3.5 and random matching of sentences.

in table 8.5. Some exemplary sentences from the data set are displayed in table 8.6. The data set is divided into training, validation, and test sets in proportions of 80%, 10%, and 10%, respectively. The data set is being made publicly available via Huggingface[13].

8.2.7 Experiments and Results

In the following sections, we define our evaluation metrics, discuss the overall training setup and evaluate the results.

Model Training Firstly we use a pretrained BERT-based model to encode both sentences. We try a total of four different models of varying sizes: DistilBERT-base-multilingual [137], BERT-multilingual-base [41], XLM-RoBERTa (large) [22] and Multilingual E5 [141]. We apply tokenization to transform an input sequence into a sequence of sub-word tokens. The sentences are separated using the [SEP]-token(s) of the respective tokenizer. Passing the tokenized sequence to the pretrained model yields a sequence of contextual token embeddings starting with a [CLS] token embedding and representing the aggregated context hidden state for the whole input. Subsequently, we use a multi-layer perceptron (MLP) to classify the CLS-Token representation. Alternatively to the CLS-Token we take the pooler output of the model for classification, if the model provides such an output. The MLP consists of fully-connected hidden layers (the number of layers is being included in the hyperparameter search, see table 8.7), ReLU activation functions and a sigmoid function at the output layer. During training, we jointly fine-tune the parameters of the BERT model and the classifying MLP to minimize the Binary Cross Entropy (BCE) loss between target labels and predicted values.

In order to estimate the best hyperparameters for training the model, we conduct a grid search comparing various parameter combinations based on their validation set performance to determine the best training setup (see table 8.7). We do this for a series of different embedding models to find the embedding that is suited best for this task.

Because the problem is a binary classification task and we have a slightly imbalanced data set, we keep track of the Binary-Accuracy and F1-Score in addition to the Binary Cross Entropy-Loss. We determine the optimal model for each BERT architecture by assessing its validation accuracy throughout the training process.

[13] https://huggingface.co/datasets/MaxHahnbueck/translation_check_synth

English sentence	German sentence	Mistake	Generation method
In the process, the Supervisory Board engaged an independent external consultant.	Im Verlauf hat der Aufsichtsrat einen unabhängigen externen Berater hinzugezogen.	No	Correct translation
Actual defaults result in derecognition of the receivables affected.	Tatsächliche Ausfälle führen zur Herausnahme der betroffenen Forderungen aus der Bilanz, was beispielsweise zu deutlichen Veränderungen in der Finanzberichterstattung und potenziellen finanziellen Verlusten für das Unternehmen führen kann.	Yes	Wrong translation (addition)
Added to this were not a lower sales share in the modernization business and higher R&D expenses to strengthen our technological position in the battery coating sector.	Hinzu kamen ein geringerer Umsatzanteil im Modernisierungsgeschäft sowie höhere F&E Ausgaben zur Stärkung unserer Technologieposition im Bereich Batteriebeschichtung.	Yes	Wrong translation (negation)
According to forecasts by the experts at UBS Bank in June 2022, the copper price is expected to average US$ 8,818/t in calendar year 2023.	Laut der Experten der UBS Bank wird der Preis für Schokolade im Juni 2022 voraussichtlich durchschnittlich 8,818 US-Dollar pro Tonne im Kalenderjahr 2023 betragen.	Yes	Wrong translation (general error)
Investment property comprises both property owned by the Dürr Group as well as property that is sublet under operating leases.	Heute sind wir in unserer Branche Vorreiter der grünen Transformation und treiben diese aktiv voran.	Yes	Incorrect match
Buoyed by strong new orders and a significant increase in the forecast at the end of July, the share staged a rally, which propelled it to a high for the year of €44.08 in August.		Yes	Empty match

Table 8.6: Exemplary sentences from the synthetic data set, generated using GPT-3.5.

Hyperparameter	Tested configurations
Batch Size	8, 16, 32, 64
Learning Rate	1e-5, 5e-6, 1e-6
Number Hidden Layers	1, 2
Optimizer	Adam
Activation Function	ReLU

Table 8.7: Evaluated hyperparameter configurations

	Size	Accuracy	F1-Score	Recall
GPT 3.5 Turbo	n.a.	0.812	0.813	0.689
GPT 4o	n.a.	0.924	0.938	**0.984**
Llama 3	70B	0.931	0.940	0.928
Mixtral 8x7B	46.7B	0.843	0.855	0.792
DistilBERT base multilingual	135M	0.862	0.867	0.838
BERT multilingual base	179M	0.898	0.908	0.865
XLM-RoBERTa	561M	0.956	0.963	0.971
Multilingual E5	560M	**0.963**	**0.967**	0.965

Table 8.8: Test set performance of each model. We report the F1- and recall scores w.r.t. the "mistake" class. Best performances are bold.

For all training runs we use a random seed of 42, set the maximum number of epochs to 20 and employ early stopping with a patience of 3 epochs.

Baseline We compare the models from section 8.2.7 with four generative models: Llama 3 70b[14] [123], Mixtral 8x7B[15], GPT 3.5 Turbo and GPT 4o [6]. These models form a strong baseline because due to their large number of parameters and extensive task-agnostic pre-training, they have decent reasoning capabilities and should therefore be able to classify the sentence pairs reasonably well without any additional fine-tuning. The prompt has been tuned by trying out a selection of prompts, including one with few-shot learning. They all perform similarly on the test data set. All prompts that were used can be found in the appendix (section A.4).

Results We report accuracy, F1-score and recall (the latter two metrics both considering the "mistake" class as positive). We argue that in an application scenario, one would be most concerned about recall and F1-score, as a well-performing system would primarily be required to identify all possible mistakes (ideally: recall = 1), while not making too many false positive predictions (quantified by the F1-score). For this reason, we pay most attention to those two metrics when evaluating our models.

Table 8.8 demonstrates that our custom-trained models consistently outperform many prominent

[14] https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
[15] https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO

generative models. Even the smallest and least effective model in our lineup surpasses certain generative models like GPT-3.5 and Mixtral. Additionally, the results indicate that larger encoder-based models deliver superior performance, with models such as XLM-RoBERTa-large and Multilingual-E5-large outperforming their smaller counterparts by 6-7 percentage points. The top-performing model, Multilingual E5, uses the following hyperparameters: a batch size of 64, a learning rate of 5e-6, and an MLP with a single hidden layer of 100 neurons. Despite its strong overall performance, it slightly lags in recall compared to GPT-4 and the XLM-RoBERTa model, both of which exhibit particularly high recall scores relative to their F1 scores. Nevertheless, the Multilingual E5 model still demonstrates robust recall performance.

A noteworthy observation is the underperformance of the GPT model used for generating training pairs compared to the smaller, custom-trained BERT models. This discrepancy can likely be attributed to the fact that the general-purpose chatbot models, such as GPT, are not fine-tuned for classification tasks, whereas our models are specifically trained for this problem. The task of generating translation errors or subtly altering correct translations is thus simpler for an LLM than identifying those errors, which may explain this performance gap.

Our findings suggest that small, custom-trained models not only provide competitive performance but also offer a more cost-effective and computationally efficient solution compared to large-scale LLMs for this specific classification task. While generative AI could replace certain parts of the inference pipeline, our results indicate that doing so would not be advantageous in this context, especially when we are in a low-resource scenario with high data security demands (which is the case when dealing with unpublished financial reports).

8.2.8 Conclusion and Summary

We presented a multi-step approach for automatically checking translations of financial reports in German-English. Our approach uses a pipeline of document parsing, sentence segmentation, sentence matching and automated translation checking. For the last step, we train and evaluate an encoder transformer model using a synthetic data set. The data set was created based on a semantic analysis of typical translation mistakes, and using the generative capabilities of GPT-3.5. We could show that while smaller, task-specific models can perform en par with more powerful ones for this specific task, large generative LMs are effective data generators and can replace expert human annotators to some extent. This suggests that the use of such systems to produce huge amounts of data will increase in the future, while humans would take on the task of providing insights and expertise in a more condensed and time-efficient way (e.g. via prompt engineering).

Due to the lack of publicly available domain-specific data sets, it was not possible for us to quantitatively evaluate our solution in a real-world scenario. Nevertheless, our qualitative evaluation shows that the synthetic data set is of overall good quality and represents the use case reasonably well. Future work would encompass compiling an expert-annotated test set for this use case, in order to facilitate model development and benchmarking. This could in parts be realized e.g. by having expert auditors use our online tool and correct or confirm the mistakes identified by the model.

We plan to extend the approach to other languages and use cases, e.g. to a medical or legal context. This would probably not require drastic changes to the pipeline, as it is agnostic to the data and setup used. However, having a multilingual pipeline would imply some modification of our approach. Extending the taxonomy of translation errors could possibly improve the output quality. Another idea would be to further classify the mistakes that have been found (i.e. into the pre-defined categories),

and provide some suggestions for correcting them. Lastly, we plan to improve our models based on feedback from users of our tool.

8.3 Contradiction Detection in Financial Reports

In the final part of this application chapter, we revisit the problem of Contradiction Detection, and examine it in the context of financial auditing.

The key contributions are:

- We collect a custom data set for the task of Contradiction Detection in financial reports, which is being annotated and reviewed by expert auditors..
- We present a comprehensive evaluation of different transformer-based aproaches for this task, and find that linguistically informed pre-training (as introduced in chapter 4) significantly boosts the results.

This research was mainly conducted by Maren Pielka and Tobias Deußer, with equal contribution from both parts. The data collection was done with the help of professional auditors from PwC, as well as an accounting student, which are listed as co-authors of the corresponding paper. The pipeline for pre-training and fine-tuning, which had initially been developed by Lisa Pucknat for another work, was re-used here.

The section is based on the following publication [27]:

- Tobias Deußer, **Maren Pielka**, Lisa Pucknat, Basil Jacob, Tim Dilmaghani, Mahdis Nourimand, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. 2023. "Contradiction Detection in Financial Reports." In Proceedings of the Northern Lights Deep Learning Workshop 2023, Tromsø, Norway. Septentrio Academic Publishing. DOI: https://doi.org/10.7557/18.6799

We will first motivate the task by outlining the potential of AI applications, and especially NLU, in auditing. Following this, we describe our custom data set as well as the training methodology. The results are being presented and discussed with respect to their applicability in a real-world scenario, thereby contributing to our research questions. We close the chapter with a comprehensive summary and outlook on future perspectives of our approaches.

8.3.1 Introduction

Contradictions in written text are abundant and everywhere to be found. Sometimes they are amusing, like in the case of a newspaper article stating that the "earth circles the moon in 365 and a fraction days"[16] while discussing the astronomy behind the summer solstice. However, in this paper, we will dedicate our efforts to contradictions of more severe consequences: contradictions in financial reports. If such contradictions are not found and corrected before publication, they can lead to a plethora of

[16] Printed in the article *Ottawa vs. the equator* by the *Ottawa Citizen* on the 20th of June 2012.

issues for the reporting company including "bad operational decisions, reputational damage, economic loss, penalties, fines, legal action and even bankruptcy" [142].

The challenge of Contradiction Detection in financial documents can be considered from two different points of view. One looks at the numeric consistency of values mentioned and described in the document, e.g., if in one sentence the net profit is stated to be $500 and in another to be $600, this *numeric* contradiction should be detected[17]. Herein, we will analyze the other type of contradiction, the *semantic* contradiction. In this case, the contradiction is not of numerical nature, but can only be inferred from the actual meaning and implication of the sentence pair. Take this made-up sentence pair for example:

> "On 14th of March, 2020, we increased our capital by offering 5,000 new shares during a seasoned equity offering."
>
> "During 2020 we did not increase our total amount of equity and thus, it remained unchanged at $10,000,000."

These two statements by themselves are perfectly fine and numerically consistent, but as offering new shares during a seasoned equity offering *does* increase the equity of a company, the contradiction is only apparent if both sentences are evaluated together and at least some financial knowledge is present.

In this section of the applications chapter, we investigate how to detect contradictions in such a financial context. We analyze 24 different configurations and find that our best performing setup consists of a XLM-RoBERTa [22] encoder, infused with some additional pre-training as described in section 8.3.3, and fine-tuned on the Stanford Natural Language Inference Corpus [13]. It achieves a remarkable F1 score of 89.55% and is planned to be integrated into the auditing process of PricewaterhouseCoopers GmbH[18].

In the following, we first review related work. Subsection 8.3.3 describes our methodology, i.e., the additional pre-training method we applied and our general model architecture. Thereafter, in subsection 8.3.4 we outline our data set and the process of acquiring it, present our experiments, and discuss the results. We close this section with some concluding remarks and an outlook into conceivable future work.

8.3.2 Related Work

Contradiction Detection is a relatively recent field of Natural Language Processing (NLP). It mainly developed from the task of Natural Language Inference, also known as recognizing textual entailment, where the objective is to find whether two sentences either entail, contradict, or are not related to each other.

Before the emergence of deep, pre-trained transformer models like BERT [41] or RoBERTa [68], Contradiction Detection models used linguistic features which were extracted from texts beforehand to build a classifier. In this vein, [145] tried to find contradictions by leveraging three types of linguistic information: negation, antonymy, and semantic and pragmatic information associated with discourse relations. [9] evaluated a data set consisting only of contradictions by categorizing them into seven

[17] The approaches described in [143] and [144] solve this issue to some extend.

[18] The German division of PricewaterhouseCoopers, one of the largest auditing companies worldwide.

different classes. Further, [146] combined shallow semantic representations derived from semantic role labeling with binary relations extracted from sentences in a rule-based framework.

More recent advances usually leverage the power of huge, pre-trained models [4, 5, 41, 68] and are diverse in their application field and their language.

Regarding different applications, [97] identified conflicting findings reported in biomedical literature. [147] detected self-contradictions on an artificially balanced corpus of 1105 self-contradicting and 1105 negative non-self-contradiction. Furthermore, [148] improved chatbot responses by looking for contradictions in preceding conversation turns.

Besides English, Contradiction Detection was applied in Spanish [71], Japanese [149], Persian [70], and German [19, 67, 72].

In the broader spectrum of automating the auditing process of financial documents, which our Contradiction Detection approach is a part of, [72] introduced a recommender-based tool that streamlines and to some extent automates the auditing of financial documents. [150] updated it to leverage the power of a BERT encoder. A capsule network for the detection of fraud in accounting reports was proposed by [151]. [152] developed a joint named entity and relation extraction model based on BERT to extract key performance indicators and their numerical values from a corpus of German financial reports. [153] applied a similar approach to reports from the Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system, a platform hosted by the U.S. Securities and Exchange Commission, and published their data set along with their results. [144] also used a joint entity and relation extraction approach to cross-check formulas in Chinese financial documents. Another important aspect, the anonymization of such financial reports, was tackled by [154] by leveraging contextualized Natural Language Processing methods to recognize named entities. [155] employed transformer-based models with joint-task learning and their ensembles to classify whether a sentence contains any causality and to label phrases that indicate causes and consequences in a data set consisting of financial news - see also section 8.1 for more details on this topic. To achieve automatic indexing and information retrieval from large volumes of financial documents, [156] presented a document processing system based on a plethora of different Machine Learning techniques. Finally, [157] tried to autonomously generate financial reports from tabular data.

8.3.3 Methodology

In this section, we describe what additional pre-training methods we applied to the already pre-trained encoder in our classification setup and following that, the complete model architecture used to find contradictions in financial documents after the specific pre-training is explained.

Additional Pre-Training Our main objective in pre-training is enhancing the semantic knowledge stored in the model. To this end, we apply part-of-speech (POS) tagging as an additional pre-training objective. The task is to predict the syntactic function of each word in a sentence. Possible labels are, for example, “noun”, “verb”, “adverb” or “determiner”. Those can be context-dependent, e.g., “fly” or “break” can mean entirely different things, and therefore have different syntactic roles depending on the context. We assign subword tokens to the label of the word they belong to. The following example from our pre-training data set illustrates the approach.

```
_We _classify _our _short   -   _term _investments _as _available   -   _for _sale   .
PRON    VERB    PRON  ADJ   PUNCT NOUN      NOUN        ADP   ADJ     PUNCT ADP  NOUN PUNCT
```

The POS-tags are generated using the spaCy framework [66]. For implementation details and more information about the approach, please refer to [21] and chapter 4 of this thesis.

Model Architecture The actual model architecture consists of an encoder and a feed-forward neural network consecutively used for contradiction classification. The encoder is a large, pre-trained language model, of which we evaluated four different model variants during our experiments, in either its *vanilla*, i.e. with no further pre-training, state or injected with additional knowledge through some further pre-training as described in subsection 8.3.3. The classifier model used for the binary classification objective of finding a contradiction comprises a feed-forward neural network with a fine-tuned hyperparameter setup.

We use four different pre-trained base models for our experiments: XLM-RoBERTa [22], Financial-BERT [158], FinBERT [159], and a RoBERTa version trained on the Financial Phrasebank corpus by [160] titled Financial RoBERTa[19]. The models differ slightly with respect to their architecture and hyperparameter settings.

XLM-RoBERTa is a multi-lingual transformer encoder, which was pre-trained on the masked language modeling task for 100 languages. It has an embedding dimensionality of 1024, 24 hidden layers and 16 attention heads per layer, amounting to a total of 355 million trainable parameters. This model has shown to produce state-of-the-art results for many NLP tasks.

FinancialBERT and FinBERT are based on the standard BERT [41] implementation, whereas Financial-RoBERTa leverages a RoBERTa [68] model. They use a bert-base[20] or roberta–base[21] checkpoint, respectively. FinBERT and Financial-RoBERTa are further pre-trained on the Financial PhraseBank corpus by [160] for financial sentiment classification. FinancialBERT is pre-trained for next-sentence prediction and masked language modeling on a corpus of 3.39 billion tokens from the financial domain. All three have an embedding dimensionality of 768, and they have 12 hidden layers with 12 attention heads each. This amounts to 110 million trainable parameters, so they are considerably smaller in size than XLM–RoBERTa–large.

8.3.4 Experiments

In the upcoming subsections, we introduce our custom, proprietary data set, describe the training setup and model selection process in detail, and evaluate results. All experiments are conducted on two Nvidia Tesla V100 GPUs and the model as well as training code is implemented in PyTorch.

Data Our data set[22] consists of 640 manually collected and annotated sentence pairs in the English language, found in published financial documents (annual reports) and annotated by auditors of PricewaterhouseCoopers GmbH.

[19] https://huggingface.co/abhilash1910/financial_roberta

[20] https://huggingface.co/bert-base-uncased

[21] https://huggingface.co/roberta-base

[22] We are currently unable to publish the data set and the accompanying Python code because both are developed and used in the context of an industrial project and especially the annotated contradictions are confidential in nature.

	Paragraph 1	Paragraph 2	Label
1	Reversals of impairment losses recognized in previous years amounted to € in fiscal 2018 (2017: €). The largest reversal of impairment losses was recognized on in at € (2017: €) due to changed expectations regarding price developments.	As in the previous year, there was no requirement to recognise impairment losses or reversals of impairment losses on intangible assets in 2018.	`contradiction`
2	No significant events occurred after the end of the fiscal year.	No events have occurred since January 1, 2019, that will have a material impact on the net assets, financial position and results of operations of .	`no contradiction`
3	The total value of fixed assets in was € (previous year: €) of which, as in the previous year, none was pledged as collateral.	The total value of fixed assets in was € (previous year: €) of which, as in the previous year, € was pledged as collateral.	`contradiction`
4	As was the case at December 31, 2017, no treasury shares are held by at December 31, 2018.	The Executive Board is authorized, subject to the approval of the Supervisory Board, to increase the share capital by February 23, 2021, by up to € once or in several installments.	`not related`

Table 8.9: Example paragraph pairs from our financial Contradiction Detection data set. Information that can be used to identify a company or individuals has been anonymized here for data privacy reasons.

The data has been collected using two different annotation procedures. For the first method, a set of paragraphs from financial documents were presented to the annotators, who were asked to come up with a statement that would contradict the original one, and which could possibly be found in a financial document as well. This approach was chosen because the chance of finding real-world contradictions in a report or even across multiple documents is likely to be rather small, given that the reports have already been reviewed at the point when we receive them, and the probability of such errors happening is therefore overall rather low. A total of 145 examples were created using this method.

For the second method, the annotators were shown a list of already matched pairs of paragraphs from financial reports. This matching was achieved based on the heuristic of putting together paragraphs that refer to the same legal requirement according to previously made annotations by financial auditors, and which fulfil a certain text similarity criterion. Namely, we filter for those pairs of paragraphs, which get assigned a similarity score of 0.8 or higher by the spacy[23] [66] document similarity metric. The pairs of paragraphs are not necessarily found in the same document, so there is a small, but crucial chance that actual contradictions can occur. The annotators are then asked to mark every sample with one of three possible labels: `contradiction`, `no contradiction` or `not related`. The latter means that the two paragraphs refer to completely different facts or events, such that it is not meaningful to compare them with the objective of detecting contradictions. Those are then excluded

[23] https://spacy.io/usage/linguistic-features

Configuration	Recall in %	Precision in %	F1 in %
XLM-RoBERTa-large			
Fine-tuned on SNLI	70.59	68.57	69.57
Fine-tuned on finCD	67.65	76.67	71.88
Fine-tuned on SNLI & finCD	85.29	78.38	81.69
Pre-trained for POS-tagging and fine-tuned on SNLI	76.47	52.00	61.90
Pre-trained for POS-tagging and fine-tuned on finCD	82.35	80.00	81.16
Pre-trained for POS-tagging and fine-tuned on SNLI & finCD	**88.24**	**90.91**	**89.55**
FinancialBERT			
Pre-trained for POS-tagging and fine-tuned on SNLI & finCD	61.76	60.00	60.67
FinBERT			
Pre-trained for POS-tagging and fine-tuned on SNLI & finCD	64.71	56.41	60.27
Financial-RoBERTa			
Pre-trained for POS-tagging and fine-tuned on SNLI & finCD	35.29	44.44	39.34

Table 8.10: Test set evaluation of the Contradiction Detection task. We exclude the inferior configurations for FinancialBERT, FinBERT, and Financial–RoBERTa. The abbreviation *finCD* stands for our proprietary financial Contradiction Detection data set, which is described in section 8.3.4.

from the final data set. We generated another 495 examples using this approach.

A few anonymized examples of our data set are illustrated in Table 8.9. Furthermore, due to a maximum sequence length of 512 tokens which include premise, hypothesis, and separator tokens, a few data points had to be excluded from the final data set, so that we end up with a total of 626 samples. Out of those, 171 are labeled `contradiction`, and 455 `no contradiction`, yielding a slightly inbalanced label distribution. For the additional pre-training described in subsection 8.3.3, we utilize a data set of 47 000 paragraphs from financial reports in English. This data set, named the Financial Statement and Notes Data, is provided by the US Securities and Exchange Commission and is freely available on their website[24].

Training Setup As described above, we intialize the model parameters from a pre-trained checkpoint (XLM–RoBERTa–large[25], FinancialBERT[26], FinBERT[27] and Financial–RoBERTa[28], respectively). To find the best hyperparameter setup for each model, we conduct an extensive grid search evaluating various parameter and pre-training combinations based on the *validation* contradiction classification F1-score on the SNLI and/or our proprietary financial Contradiction Detection set. As a result of this hyperparameter optimization, we utilize the AdamW [80] optimizer in combination with a binary cross-entropy loss and a linear warm-up of three epochs (for pre-training) and two epochs (for fine-tuning). A learning rate of $5e^{-6}$ is used throughout the whole training procedure. Further, a dropout regularization of 0.2 is being applied during fine-tuning.

[24] https://www.sec.gov/dera/data/financial-statement-and-notes-data-set.html
[25] https://huggingface.co/xlm-roberta-large
[26] https://huggingface.co/ahmedrachid/FinancialBERT
[27] https://huggingface.co/ProsusAI/finbert
[28] https://huggingface.co/abhilash1910/financial_roberta

We train each model variation for 15 epochs and determine its best checkpoint via early stopping[29]. For the custom part-of-speech tagging pre-training, the model is being trained for a maximum of 25 epochs, as we observe that convergence happens slower than during fine-tuning.

Results As shown in Table 8.10, we achieve remarkable results in our task of Contradiction Detection in financial documents, demonstrated by the F1 score of 89.55% of our best model, the XLM-RoBERTa-large encoder, pre-trained for POS-tagging and fine-tuned both on the SNLI and our financial Contradiction Detection data set.

In detail, we find that the pre-training routine described in section 8.3.3 improves the performance significantly. Additionally, fine-tuning the Contradiction Detection model on both the SNLI and our proprietary financial contradiction data set further enhances the predictive power of our model. Furthermore, we observe a striking superiority of the XLM–RoBERTa-large encoder when compared to all smaller models, but especially those trained for financial documents.

8.3.5 Conclusion and Summary

In this section, we investigated how we can detect contradictions in a corpus of financial documents, which had been collected and annotated by expert financial auditors. We achieve a noteworthy performance with a Contradiction Detection F1-score of 89.55%, obtained by our best model, which incorporates a XLM–RoBERTa encoder further pre-trained for POS tagging and fine-tuned on the Stanford Natural Language Inference as well as on our financial Contradiction Detection data set.

Interestingly, the three encoder models pre-trained on financial data, namely FinancialBERT, FinBERT, and Financial–RoBERTa, underperformed compared to the "more general" XLM–RoBERTa by a considerable margin. We assume that there are two reasons for this. First, these three models are smaller in size than XLM-RoBERTa and second, the conducted pre-training on different financial documents and tasks might not generalize to our challenge of detecting financial contradictions.

This work and its accompanying industry project are part of a larger venture and long-time research project to enhance the financial auditing process with machine learning to lighten the workload of auditors and to find novel solutions to a plethora of issues faced by practitioners during the audit process. As a next step, the model described here will be integrated into a Machine Learning enhanced auditing software solution to help auditors find contradictions in financial documents. This will allow us to collect more data on identified and corrected contradictions, snowballing into an even better detection rate. Separate from this development, we are planning to provide models for Contradiction Detection in other languages, because financial reports of smaller companies are commonly only published in their local language. Furthermore, we plan on developing a generative model for contradiction generation based on our available data to be able to create financial contradictions from an arbitrary input document to alleviate the issue of the tedious manual annotation process.

Another, more practical open point with respect to this application is the issue of pre-filtering contradiction candidates. In our current evaluation setup, we only consider pairs of paragraphs that relate to the same topic or event, which is in line with the standard Natural Language Inference problem formulation. Looking at real-world use cases though, the problem is not so simple. If we sample sentence or paragraph pairs from a document, most of those will not be related in any way. In order to build a functional Contradiction Detection system for financial reports, this pre-filtering step

[29] Our best validation set contradiction F1-score is achieved in epoch 8.

would have to be addressed. There are multiple possible solutions, e.g., one could train a three-way classifier that distinguishes the categories `contradiction`, `no contradiction` or `not related`. Additionally, it might also be possible to implement a two-step approach, using a dedicated classifier or a heuristic to pre-filter pairs of paragraphs that are possibly related, and then apply Contradiction Detection on the remaining samples. In any case, there is the issue of a huge data imbalance, as the vast majority of possible pairs would actually not be related.

Furthermore, in order to determine whether a given pair of paragraphs are contradictory, some context information might be needed. So ideally, a model should take the whole document, or at least the surrounding paragraphs, into account. This could be accomplished by combining the transformer model with a recurrent mechanism that reads through the document from top to bottom (and/or the other way around).

This section concludes the applications chapter, which is the last major contribution of this thesis. We will continue with a conclusion that recaps the main findings from all chapters, provides a comprehensive summary w.r.t. our research questions, and outlines possibilities for future work.

CHAPTER 9

Conclusion

In this final chapter, we conclude the thesis by providing a comprehensive summary of our work. To this end, we revisit the three research questions that were formulated in the introduction (section 1.3), and elaborate on the main findings we gathered throughout this work. We also discuss limitations and possibilities for future research, as well as potential areas for applying our proposed approaches in the future. Finally, ethical and legal considerations concerning the applicability of our work are discussed.

9.1 Conclusion and Summary

In the following, we want to take a closer look at the contributions of this thesis with respect to each of the three research questions, starting with RQ1.

Research Question 1 (RQ1)

Can training with linguistically informed objectives and/or data augmentation improve language model efficiency in terms of training set size, model size and/or training time, while maintaining downstream performance?

This research question summarizes the key motivation for this thesis, and was mainly covered in chapters 4 and 7. In order to answer it, we developed two directions af research and implemented a set of solutions for each of those. The first paradigm is a novel pre-training regime for LMs with linguistically informed objectives, which was introduced in chapter 4. Specifically, we showcase three token-level objectives, namely Part-of-Speech-Tagging, Synset Prediction and syntactic Parent Prediction, that we apply to the LM before fine-tuning on the downstream task. Especially POS-Tagging and Synset Prediction show effective in improving model performance, which is particularly evident when using a smaller BERT-based model, supporting our hypothesis that those methods can help decrease model size while maintaining downstream performance.

The second research direction, which was covered in chapter 7, is about data augmentation using linguistic knowledge. We introduce four approaches that make use of generative AI and rules, thereby producing prototyical data that can be added to an existing training corpus, while reducing the overall data set size. We find that using these approaches is effective in enhancing model performance, when compared to a model being trained on a data set of similar size, but without prototypical samples.

This is a remarkable result, given the comparably small size of our data set. So we can show that our data augmentation procedures are successful in reducing the training set size (and thereby training time as well), without significantly sacrificing downstream performance.

To this end, the key contributions for RQ1 can be summarized as follows:

- **Informed Pre-Training Objectives:** We present a novel pre-training regime with the potential to enhance the development of resource-efficient LMs in the future.
- **Data Collection Routines:** We provide four novel approaches for data collection on the Contradiction Detection task, using linguistically informed generation approaches.
- **Linguistic Insights:** We present some linguistic insights on the nature of contradictory statements in German and English, which can potentially be helpful in future research on informed, resource-aware ML methods.

Research Question 2 (RQ2)

How do smaller, language-specific models trained with linguistically informed objectives and/or data augmentation perform compared to larger language-agnostic models for low-resource scenarios?

This RQ was investigated in the context of Natural Language Inference for Arabic and German, which can both be considered low-resource languages with respect to this task. In case of Arabic, we apply the informed pre-training routine presented in chapter 4, by training the model with semi-supervised labels for the Named Entity Recognition task before fine-tuning for NLI/CD. We find that it helps improve the performance especially for smaller transformer models such as AraBERT (see chapter 5).

For German, we follow a data augmentation approach, by machine-translating a large part of the SNLI training set from English. Our qualitative and quantitative evaluation verifies the soundness of this approach, as we can show that the data quality is overall high. We further train and evaluate different deep learning models, finding that RNN-based approaches are able to outperform multilingual BERT for this task on the translated data, which supports our hypothesis.

Overall, our key contributions to this RQ can be summarized as follows:

- **Novel Data Sets:** We introduce two new large-scale data collections for NLI in Arabic and German, which we make publicly available.
- **First Benchmark of Linguistically Informed Pre-Training in Arabic:** We provide the first comprehensive evaluation of our informed pre-training routine on a low-resource language with a non-Latin alphabet, showcasing its effectiveness in this specialized scenario.
- **First NLI Benchmark on a Machine-Translated Dataset in German:** We demonstrate the effectiveness of Machine Translation for data generation, using NLI in German language as a benchmark.

Research Question 3 (RQ3)

How do linguistically informed methods influence language modeling and understanding performance in real-world industry use cases?

We investigated this RQ in chapter 8, where we looked at three use cases with real-world relevance from the financial domain. In all three cases, linguistically informed methods yield some improvement over more generic, language-agnostic models, suggesting that they are to be considered a promising option in those scenarios. They can also to some extent alleviate the need for large quantities of manually annotated data, by transferring condensed expert knowledge into a data representation. Furthermore, the resulting models are relatively light-weight and can thus easily be deployed in resource-sparse scenarios with high compliance demands.

Specifically, we introduce an informed prompting approach for Causality Detection, an LLM-based data generation method for Translation Checks, and an application for the informed pre-training approach from section 4 in the context of Contradiction Detection in financial reports.

We summarize our key contributions as follows:

- **Linguistic Insights:** We gain some valuable insights into the linguistic nature of financial causality, opening up many possibilities for future research in this domain.
- **Informed Prompting Approach:** We present an informed promping approach for detecting financial causality, which could be extended to other use cases and domains, and validate its efficiency.
- **First Benchmark of LLM-based Data Generation for Translation Checks:** We validate the approach of using LLMs with informed prompting as data generators for the Translation Check task.
- **First Benchmark of Linguistically Informed Pre-Training for Contradiction Detection on Finance Data:** We demonstrate the effectiveness of our informed pre-training approach using POS-Tagging as an additional pre-training objective, by applying it on the task of detecting contradictions in financial reports.

9.2 Limitations and Future Work

While this thesis has laid the foundation for new research directions in the context of informed LM training and prompting, there are a number of limitations that need to be discussed, some of which have already been touched upon in the last chapters. To that effect, many possiblities for future work remain. In the following, we will summarize the limitations and corresponding objectives for upcoming research.

We consider these points as the main limitations of the work presented in this thesis:

- **Language-Dependency:** Most informed approaches presented throughout this thesis are more or less language-dependent. This is mostly due to the fact that information extraction methods such as Part-of-Speech-Tagging, Named Entity Regconition and Synset Extraction are relying

on resources and algorithms that have been implemented for a specific language. This issue could to some extent be alleviated by potent and language-agnostic LLM-based methods, although their use might not be feasible in every application scenario. It also goes against one of the main objectives of this work, namely the development of resource-efficient, light-weight methods. It would therefeore be worthwhile to investigate methods that allow for training smaller, multilingual models.

- **Lack of Data Validation:** The LLM-generated data collections from chapters 6, 7 and 8 have, for the most part, not been validated by human experts. This was due to constraints in time and resources, as manual validation of large data quantities is very costly. Also, we did not consider it a primary objective of this work, as we wanted to show that we can already achieve significant improvements without investing a lot of resources. Still, in order to further improve the approaches, some degree of human validation and refinement would be necessary.
- **Focus on Encoder-Based Models:** For a similar reason, mostly encoder-based models were trained and evaluated in the scope of this thesis. This is because we were focusing on classification tasks which do not require generative capabilities, and we wanted to show that we can already reach good performance using smaller, specialized models and minimal training effort. Still, in order to gain a broader understanding, and given the fact that text generation is one of the most relevant use cases nowadays, it would be important to also apply our methods to generative encoder-decoder or decoder-only models.
- **Limited Number of Application Domains:** While a number of different downstream applications for our approaches was discussed, those only reflect a small part of all relevant domains in which our methods could potentially be useful. Especially evaluation on more non-Indogermanic languages and specialized industry use cases (e.g., in medicine or law) would be very insightful. In order to gain an even better understanding on the nature of language and the strengths and weaknesses of our trained models, it would also be invaluable to analyze their outputs from a linguistic point of view.

In order to address those limitations, and to extend the impact of our research, the following points would be essential to address in future work:

- **Extension to more Languages and Application Domains:** It would be worthwhile to investigate the effectiveness of our proposed approaches in more languages, especially non-Indogermanic ones and those with a non-Latin alphabet, in order to test their robustness and generalizability. Also, we are very interested in the linguistic parallels and differences between different language families that could be discovered in this way.
- **Multilingual and Cross-lingual Pre-Training Objectives:** One main limitation of our approaches is their language-dependency (see above). We see a possibility to mitigate this to some extent by introducing multi- and cross-lingual informed pre-training objectives. This could be for example cross-lingual Masked Language Modeling, having the model predict a word from its context, but in a different language. In the context of generative models, cross-lingual reconstruction or denoising might be worth investigating, meaning that the model is trained to translate a mixed sample into one of its original languages . It could also be an interesting approach to pre-train on data in mixed languages using a language modeling objective, gradually

exchanging words from one language to the other, with the goal that the model would pick up on common features.

- **Informed Pre-Training of Generative Models:** The research in this thesis was mostly limited to encoder-only models, but could be extended to encoder-decoder or decoder-only models. This could be achieved e.g. using one of the informed pre-training objectives that were explored in this thesis, including the cross-lingual ones defined above, or by augmenting the data with additional encoded linguistic features.
- **More Work on Explainability:** The topic of explainability and content safety of LLM output was touched upon in this thesis, but not investigated in-depth. In order to build more robust methods, evaluation criteria and training objectives would need to be defined which explicitly take those features into account. These approaches could be derived e.g. using linguistic criteria such as readability and textual coherence, and/or using LLMs-as-a-judge with informed prompting.

9.3 Ethical and Legal Considerations

With respect to ethical and legal implications of the proposed approaches, to our best knowledge and understanding, there are three main issues that need to be considered. Those are mostly beyond the scope of this thesis, as they touch upon law and philosophy, but should be mentioned for the sake of completeness and transparency.

Firstly, the use of data that was generated by LLMs such as GPT-4 is legally difficult due to potential copyright issues. It is still an open question, whether models trained on this data can be used without legal risks especially in industry applications. This is why we suggest to employ open-source models for data generation in such scenarios, as it has been made transparent what data was used for training those models.

Secondly, the use of AI in the context of highly specialized industry use cases, particularly financial auditing, is always associated with some legal and ethical considerations. It has to be ensured that the AI does not make any decisions on its own and is always supervised by human experts, as it is not a legal entity and cannot be held accountable for any potential mistakes. This, of course, limits the scope of AI usage considerably. Decisionmakers have to carefully evaluate the pros and cons of implementing an AI component, and define a policy for human-AI interaction.

Last but not least, a general ethical implication of AI use in industry scenarios is its societal impact, e.g. in terms of human workers becoming partially or completely obsolete. All of the above-mentioned issues are difficult topics and require all parts of society to agree on a common understanding, as well as political measures to ensure that no one is left behind in this regard.

APPENDIX A

Supplementary Material

A.1 Prompts for Generating Contradiction Instances and Types (Chapter 7)

The following prompt is utilized for generating the hypotheses for the list of premises:

> **System**: You are an expert on semantics and linguistics, with a profound knowledge in Natural Language Processing. You are especially aware of the work by Marneffe et al., classifying different types of contradictions, such as factive, structural, lexical and world knowledge contradictions. The Premise is provided, you have to create a Hypothesis of one of the contradiction types for this premise. **User:** Please generate one contradictory Hypothesis for a PREMISE, based on CONTRADICTION_TYPE_DESCRIPTION. Format your response in the following way: CONTRADICTION_TYPE_NAME 'P: [PREMISE], H: [HYPOTHESIS]'. **Assistant:** CONTRADICTION_TYPE_DESCRIPTION

The placeholders stand for the following entities:

- PREMISE: the premise from SNLI data set which is used by the model as base for hypothesis generation
- CONTRADICTION_TYPE_NAME: Name of the type of contradiction that should be generated
- CONTRADICTION_TYPE_DESCRIPTION: Short description of the contradiction type to generate.

We use the following prompt for generating new contradiction instances, given a specific contradiction type:

System: You are an expert on semantics and linguistics, with a profound knowledge in Natural Language Processing. You are especially aware of the work by Marneffe et al., classifying different types of contradictions, such as contradictions arising from antonymy, negation, or numeric mismatch. To this end, a contradiction is defined as a mismatch between two statements, such that they cannot possibly both be true. It is assumed, that both statements refer to the same fact or event, even if this is not explicitly stated.

User: Please generate NUM_CONTRADICTIONS different contradictions based on CONTRADICTION_TYPE_NAME. The contradictions should be original and reasonably different from each other. Both premise and hypothesis should contain at least 10 words each, and should not be too similar. Please take care that they are actually contradicting and semantically meaningful. Be creative! Format your response in the following way: 'Premise: [PREMISE], Hypothesis: [HYPOTHESIS]'. Keep to this format strictly and do not add extra text or numbers.

Assistant: CONTRADICTION_TYPE_DESCRIPTION

The placeholders stand for the following entities:

- NUM_CONTRADICTIONS: Pre-defined number of contradictions to generate per type and iteration (set to 5)
- CONTRADICTION_TYPE_NAME: Name of the type of contradiction that should be generated
- CONTRADICTION_TYPE_DESCRIPTION: Short description of the contradiction type to generate.

The following prompt is being used for generating new contradiction types:

System: You are an expert on semantics and linguistics, with a profound knowledge in Natural Language Processing. You are especially aware of the work by Marneffe et al., classifying different types of contradictions, such as contradictions arising from antonymy, negation, or numeric mismatch.

User: Please come up with a new category of contradiction (other than KNOWN_TYPES). Format your output in the following way: Contradiction type name: [TYPE_NAME], Contradiction type description: [TYPE_DESCRIPTION].

Assistant: CONTRADICTION_TYPE_DESCRIPTIONS

Here the placeholders stand for:

- KNOWN_TYPES: List of all contradiction types already known at that point (both initial and self-generated).
- CONTRADICTION_TYPE_DESCRIPTIONS: List of descriptions for three randomly selected contradiction types from the pool of existing types.

A.2 Descriptions of the Contradiction Types which were Used in Prompts for Contradiction Generation (Chapter 7)

Factive (embedding context): Factive contradiction based on the embedding context means that a contradiction:

- arises from the mismatch in the embedding context of the verb phrase in the Premise and Hypothesis;
- contains similar or identical entities in the Premise and Hypothesis;
- Hypothesis does not contain any negations and any antonyms of the verb phrase of the Premise.

Example:
P: Sudan accepted U.N. troops in Darfur.
H: Sudan refused to accept U.N. troops.[a]

[a] https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

Factive (antonymy based): Factive contradiction based on the antonymy of a verb means that a contradiction arises between two statements (Premise and Hypothesis), because the verb phrase in Hypothesis has an opposite or contradictory meaning compared to the verb phrase of the Premise.

Example:
P: Sudan refused to allow U.N. troops in Darfur.
H: Sudan will grant permission for United Nations peacekeeping forces to take up station in Darfur.[a]

[a] https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

Structure: Structure contradiction arises from the mismatch in the sentence structure of the premise and hypothesis. The mismatch in the sentence structure has following features:

- the created Hypothesis has the same verb phrase as the Premise;
- either there are new entities which function as new objects of the same verb in the hypothesis, which creates the contradictory meaning toward the meaning of the premise or the subject and the object in the premise are swapped in the hypothesis;

Example:
P: The children are smiling and waving at the camera. H: The children are smiling and waving to each other.

Lexical: Lexical contradiction based on the mismatch in the lexical context has following features:

- the Premise and the Hypothesis has both the same topic or verb subject;
- the created Hypothesis has subtly different lexical meaning;
- the Hypothesis has a contradictory meaning due to the created opposite context of the topic in the premise;

Example:
P: Tariq Aziz kept outside the closed circle of Saddam s Sunni Moslem cronies.
H: Tariq Aziz was in Saddam s inner circle.[a]

[a] https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

World Knowledge Lexical contradiction based on the mismatch in world knowledge has following features:

- the Premise contains the well known knowledge about the world;
- the facts and knowledge from the Hypothesis contradict to the world knowledge in the Premise;

Examples: Premise='Al-zarqawi was Palestinian.' Hypothesis='Al-zarqawi was Jordanian.'[a]

[a] https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

A.3 Descriptions of the Contradiction Types Generated by GPT-4 (Chapter 7)

Temporal mismatch

Description: This contradiction arises when there's inconsistency between the time frames or chronological events presented in two statements. Hypothetically, if one statement indicates an event happening before another, and the contradictory statement implies the opposite sequence or suggests the events are simultaneous, a temporal mismatch is present.

Example:
P: The movie was released two months ago.
H: The actors are currently filming the sequel.

Aspectual contradictions

Description: These contradictions arise from mismatches in the aspectual properties of verbs or verb phrases between the premise and the claim. Aspect refers to the temporal structure of events or states as they are viewed from a specific standpoint in time. Aspectual contradictions can occur when the same event-state is characterized in conflicting ways; for example, in terms of its completion, frequency, duration or temporality. For instance, a premise stating 'John has been running for an hour' and a claim asserting 'John just started running' would form an aspectual contradiction, as they present the same action but with incompatible aspectual properties; specifically, contradictory assertions about the action's duration or initiation.

Example:
P: Mary has been studying French for years.
H: Mary has never studied French before.

Causal mismatch

Description: This contradiction arises when the cause and effect relationship implied in one statement is fundamentally at odds with or invalidated by another statement. For example, if one statement posits that a certain result is due to a specific cause, and a contradictory statement suggests that the same result is due to a completely different cause, or that the first cause doesn't lead to the mentioned effect, a causal mismatch is present. The contradiction is formed because the cause-and-effect relationships in the statements are incompatible. For example, a premise saying 'Rain makes the road slippery' and a claim stating 'Rain makes the road dry' would constitute a causal mismatch.

Example:
P: Eating a healthy diet leads to weight loss.
H: Eating junk food leads to weight loss.

Spatial mismatch

Description: This type of contradiction occurs when two statements or pieces of information present conflicting descriptions of physical or spatial arrangements. For example, one might state that a certain object or person is in a specific location, while the other places it somewhere else. This includes contradictions related to proximity, relative position, direction and geographical location.

Example:
P: The house is located on the top of the hill.
H: The house is situated in a valley deep underground.

Ideological mismatch

Description: This type of contradiction arises when two statements, while not necessarily directly opposing, conflict based on underlying ideological, philosophical, or theoretical frameworks. This could involve contradictions originating from differences in belief systems, moral values, or personal convictions. These contradictions may not result from antonymy, negation, or numeric mismatch. Rather, they emanate from deeper cognitive dissonance or juxtaposition of incongruent worldviews. For instance, two statements like 'Justice is swift punishment' and 'Justice is rehabilitation not punishment' may constitute an ideological mismatch, as they are based on fundamentally different beliefs about what 'justice' entails.

Example:
P: Capitalism is the only economic model that promotes and preserves individual liberty.
H: Socialism is a beneficial economic model that supports collective welfare and liberty.

Modal mismatch

Description: This category of contradiction arises when two statements are discordant in the modalities they imply or express. Modalities can range from possibility, necessity, obligation, permission, and ability. For example, the premise may assert that a course of action is necessary, while the contradicting statement may imply that the same action is merely possible or even unnecessary. This mismatch in modal claims leads to contradiction.

Example:
P: John is legally obliged to finish the project by next week as stated in their contractual agreement.
H: John has the option to complete the project anytime he wishes without any mandatory deadlines.

Quantitative mismatch

Description: This type of contradiction arises when two statements conflict due to inconsistent quantitative information.

Unlike numeric mismatch where explicit numbers contradict each other, quantitative mismatch happens when imprecise measures, orders of magnitude, or qualitative quantities clash between statements. For example, one statement might refer to an event or entity as being 'rare', while a conflicting statement describes it as 'common'. Similarly, one could say 'He consumed a large amount of water', while another says, 'He had little to drink.' This category is subtle as it requires inferencing and understanding of relative measures and estimates to detect contradictions.

Example:
P: The attendance at the local football match was exceptionally high, filling the stadium to the brim.
H: The local football match was not popular, with most of the stadium remaining empty.

Probabilistic mismatch

Description: This type of contradiction occurs when two statements provide different estimations of probability or likelihood for the same event or outcome. One statement may suggest that something is very likely to happen, while the other statement asserts that it's very unlikely or even impossible. In a broader sense, the contradiction can also include cases where the level of certainty or definiteness implicated in the statements is at odds. For instance, 'John will certainly attend the party' versus 'It's unlikely that John will attend the party' represents a Probabilistic Mismatch.

Example:
P: The meteorologist stated with certainty that the hurricane will strike the coast tomorrow morning.
H: The weather report forecasts a small chance of the hurricane reaching the coast tomorrow morning.

A.4 Prompts to Generate Data for the Translation Check Task (chapter 8, section 8.2)

In the following, we list the prompts that have been used to evaluate the performance of LLMs on the translation check task. "<ENGLISH>" and "<GERMAN>" are placeholders for the English and German sentence, respectively. All prompts result in similar performance.

Very compact prompts

Check if the following sentences are the correct translation of each other and respond with True or False only: 1. <ENGLISH> 2. <GERMAN>

Check if the following sentences are the correct translation of each other. A translation is considered incorrect too if one sentence has more information than the other. Answer with True if the translation is correct otherwise answer with false.
1. <ENGLISH> 2. <GERMAN>

Check if the following sentences are correct translation of each other. They are considered incorrect if they talk about different topics, contain a different amount of information or if they contain major mistakes such as a negation of the other sentence. Answer with True and false only.
1. <ENGLISH> 2. <GERMAN>

Few-shot prompt

Check if two sentences are the correct translation of each other. Respond with True or False only. Here are three examples:

1. Investment property Properties are allocated to investment property if a change in use has occurred, which is substantiated by their being occupied by another party after the end of owner-occupation or the inception of an operating lease with another party.

2. Wirtschaftliches Umfeld Die Risiken aus dem wirtschaftlichen Umfeld sind aufgrund der Fortschritte bei der Bekämpfung der Corona-Pandemie im Jahr 2021 gesunken.

Label: False

1. The aim of the reallocation is the further extension of the industry and automotive business of Benz outside the woodworking industry.

2. Das Ziel der Umverteilung besteht darin, das Industrie- und Automobilgeschäft von Benz außerhalb der Holzverarbeitungsindustrie weiter auszubauen, einschließlich des Einstiegs in neue Märkte wie die Luft- und Raumfahrtindustrie und die digitale Technologiebranche.

Label: False

1. As far as these are services, they are invoiced based on existing contracts.

2. Soweit es sich dabei um Dienstleistungen handelt, werden diese auf Basis bestehender Verträge abgerechnet.

Label: True

1. <ENGLISH> 2. <GERMAN>

Prompt created using GPT-3.5

Check if the following English and German sentences are correct translations of each other:
English Sentence: <ENGLISH>
German Sentence: <GERMAN>"
Is the German sentence a correct translation of the English sentence? (Answer: True/False)

APPENDIX B

List of Publications

- *Conference Papers (peer reviewed):*
  1. Rafet Sifa, **Maren Pielka**, Rajkumar Ramamurthy, Anna Ladi, Lars Hillebrand, and Christian Bauckhage. 2019. "Towards Contradiction Detection in German: a Translation-Driven Approach." In 2019 IEEE Symposium Series on Computational Intelligence (SSCI), pages 2497-2505, Xiamen, China. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/SSCI44817.2019.9003090
  2. **Maren Pielka**, Rafet Sifa, Lars Hillebrand, David Biesner, Rajkumar Ramamurthy, and Anna Ladi. 2020. "Tackling Contradiction Detection in German Using Machine Translation and End-to-End Recurrent Neural Networks." In 2020 25th International Conference on Pattern Recognition (ICPR), pages 6696-6701, Milan, Italy. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/ICPR48806.2021.9413257
  3. **Maren Pielka**, Felix Rode, Lisa Pucknat, Tobias Deußer, and Rafet Sifa. 2022. "A Linguistic Investigation of Machine Learning based Contradiction Detection Models: An Empirical Analysis and Future Perspectives." In 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), pages 1649-1653, Nassau, Bahamas. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/ICMLA55696.2022.00253
  4. **Maren Pielka**, Svetlana Schmidt, Lisa Pucknat, and Rafet Sifa. 2023. "Towards Linguistically Informed Multi-Objective Transformer Pre-Training for Natural Language Inference." In Advances in Information Retrieval (ECIR 2023), pages 553–561, Dublin, Ireland. Springer. DOI: https://doi.org/10.1007/978-3-031-28238-6_46
  5. Mohammad Majd Saad Al Deen, **Maren Pielka**, Jörn Hees, Bouthaina Soulef Abdou, and Rafet Sifa. 2023. "Improving Natural Language Inference in Arabic Using Transformer Models and Linguistically Informed Pre-Training." In 2023 IEEE Symposium Series on Computational Intelligence (SSCI), pages 318-322, Mexico City, Mexico. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/SSCI52147.2023.10371891
  6. **Maren Pielka** and Rafet Sifa. 2024. "Insights About Causalities in Financial Text - Towards an Informed Approach." In 2024 IEEE International Conference on Big Data

(BigData), pages 8801-8804, Washington, USA. DOI: https://doi.org/10.1109/BigData62323.2024.10825863

7. **Maren Pielka**, Marie-Christin Freischlad, Svetlana Schmidt, and Rafet Sifa. 2025. "Improving Language Model Performance by Training on Prototypical Contradictions." In Advances in Information Retrieval (ECIR 2025), pages 148-155, Lucca, Italy. Springer. DOI: https://doi.org/10.1007/978-3-031-88714-7_12

- *Journal Articles (peer reviewed):*

1. **Maren Pielka**, Max Hahnbück, Tobias Deußer, Daniel Uedelhoven, Moinam Chatterjee, Vijul Shah, Osama Soliman, Jannis von der Bank, Writwick Das, Maria Chiara Talarico, Cong Zhao, Carolina Held Celis, Christian Temath, and Rafet Sifa. 2025. "Automating Translation Checks of Financial Documents Using Large Language Models." In Language Resources and Evaluation (2025). Springer. DOI: https://doi.org/10.1007/s10579-025-09862-z

- *Workshop Articles (peer reviewed):*

1. **Maren Pielka**, Rajkumar Ramamurthy, Anna Ladi, Eduardo Brito, Clayton Chapman, Paul Mayer, and Rafet Sifa. 2020. "Fraunhofer IAIS at FinCausal 2020, Tasks 1 & 2: Using Ensemble Methods and Sequence Tagging to Detect Causality in Financial Documents." In Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 64-68, Barcelona, Spain. Association for Computational Linguistics.
2. Tobias Deußer, **Maren Pielka**, Lisa Pucknat, Basil Jacob, Tim Dilmaghani, Mahdis Nourimand, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. 2023. "Contradiction Detection in Financial Reports." In Proceedings of the Northern Lights Deep Learning Workshop 2023, Tromsø, Norway. Septentrio Academic Publishing. DOI: https://doi.org/10.7557/18.6799
3. **Maren Pielka**, Svetlana Schmidt, and Rafet Sifa. 2023. "Generating Prototypes for Contradiction Detection Using Large Language Models and Linguistic Rules." In 2023 IEEE International Conference on Big Data (BigData), pages 4684-4692, Sorrento, Italy. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/BigData59044.2023.10386499

- *Miscellaneous Papers (peer reviewed):*

  The following publications originated during the thesis and are related to the topics discussed here, but are not part of the thesis itself.

1. Lisa Pucknat, **Maren Pielka**, and Rafet Sifa. 2021. "Detecting Contradictions in German Text: A Comparative Study." In 2021 IEEE Symposium Series on Computational Intelligence (SSCI), pages 1-7, Orlando, USA. Institute for Electrical and Electronics Engineers. DOI: https://doi.org/10.1109/SSCI50451.2021.9659881
2. Lisa Pucknat, **Maren Pielka**, and Rafet Sifa. 2022. "Towards Informed Pre-Training for Critical Error Detection in English-German." In LWDA 2022 Workshops: FGWM, FGKD, and FGDB. Proceedings, pages 104-110, Hildesheim, Germany. CEUR Workshop Proceedings. DOI: https://doi.org/10.24406/publica-1332

Bibliography

[1] K. Spärck Jones, *A statistical interpretation of term specificity and its application in retrieval*, Journal of Documentation (1972) (cit. on pp. 1, 15, 47).

[2] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation", *Empirical Methods in Natural Language Processing (EMNLP)*, 2014 (cit. on pp. 1, 16, 58).

[3] A. Vaswani et al., "Attention is All You Need", *Conference on Neural Information Processing Systems*, 2017 (cit. on pp. 1, 15, 17, 97).

[4] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, *Improving Language Understanding by Generative Pre-Training*, (2018), URL: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf (cit. on pp. 1, 74, 87, 106).

[5] A. Radford et al., *Language Models are Unsupervised Multitask Learners*, (2019), URL: https://openai.com/blog/better-language-models (cit. on pp. 1, 74, 106).

[6] OpenAI, *GPT-4 Technical Report*, arXiv:2303.08774 (2023) (cit. on pp. 1, 32, 51, 74, 87, 97, 99, 102).

[7] DeepSeek-AI et al., *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*, 2025, arXiv: 2501.12948 [cs.CL], URL: https://arxiv.org/abs/2501.12948 (cit. on p. 1).

[8] DeepSeek-AI et al., *DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model*, 2024, arXiv: 2405.04434 [cs.CL], URL: https://arxiv.org/abs/2405.04434 (cit. on p. 1).

[9] M.-C. de Marneffe, A. N. Rafferty and C. D. Manning, "Finding Contradictions in Text", *Annual Meeting of the Association of Computational Linguistics (ACL)*, ACL, 2008 (cit. on pp. 2, 6, 22, 24, 29, 34, 47, 70, 71, 73, 75–77, 81, 105).

[10] M. Pielka, S. Schmidt and R. Sifa, "Generating Prototypes for Contradiction Detection Using Large Language Models and Linguistic Rules", *IEEE Big Data 2023*, 2023 (cit. on pp. 2, 7, 72, 95).

[11] M. Gritta and I. Iacobacci, "XeroAlign: Zero-shot cross-lingual transformer alignment", *International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2021 (cit. on pp. 2, 30).

[12] K. Yu, H. Li and B. Oguz, "Multilingual Seq2seq Training with Similarity Loss for Cross-Lingual Document Classification",
Third Workshop in Representation Learning for NLP, 2018 (cit. on pp. 2, 30).

[13] S. R. Bowman, G. Angeli, C. Potts and C. D. Manning,
"A Large Annotated Corpus for Learning Natural Language Inference",
Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015
(cit. on pp. 6, 24, 32–34, 47, 48, 55, 67, 73, 74, 105).

[14] L. Specia et al., "Findings of the WMT 2021 shared task on quality estimation",
Sixth Conference on Machine Translation (WMT), 2021 (cit. on pp. 6, 24, 25, 97, 99).

[15] D. Mariko, H. Abi-Akl, K. Trottier and M. El-Haj,
"The Financial Causality Extraction Shared Task (FinCausal 2022)",
4th Joint Workshop on Financial Narrative Processing @LREC2022, 2022
(cit. on pp. 6, 25, 85, 87, 92).

[16] D. Mariko et al.,
"The Financial Document Causality Detection Shared Task (FinCausal 2020)", *1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, 2020
(cit. on pp. 6, 25, 87, 92).

[17] A. Moreno-Sandoval et al.,
"The Financial Document Causality Detection Shared Task (FinCausal 2023)",
4th Joint Workshop on Financial Narrative Processing @BigData2023, 2023
(cit. on pp. 6, 25, 87, 92).

[18] R. Sifa et al., "Towards Contradiction Detection in German: A Translation-driven Approach",
IEEE Symposium Series on Computational Intelligence (SSCI), 2019
(cit. on pp. 7, 32, 33, 35, 53).

[19] M. Pielka et al., "Tackling Contradiction Detection in German Using Machine Translation and End-to-End Recurrent Neural Networks",
International Conference on Pattern Recognition (ICPR), 2021 (cit. on pp. 7, 32, 33, 53, 106).

[20] M. Pielka, F. Rode, L. Pucknat, T. Deußer and R. Sifa,
"A Linguistic Investigation of Machine Learning based Contradiction Detection Models: An Empirical Analysis and Future Perspectives",
International Conference on Machine Learning and Applications (ICMLA), 2022
(cit. on pp. 7, 32, 74).

[21] M. Pielka, S. Schmidt, L. Pucknat and R. Sifa, "Towards Linguistically Informed Multi-Objective Transformer Pre-Training for Natural Language Inference",
European Conference on Information Retrieval (ECIR), 2023
(cit. on pp. 7, 32, 47, 50, 74, 107).

[22] A. Conneau et al., *Unsupervised cross-lingual representation learning at scale*,
Annual Meeting of the Association for Computational Linguistics (ACL) (2019)
(cit. on pp. 7, 32–35, 49, 82, 97, 100, 105, 107).

[23] A. Wahab and R. Sifa,
"DIBERT: Dependency Injected Bidirectional Encoder Representations from Transformers",
IEEE Symposium Series on Computational Intelligence (SSCI), 2021
(cit. on pp. 7, 30, 33, 40).

[24] M. Saad Al Deen, M. Pielka, J. Hees, B. Abdou and R. Sifa, "Improving Natural Language Inference in Arabic Using Transformer Models and Linguistically Informed Pre-Training",
IEEE Symposium Series on Computational Intelligence (SSCI), 2023 (cit. on pp. 7, 45).

[25] W. Antoun, F. Baly and H. Hajj,
AraBERT: Transformer-based Model for Arabic Language Understanding,
4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection (2020) (cit. on p. 7).

[26] M. Pielka et al., "Fraunhofer IAIS at FinCausal 2020, Tasks 1 & 2: Using Ensemble Methods and Sequence Tagging to Detect Causality in Financial Documents",
FNP-FNS at Coling 2020, 2020 (cit. on pp. 8, 86).

[27] T. Deußer et al., "Contradiction Detection in Financial Reports",
Northern Lights Deel Learning (NLDL) Workshop, 2023 (cit. on pp. 8, 42, 87, 95, 97, 104).

[28] M. Pielka and R. Sifa,
"Insights About Causalities in Financial Text - Towards an Informed Approach",
IEEE Big Data Conference, 2024 (cit. on pp. 8, 86).

[29] M. Pielka et al.,
"Automating Translation Checks of Financial Documents Using Large Language Models",
Language Resources and Evaluation journal, 2025 (cit. on pp. 8, 96).

[30] A. Cauchy, *Méthode générale pour la résolution des systèmes d'équations simultanées*,
C.R. Acad. Sci. Paris (1847) (cit. on p. 15).

[31] M. Mohri, *Foundationd of Machine Learning*, MIT Press, 2018 (cit. on p. 15).

[32] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016 (cit. on p. 15).

[33] K. O'Shea and R. Nash, *An Introduction to Convolutional Neural Networks*,
ArXiv **abs/1511.08458** (2015),
URL: https://api.semanticscholar.org/CorpusID:9398408 (cit. on p. 15).

[34] L. C. Jain and L. R. Medsker, "Recurrent Neural Networks: Design and Applications", 1999,
URL: https://api.semanticscholar.org/CorpusID:262144264 (cit. on p. 15).

[35] U. Michelucci, *An Introduction to Autoencoders*, ArXiv **abs/2201.03898** (2022),
URL: https://api.semanticscholar.org/CorpusID:245853675 (cit. on p. 15).

[36] M. Wankhade, A. C. S. Rao and C. Kulkarni,
A survey on sentiment analysis methods, applications, and challenges,
Artificial Intelligence Review **55** (2022) (cit. on p. 15).

[37] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification",
Named Entities: Recognition, classification and use,
John Benjamins publishing company, 2009 (cit. on p. 15).

[38] T. Mikolov, K. Chen, G. S. Corrado and J. Dean,
"Efficient Estimation of Word Representations in Vector Space",
International Conference on Learning Representations, 2013 (cit. on pp. 16, 47).

[39] M. E. Peters et al., "Deep Contextualized Word Representations",
Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers),
Association for Computational Linguistics, 2018 (cit. on pp. 16, 89).

[40] A. Akbik, D. Blythe and R. Vollgraf, "Contextual String Embeddings for Sequence Labeling",
27th International Conference on Computational Linguistics,
Association for Computational Linguistics, 2018 (cit. on pp. 16, 58, 62, 63).

[41] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova,
"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding",
North American Chapter of the Association for Computational Linguistics, 2019
(cit. on pp. 16, 30, 33, 49, 79, 97, 100, 105–107).

[42] Y. Wu et al., *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*, ArXiv **abs/1609.08144** (2016),
URL: https://api.semanticscholar.org/CorpusID:3603249 (cit. on p. 17).

[43] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing", *2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*,
Association for Computational Linguistics, 2018,
URL: https://aclanthology.org/D18-2012/ (cit. on p. 17).

[44] P. Gage, *A New Algorithm for Data Compression*, The C User Journal (1994),
URL: http://www.pennelynn.com/Documents/CUJ/HTML/94HTML/19940045.HTM
(cit. on p. 17).

[45] M. Ali et al., "Tokenizer Choice For LLM Training: Negligible or Crucial?",
Findings of the Association for Computational Linguistics: NAACL 2024,
Association for Computational Linguistics, 2024 (cit. on p. 17).

[46] J. Gehring, M. Auli, D. Grangier, D. Yarats and Y. N. Dauphin,
"Convolutional sequence to sequence learning",
34th International Conference on Machine Learning - Volume 70, JMLR.org, 2017
(cit. on p. 17).

[47] P. J. Liu et al., *Generating Wikipedia by Summarizing Long Sequences*,
ArXiv **abs/1801.10198** (2018) (cit. on p. 19).

[48] K. Hengeveld, *Parts of speech*, Anstey, MP et Mackenzie, JL (éds). Crucial readings in functional grammar. Berlin: Mouton de Gruyter (2005) (cit. on p. 20).

[49] M. Maru, F. Scozzafava, F. Martelli and R. Navigli, "SyntagNet: Challenging Supervised Word Sense Disambiguation with Lexical-Semantic Combinations",
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,
URL: https://aclanthology.org/D19-1359/ (cit. on p. 22).

[50] C. Fellbaum, *WordNet: An Electronic Lexical Database*, Bradford Books, 1998 (cit. on pp. 22, 39, 40).

[51] B. Hamp and H. Feldweg, "GermaNet - a Lexical-Semantic Net for German", *ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Association for Computational Linguistics, 1997 (cit. on p. 22).

[52] V. Henrich and E. Hinrichs, "GernEdiT - The GermaNet Editing Tool", *Seventh Conference on International Language Resources and Evaluation (LREC 2010)*, Association for Computational Linguistics, 2010 (cit. on p. 22).

[53] M. Bunge, *Causality and Modern Science*, Courier Corp., 2012 (cit. on p. 23).

[54] V. Jijkoun and M. Rijke, *Recognizing Textual Entailment Using Lexical Similarity*, Journal of Colloid and Interface Science (2005) (cit. on pp. 23, 28).

[55] B. MacCartney, T. Grenager, M.-C. de Marneffe, D. M. Cer and C. D. Manning, "Learning to recognize features of valid textual entailments", *North American Chapter of the Association for Computational Linguistics*, 2006 (cit. on pp. 24, 28).

[56] J. Yang, S. C. Han and J. Poon, *A survey on extraction of causal relations from natural language text*, Knowledge and Information Systems **64** (2022) (cit. on p. 25).

[57] E. Blanco, N. Castell and D. I. Moldovan, "Causal Relation Extraction.", *Lrec*, vol. 66, 2008 74 (cit. on p. 25).

[58] G. E. Hinton, O. Vinyals and J. Dean, *Distilling the Knowledge in a Neural Network*, NeurIPS Workshop 2014 **abs/1503.02531** (2015), URL: https://api.semanticscholar.org/CorpusID:7200347 (cit. on p. 28).

[59] M. Abdin et al., *Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone*, 2024, URL: https://arxiv.org/abs/2404.14219 (cit. on pp. 28, 29, 83).

[60] I. Dagan, O. Glickman and B. Magnini, "The PASCAL recognising textual entailment challenge", Springer-Verlag, 2005 (cit. on p. 28).

[61] M.-C. de Marneffe et al., "Aligning Semantic Graphs for Textual Inference and Machine Reading", 2007 (cit. on p. 28).

[62] L. Pucknat, M. Pielka and R. Sifa, "Towards Informed Pre-Training for Critical Error Detection in English-German", *Lernen. Wissen. Daten. Analysen. (LWDA)*, 2022 (cit. on pp. 29, 97).

[63] S. Gunasekar et al., *Textbooks Are All You Need*, ArXiv **abs/2306.11644** (2023), URL: https://api.semanticscholar.org/CorpusID:259203998 (cit. on p. 29).

[64] L. von Rueden, S. Houben, K. Cvejoski, C. Bauckhage and N. Piatkowski, *Informed Pre-Training on Prior Knowledge*, arXiv preprint arXiv:2205.11433 (2022) (cit. on pp. 29, 74).

[65] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits", 2005, URL: https://api.semanticscholar.org/CorpusID:60282629 (cit. on p. 29).

[66] M. Honnibal, I. Montani, S. Van Landeghem and A. Boyd, *spaCy: Industrial-strength natural language processing in python*, (2020) (cit. on pp. 30, 39, 107, 108).

[67] L. Pucknat, M. Pielka and R. Sifa, "Detecting Contradictions in German Text: A Comparative Study", *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2021 (cit. on pp. 32–34, 36, 47, 106).

[68] Y. Liu et al., *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, ArXiv **abs/1907.11692** (2019), URL: https://api.semanticscholar.org/CorpusID:198953378 (cit. on pp. 32–35, 46, 105–107).

[69] I. Dagan, O. Glickman and B. Magnini, "The Pascal Recognising Textual Entailment Challenge", *Machine Learning Challenges Workshop*, Springer, 2005 (cit. on p. 33).

[70] Z. Rahimi and M. ShamsFard, *Contradiction Detection in Persian Text*, arXiv preprint arXiv:2107.01987 (2021) (cit. on pp. 33, 106).

[71] R. Sepúlveda-Torres, A. Bonet-Jover and E. Saquete, *"Here Are the Rules: Ignore All Rules": Automatic Contradiction Detection in Spanish*, Applied Sciences **11** (2021) (cit. on pp. 33, 106).

[72] R. Sifa et al., "Towards Automated Auditing with Machine Learning", *ACM Symposium on Document Engineering 2019*, 2019 (cit. on pp. 33, 54, 87, 95, 97, 106).

[73] L. Li, B. Qin and T. Liu, *Contradiction Detection with Contradiction-Specific Word Embedding*, Algorithms **10** (2017) (cit. on pp. 34, 66, 70).

[74] M. Barbouch, S. Verberne and T. Verhoef, *WN-BERT: Integrating WordNet and BERT for Lexical Semantics in Natural Language Understanding*, Computational Linguistics in the Netherlands Journal **11** (2021) (cit. on p. 34).

[75] A. Wang et al., *GLUE: A multi-task benchmark and analysis platform for natural language understanding*, arXiv preprint arXiv:1804.07461 (2018) (cit. on p. 34).

[76] J. Zhou, Z. Zhang and H. Zhao, *LIMIT-BERT : Linguistic Informed Multi-Task BERT*, CoRR **abs/1910.14296** (2019) (cit. on p. 34).

[77] P. Qi, Y. Zhang, Y. Zhang, J. Bolton and C. D. Manning, *Stanza: A Python natural language processing toolkit for many human languages*, arXiv preprint arXiv:2003.07082 (2020) (cit. on p. 40).

[78] G. A. Miller, *WordNet: a lexical database for English*, Communications of the ACM **38** (1995) (cit. on pp. 40, 75).

[79] Y. Zhu et al., *Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books*, 2015, URL: https://arxiv.org/abs/1506.06724 (cit. on p. 41).

[80] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, arXiv preprint arXiv:1711.05101 (2017) (cit. on pp. 41, 50, 109).

[81] S. Wang, H. Fang, M. Khabsa, H. Mao and H. Ma, *Entailment as Few-Shot Learner*, 2021 (cit. on pp. 41, 42).

[82] Z. Sun et al., *Self-Explaining Structures Improve NLP Models*, arXiv preprint arXiv:2012.01786v2 (2020) (cit. on p. 42).

[83] M. S. al Deen, *Informierte Pre-Training Methoden für Natural Language Inference im Arabischen*, Bachelor's thesis: Hochschule Bonn-Rhein-Sieg, 2023 (cit. on p. 45).

[84] H. Elsafty et al., "ArDia: Improving Arabic Dialectal Language Classification Using a Novel Dataset", *International AAAI Conference on Web and Social Media (ICWSM)*, Association for the Advancement of Artificial Intelligence, 2025 (cit. on p. 46).

[85] F. Aldabbas, S. Ashraf, R. Sifa and L. Flek, "MultiProp Framework: Ensemble Models for Enhanced Cross-Lingual Propaganda Detection in Social Media and News using Data Augmentation, Text Segmentation, and Meta-Learning", *1st Workshop on NLP for Languages Using Arabic Script*, Association for Computational Linguistics, 2025 (cit. on p. 46).

[86] B. MacCartney and C. D. Manning, "Natural Logic for Textual Inference", *ACL-PASCAL@ACL*, 2007 (cit. on p. 46).

[87] A. Mishra et al., "Reading Comprehension as Natural Language Inference:A Semantic Analysis", *Ninth Joint Conference on Lexical and Computational Semantics*, Association for Computational Linguistics, 2020 (cit. on p. 46).

[88] G. Lai, Q. Xie, H. Liu, Y. Yang and E. Hovy, "RACE: Large-scale ReAding Comprehension Dataset From Examinations", *Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2017 (cit. on p. 46).

[89] K. A. Jallad and N. Ghneim, *ArNLI: Arabic Natural Language Inference for Entailment and Contradiction Detection*, Comput. Sci. **24** (2022) (cit. on pp. 47, 48).

[90] M. F. Porter, *An algorithm for suffix stripping*, Program **40** (1997) (cit. on p. 47).

[91] A. Conneau et al., "XNLI: Evaluating Cross-lingual Sentence Representations", *Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018 (cit. on pp. 47, 48, 64, 70).

[92] Ž. Agić and N. Schluter, "Baselines and Test Data for Cross-Lingual Inference", *Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), 2018 (cit. on p. 48).

[93] O. Obeid et al.,
"CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing",
Language Resources and Evaluation Conference,
European Language Resources Association, 2020 (cit. on p. 48).

[94] W. Antoun, F. Baly and H. Hajj,
"AraBERT: Transformer-based Model for Arabic Language Understanding",
4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, European Language Resource Association, 2020
(cit. on p. 49).

[95] T. Akiba, S. Sano, T. Yanase, T. Ohta and M. Koyama,
"Optuna: A Next-generation Hyperparameter Optimization Framework", *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*,
Association for Computing Machinery, 2019 (cit. on p. 50).

[96] M. Pielka, *Neural Network Methods for Natural Language Inference in German*,
Master's thesis: Rheinische Friedrich-Wilhelms-Universität Bonn, 2019 (cit. on p. 53).

[97] N. S. Tawfik and M. R. Spruit,
"Automated Contradiction Detection in Biomedical Literature", *IAPR International Conference on Machine Learning and Data Mining in Pattern Recognition*, 2018
(cit. on pp. 54, 106).

[98] A. Conneau, D. Kiela, H. Schwenk, L. Barrault and A. Bordes, "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data",
Conference on Empirical Methods in Natural Language Processing (EMNLP)
(cit. on pp. 54, 55, 58, 59, 63).

[99] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kociský and P. Blunsom,
Reasoning about Entailment with Neural Attention, CoRR **abs/1509.06664** (2015)
(cit. on pp. 55, 58, 60).

[100] X. Liu, K. Duh and J. Gao, "Stochastic Answer Networks for Natural Language Inference",
56th Annual Meeting of the Association for Computational Linguistics,
Association for Computational Linguistics, 2018 (cit. on p. 55).

[101] X. Liu, P. He, W. Chen and J. Gao,
"Multi-Task Deep Neural Networks for Natural Language Understanding",
57th Annual Meeting of the Association for Computational Linguistics,
Association for Computational Linguistics, 2019 (cit. on p. 55).

[102] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu,
"BLEU: a Method for Automatic Evaluation of Machine Translation",
40th Annual Meeting of the Association for Computational Linguistics, 2002
(cit. on pp. 56, 57).

[103] D. Bahdanau, K. Cho and Y. Bengio,
Neural Machine Translation by Jointly Learning to Align and Translate,
CoRR **abs/1409.0473** (2014) (cit. on pp. 57, 60).

[104] M. Johnson et al.,
Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation,
Transactions of the Association for Computational Linguistics **5** (2017) (cit. on p. 57).

[105] G. Lample and A. Conneau, *Cross-lingual Language Model Pretraining*,
Advances in Neural Information Processing Systems (NeurIPS) (2019) (cit. on pp. 58, 64).

[106] J. Ramos, *Using TF-IDF to Determine Word Relevance in Document Queries*, tech. rep.,
Rutgers University Dept. of CS, 2003 (cit. on p. 58).

[107] C. Manning, P. Raghavan and H. Schütze, *Introduction to Information Retrieval*,
Cambridge University Press, 2008 (cit. on p. 58).

[108] Q. V. Le and T. Mikolov, *Distributed Representations of Sentences and Documents*,
CoRR **abs/1405.4053** (2014) (cit. on pp. 58, 62, 88).

[109] K. Cho et al., *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, CoRR **abs/1406.1078** (2014) (cit. on p. 58).

[110] A. Graves, N. Jaitly and A.-r. Mohamed,
"Hybrid Speech Recognition with Deep Bidirectional LSTM",
2013 IEEE workshop on automatic speech recognition and understanding, IEEE, 2013
(cit. on p. 58).

[111] J. Wieting and D. Kiela,
No training required: Exploring random encoders for sentence classification,
arXiv preprint arXiv:1901.10444 (2019) (cit. on p. 58).

[112] J. Chung, Ç. Gülçehre, K. Cho and Y. Bengio,
Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,
arXiv preprint 1412.3555 (2014) (cit. on p. 58).

[113] E. Million, *The Hadamard Product*, (2007), URL:
http://buzzard.ups.edu/courses/2007spring/projects/million-paper.pdf
(cit. on p. 59).

[114] E. Brito, R. Sifa, K. Cvejoski, C. Ojeda and C. Bauckhage,
"Towards German Word Embeddings: A Use Case with Predictive Sentiment Analysis",
Data Science–Analytics and Applications, Springer, 2017 (cit. on p. 62).

[115] J. H. Lau and T. Baldwin, "An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation", *1st Workshop on Representation Learning for NLP*,
Association for Computational Linguistics, 2016 (cit. on p. 62).

[116] A. Williams, N. Nangia and S. Bowman,
"A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference",
Conference of the North American Chapter of the Association for Computational Linguistics,
Association for Computational Linguistics (cit. on p. 70).

[117] S. Schmidt, *Automatic contradiction detection using a dataset of prototypical contradictions*,
Master's thesis: Ruhr-Universität Bochum, 2024 (cit. on p. 72).

[118] M. Pielka, M.-C. Freischlad, S. Schmidt and R. Sifa,
"Improving Language Model Performance by Training on Prototypical Contradictions",
European Conference on Information Retrieval (ECIR), 2025 (cit. on p. 72).

[119] R. B. Haim et al., "The second pascal recognising textual entailment challenge", *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, vol. 7, 2006 (cit. on p. 73).

[120] T. Brown et al., "Language Models are Few-Shot Learners", *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020 (cit. on pp. 74, 87).

[121] Y. Wang et al., "Self-Instruct: Aligning Language Models with Self-Generated Instructions", *Annual Meeting of the Association for Computational Linguistics*, 2022 (cit. on pp. 74, 77).

[122] R. Nairn, C. Condoravdi and L. Karttunen, "Computing relative polarity for textual inference", *Proceedings of the fifth international workshop on inference in computational semantics (icos-5)*, 2006 (cit. on p. 76).

[123] H. Touvron et al., *LLaMA: Open and Efficient Foundation Language Models*, ArXiv **abs/2302.13971** (2023), URL: https://api.semanticscholar.org/CorpusID:257219404 (cit. on pp. 83, 97, 102).

[124] L. Hillebrand et al., "KPI-BERT: A joint Named Entity Recognition and Relation Extraction Model for Financial Reports", *International Conference on Pattern Recognition (ICPR)*, 2022 (cit. on pp. 87, 97).

[125] F. Pedregosa et al., *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research (2011) (cit. on p. 88).

[126] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, 2016 (cit. on p. 88).

[127] W. Yin, K. Kann, M. Yu and H. Schütze, *Comparative Study of CNN and RNN for Natural Language Processing*, ArXiv **abs/1702.01923** (2017) (cit. on p. 88).

[128] Martín Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015, URL: http://tensorflow.org/ (cit. on p. 88).

[129] F. Chollet et al., *Keras*, 2015, URL: https://github.com/fchollet/keras (cit. on p. 88).

[130] A. Akbik, D. Blythe and R. Vollgraf, "Contextual String Embeddings for Sequence Labeling", *COLING 2018, 27th International Conference on Computational Linguistics*, 2018 (cit. on p. 89).

[131] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing", *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, 2020 (cit. on p. 89).

[132] J. Ratcliff and D. Metzener, "Pattern Matching: The Gestalt Approach", *Dr. Dobb's Journal*, vol. 46, 1988 (cit. on p. 94).

[133] E. Brito et al., "A Hybrid AI Tool to Extract Key Performance Indicators from Financial Reports for Benchmarking", *Proceedings of the ACM Symposium on Document Engineering 2019*, 2019 (cit. on p. 95).

[134] E. Brito, M. Lübbering, D. Biesner, L. P. Hillebrand and C. Bauckhage, *Towards Supervised Extractive Text Summarization via RNN-based Sequence Classification*, arXiv preprint arXiv:1911.06121 (2019) (cit. on p. 95).

[135] D. Biesner, E. Brito, L. P. Hillebrand and R. Sifa, "Hybrid Ensemble Predictor as Quality Metric for German Text Summarization: Fraunhofer IAIS at GermEval 2020 Task 3", *5th Swiss Text Analytics Conference (SwissText) & 16th Conference on Natural Language Processing (KONVENS)*, 2020 (cit. on p. 95).

[136] C. Zerva et al., "Findings of the WMT 2022 Shared Task on Quality Estimation", *Seventh Conference on Machine Translation (WMT)*, Association for Computational Linguistics, 2022 (cit. on p. 97).

[137] V. Sanh, L. Debut, J. Chaumond and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter", *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019*, 2019 (cit. on pp. 97, 100).

[138] L. Hillebrand et al., "Towards automating Numerical Consistency Checks in Financial Reports", *IEEE International Conference on Big Data (IEEE BigData)*, 2022 (cit. on p. 97).

[139] S. Bird, E. Loper and E. Klein, *Natural Language Processing with Python*, O'Reilly Media Inc., 2009 (cit. on p. 98).

[140] W. A. Gale, K. W. Church et al., *A program for aligning sentences in bilingual corpora*, Computational linguistics **19** (1994) (cit. on p. 98).

[141] L. Wang et al., *Multilingual E5 Text Embeddings: A Technical Report*, arXiv preprint arXiv:2402.05672 (2024) (cit. on p. 100).

[142] K. Russo, *What Are the Risks of Inaccurate Financial Reporting?*, ed. by O. Corporation, [Online; posted 21/03/2022; retrieved 22/08/2022], 2022, URL: https://www.netsuite.com/portal/resource/articles/accounting/inaccurate-financial-reporting.shtml (cit. on p. 105).

[143] L. P. Hillebrand et al., *Towards automating Numerical Consistency Checks in Financial Reports*, 2022 IEEE International Conference on Big Data (Big Data) (2022) (cit. on p. 105).

[144] Y. Cao, H. Li, P. Luo and J. Yao, "Towards Automatic Numerical Cross-Checking: Extracting Formulas from Text", *2018 World Wide Web Conference*, International World Wide Web Conferences Steering Committee, 2018 (cit. on pp. 105, 106).

[145] S. Harabagiu, A. Hickl and F. Lacatusu, "Negation, contrast and contradiction in text processing", *21st National Conference on Artificial Intelligence - Volume 1*, AAAI Press, 2006 (cit. on p. 105).

[146] M. Q. N. Pham, M. L. Nguyen and A. Shimazu,
"Using Shallow Semantic Parsing and Relation Extraction for Finding Contradiction in Text",
Sixth International Joint Conference on Natural Language Processing,
Asian Federation of Natural Language Processing, 2013 (cit. on p. 106).

[147] C.-M. Hsu, C.-t. Li, D. Sáez-Trumper and Y.-Z. Hsu,
WikiContradiction: Detecting Self-Contradiction Articles on Wikipedia,
2021 IEEE International Conference on Big Data (BigData) (2021) (cit. on p. 106).

[148] D. Jin, S. Liu, Y. Liu and D. Hakkani-Tur,
"Improving Bot Response Contradiction Detection via Utterance Rewriting",
23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue,
Association for Computational Linguistics, 2022 (cit. on p. 106).

[149] Y. Takabatake et al.,
"Classification and Acquisition of Contradictory Event Pairs using Crowdsourcing",
3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation,
Association for Computational Linguistics, 2015 (cit. on p. 106).

[150] R. Ramamurthy et al., "ALiBERT: improved automated list inspection (ALI) with BERT",
21st ACM Symposium on Document Engineering,
Association for Computing Machinery, 2021 (cit. on p. 106).

[151] F. Zhu, D. Ning, Y. Wang and S. Liu,
"A Novel Cost-sensitive Capsule Network for Audit Fraud Detection", *20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*,
2021 (cit. on p. 106).

[152] L. Hillebrand et al., "KPI-BERT: A Joint Named Entity Recognition and Relation Extraction Model for Financial Reports", *26th International Conference on Pattern Recognition (ICPR)*,
2022 (cit. on p. 106).

[153] T. Deußer et al., "KPI-EDGAR: A Novel Dataset and Accompanying Metric for Relation Extraction from Financial Documents",
21st IEEE International Conference on Machine Learning and Applications (ICMLA), 2022
(cit. on p. 106).

[154] D. Biesner et al., *Anonymization of German financial documents using neural network-based language models with contextual word representations*,
Springer International Journal of Data Science and Analytics (2021) (cit. on p. 106).

[155] D. Gordeev, A. Davletov, A. Rey and N. Arefiev, "LIORI at the FinCausal 2020 Shared task",
1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, COLING, 2020 (cit. on p. 106).

[156] R. Ramamurthy et al., "2021 IEEE Symposium Series on Computational Intelligence",
Proc. SSCI, 2021 (cit. on p. 106).

[157] C. L. Chapman et al.,
"Towards Generating Financial Reports from Tabular Data Using Transformers",
International IFIP Cross Domain (CD) Conference for Machine Learning & Knowledge Extraction (MAKE) CD-MAKE, Springer, 2022 (cit. on p. 106).

[158] A. Hazourli, *FinancialBERT - A Pretrained Language Model for Financial Text Mining*, (2022) (cit. on p. 107).

[159] D. Araci, *Finbert: Financial sentiment analysis with pre-trained language models*, arXiv preprint arXiv:1908.10063 (2019) (cit. on p. 107).

[160] P. Malo, A. Sinha, P. Korhonen, J. Wallenius and P. Takala, *Good debt or bad debt: Detecting semantic orientations in economic texts*, Journal of the Association for Information Science and Technology **65** (2014) (cit. on p. 107).

List of Figures

List of Tables