Unsupervised German Predicate Entailment Using The Distributional Inclusion Hypothesis

Sabine Weber

Doctor of Philosophy
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh
2022

Abstract

Recognizing textual entailment is an important prerequisite to many tasks in NLP, e.g. question answering and semantic parsing. Knowing that for example *buying* a thing entails subsequently *owning* it is a relation that humans learn by interacting with the world, while machines need other ways to acquire this knowledge. Previous approaches at learning predicate entailment relations from text have focused only on English. In this thesis we present the adaptation of the unsupervised entailment graph building algorithm of Hosseini et al. (2018) to German, which can be seen as a study of challenges in language adaptation for this task in general. We create a variety of German tools necessary for this approach and give a detailed account of the challenges faced and the insights gained from them.

First, we create a German relation extraction system and compare it against the English system presented by Hosseini et al. (2018). Finding that the typing of German entities constitutes a bottleneck, we create German fine-grained typing system for named and general entities. In doing so we examine the methods of annotation projection and zero-shot cross-lingual transfer, finding that for German fine-grained named entity typing zero-shot cross-lingual transfer performs best. We then move on to creating a German system that types general entities (e.g. “ex-president”) as well as named entities (e.g. “Obama”), by augmenting our training data with data automatically generated from a German WordNet (Hamp and Feldweg, 1997). We find that this way up to 10 percent points improvement in general entity typing performance can be reached, while only slightly impacting named entity typing performance by 1 percent point. We use these components in the pipeline to construct German entailment graphs.

We also present a method that uses German and English entailment graphs to generate training data for a supervised predicate entailment detection system, and show that this method outperforms current approaches at this task. This way we create a multilingual predicate entailment detection system, that outperforms both the monolingual German system and the zero-shot cross-lingual system on German test data, and also performs better than a monolingual English system on English test data.

Lay Summary

When humans convey information to each other, they can build on an existing shared understanding of the world. If someone was to say "John bought an apple." we would immediately assume that John owns the apple, that he can eat it, sell it or give it to another person. The relation between the word *buy* and *own* is called "entailment". We acquire knowledge of these relations by interacting with the world. Computers are not able to participate in our lives the way humans do and need other ways to acquire this information.

There are many different ways in which a computer can acquire knowledge: We can either write a program that provides it with set rules (so called rule-based approaches), we can provide it with a set of examples to learn from (so called supervised learning), or we can use known statistic properties of certain words to let a computer make decisions (so called unsupervised learning). This thesis examines the last kind of approach. We use the distributional inclusion hypothesis to induce entailment relations between predicates. This hypothesis states that more general words are seen in a wider variety of contexts and more precise words are seen in a smaller set of contexts. If the contexts of the more general word include the contexts of the more precise word, we assume the more precise word entails the more general word. For example the word *cat* entails the word *animal*, because *cat* can always be replaced with *animal* in a sentence without making the sentence incorrect, but not vice versa.

While this approach has been used successfully for English nouns and predicates, it has not been used for German words yet. In this thesis we apply it to German predicates for the first time, and present different ways in which the approach has to change to make it work for German. We also present different ways in which knowledge of predicate entailments in English and German can be combined to create a system that uses knowledge of entailments in one language to make correct predictions in the other language. We show that a combination of an English and a German system performs better than each monolingual system on its own.

Acknowledgements

I first want to thank my examiners Alexandra Birch and Anette Frank for their time and dedication to making this thesis the best it can be. I also want to thank my supervisors Mark Steedman and Mirella Lapata. Their guidance, patience and curiosity made this thesis possible. Rico Sennrich, Sharon Goldwater and Alexandra Birch helped me immensely as examiners on my yearly review committee. Next I want to thank the members of my research group, who were with me on this journey of entailment graphs, and whose advice, comments and code helped me a lot along the way: Liane Guillou, Miloš Stanojević, Thomas Kober, Javad Hosseini, Sander Bijl de Vroe, Ida Szubert, Nick McKenna, Tianyi Li and Liang Cheng. I also want to extend my thanks to the group members who do not work on entailment graphs, but nevertheless listened to me talk about them at every group meeting: Elizabeth Nielsen, Nikita Moghe, Matt Grenander, Tianyang Liu and Ratish Puduppully.

The Institute for Language, Cognition and Computation at the University of Edinburgh and in extension the whole Informatics Forum has been an incredible treasure trove of knowledge and people. I want to thank the organisers of the ILCC seminar series for countless invited talks and payed for lunches which got me into conversations with brilliant NLP researchers from all over the world. I also want to acknowledge all the other PhD students, post-docs and professors who spend some time with me sitting on sofas, sharing tea and solving all of the big and small problems that come your way as a beginning researcher (in no particular order): Adam Lopez, Naomi Saphra, Jonathan Mallison, Alexander Robertson, Mattias Appelgren, Esma Balkir, Toms Bergmanis, Arlene Casey, Carol Chermaz, Denis Emelin, Elaine Farrow, Seraphina Goldfarb-Tarrant, Zack Hodari, Craig Innes, Dilara Kekulluoglu, Kate McCurdy, Joana Ribeiro, Ramon Sanabria, Tom Sherborne, Mohammad Tahaei, David Wilmot, Wen Kokke, Rachel Bawden, Desmond Elliot, Yevgen Matusevych, Ulli German, Sorcha Gilroy, Federico Fancellu, Pippa Shoemark, Janie Sinclair, Borislav Ikonomov, Clara Vania and many others who's names I forgot. Two people especially kept me sane and happy after COVID put us all into home office: Andreas Grivas who was my daily accountability person, and Sameer Bansal who mentored me trough the rough waters of paper rejections and thesis writing.

When we talk about the shoulders of giants that we stand on, we think about people who wrote the books and papers that we cite, but rarely of the people who keep our offices clean and safe, our salaries transferred punctually every month and our travel expenses reimbursed. This is why I want to especially thank the administrative and custodial staff of the University of Edinburgh. Science can only happen because of them. I also want to highlight the members of the Staff Pride Network who have been working tirelessly to make Edinburgh University safe and welcoming for the LGBT+ community, especially Jonathan MacBride, Katie Nicoll Baines and Robert Court. Whether it was at protest marches in the pouring rain or at coffee and cake at the Bayes Cafe, they inspired me to stand up for who I am and what I believe. In the same vein I want to thank all the wonderful volunteers of Queer in AI, who keep pushing the boundaries of NLP conferences to make them more welcoming and accessible.

Lastly I want to thank my parents who patiently payed the bills while I was out collecting academic degrees, and my friend, flatmate, lockdown companion, sounding board, spellchecker and overall awesome human Andrew McLeod, who has been with me from the beginning of this PhD to its very end. I can't wait to see what we will do next.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Sabine Weber*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

When we read a sentence like “John bought an apple” we assume that John owns the apple, which enables him to do other things with it: he can eat it, bake it in a cake or sell it to someone else. This relation between the word “buying” and its consequences is called entailment. Entailment is directional: Owning an apple does not necessarily mean that one has bought it, but buying an apple always makes one the owner.

Knowledge of entailment enables us to make useful inferences. If someone asks us who has an apple, the knowledge that John has bought one would be enough to answer the question. While humans acquire this sort of knowledge about the world by interacting with it, machines need other methods to learn it. If someone was posing the query “Who owns Youtube?” to a search engine, it would be up to the algorithm to not only select answer sentences that contain the word “own” but also its paraphrases (e.g. “possess”) and relations that entail ownership, such as “bought”. In this thesis we offer one way in which machines can learn entailment relations. We use this knowledge to construct so called *entailment graphs*, in which the nodes are predicates (like “buy” and “own”) and the edges the entailment relation between them.

1.2 Language Independent Semantics

Entailments are mostly language independent. The German translation of the earlier example “buying” entails the German translations of “owning”. This is because entailments are grounded in extra-linguistic realities, e.g. the entailment in the former example it is grounded in how our societies define the concept of ownership.

In this thesis we look at entailment detection in English and German. Extracting entailments from more than one language and aligning them across languages can help to disambiguate expressions, because the same concept might be expressed in a more or in a less analytic way in either of the languages. Where English might use a frozen metaphor to express a certain concept, German might have a specific verb. Using both German and English data might improve performance in either language because the German training corpus might contain different entailments than the English one, amounting to more relations in total.

On the other hand, working with languages other than English poses unique challenges. NLP research has for a long time focused on English applications and the tools and methods available to us are affected by this. Even a high resource language like German falls behind English in terms of available test and training corpora and prior work. This thesis is an examination of the challenges and difficulties we face when adapting a method that works in English to another language. It can serve as a useful guideline to others who attempt to adapt our approach to different languages, and as a study on language adaptation in general.

1.3 Outlining Our Task in Contrast to Other Tasks

In this thesis we look at the task of extracting predicate entailments from German and English text. We want to distinguish our task from other, similar tasks like 1) natural language inference and sentence level entailment detection 2) common sense knowledge and reasoning and 3) hypernym detection.

The task of **natural language inference (NLI)** or sentence level entailment detection came into focus of the larger NLP community with the release of the data sets MNLI (Williams et al., 2018a) and XNLI (Conneau et al., 2018). These data sets contain entailing sentence pairs that span various genres and with XNLI, also various languages. Methods approaching these data sets treat the problem as a supervised learning task, often using a large language model combined with a classification layer (e.g. Liu et al. (2019) and Conneau and Lample (2019)).

This approach to entailment comes with several caveats. It requires large amounts of human labeled training data, which is expensive and time intensive to produce and limits the application of this method to new languages and domains. Other papers have also pointed out that the task of NLI is not well defined, and that human agreement on labels is low (Pavlick and Kwiatkowski, 2019; Chen et al., 2020b).

	Premise	Hypothesis	Label
MNLI	At the other end of Pennsylvania Avenue, people began to line up for a White House tour.	People formed a line at the end of Pennsylvania Avenue.	entailment
Levy-Holt	Griseofulvin,is indicated in the treatment of,infections	Griseofulvin,kills, infections	entailment

Table 1.1: MNLI contains long sentences, while the Levy-Holt data set contains minimal pairs. The sentences only differ in the predicate. Moeover, subject, predicate and object are annotated in the Levy-Holt data set

Another problem with this approach is explainability. The sentences contained within MNLI and similar data sets are relatively long. It is unclear which exact words or phrases trigger the entailment. Moreover these models constitute a black box which makes it difficult to trace biases that a model expresses to certain examples within the training data.

In contrast to this we take an unsupervised learning approach using the distributional inclusion hypothesis (see chapter 2). We treat the task at phrase level, determining the entailments of predicates in sentences. As a test corpus we do not use MNLI style data sets, but the Levy-Holt data set (Levy and Dagan, 2016a; Ricketts Holt et al., 2018) that contrasts minimal pairs of short sentences rather than the long and complicated sentences seen in MNLI (see Table 1.1). Another factor that adds to the explainability of our model is that entailments can be traced back to co-occurrence statistics. This way we can pinpoint incorrect entailments to instances in the corpus that we use.

Lastly, while models that use the language model plus classifier architecture can only label a sentence pair as entailed or non-entailed, the entailment graphs that we construct in this work can be used for both classification of sentence pairs and generation of sentence pairs. To make a judgement whether two predicates are entailed, we can check if they are entailed in the entailment graph. But we can also use the entailments contained within the entailment graph to generate or augment training data for tasks like question answering or link prediction in knowledge graphs (for more details, see chapter 7).

Some entailments between predicates could be framed as **common sense reasoning**, like the buying and owning example I gave earlier. But many of the relations expressed in common sense reasoning data sets are relations that are not expressed in human generated text because they are things so basic that humans don't state them explicitly, e.g. that one thing can only be at one place at a time. While data bases of these relations exist (Lenat et al., 1986; Havasi et al., 2007), our method depends on relations that can be found in large text corpora.

Lastly, our task is different from **hypernym detection**, as encoded in graphs like WordNet (Miller, 1995). Hyponym and hypernym are semantically similar, like "whisper" and "speak", where whispering is a type of speaking. All hypernym-hypomym pairs are entailments, but not all entailments express a hyponym relation, e.g. buying and owning are not semantically similar and buying is not a type of owning, but the former causes the latter. Another feature that distinguishes the entailments that we focus on, is the role of modifiers and negation. Some predicates can be negated without breaking the entailment relation to another predicate. For others, the negation of one predicate turns the entailment into a contradiction or a neutral statement. For example, "George learns about the game" entails "George knows about the game.". A negation of the first sentence causes a contradiction of the second sentence: "George didn't learn about the game" means "George doesn't know about the game." The case is different for the sentence "George won the game." which entails "George played the game.". The negation "George didn't win the game." does still entail "George played the game". Winning presupposes playing. Similar examples can be found for predicates that contain modifiers like "try to do something" or "manage to do something". These relations are captured in entailment graphs, but not in WordNet.

1.4 Thesis Structure and Contributions

The goal of this thesis is to create a German entailment graph and to use it in combination with the existing English entailment graph by Hosseini et al. (2018) to create a multilingual predicate entailment representation. This thesis is also a a study of language adaptation and the challenges faced when taking an approach from English to a different language. In the following chapters we walk through the necessary steps to extract predicate entailment relations from German text. We first address the theoretic foundations of our approach and previous work in this field, before we examine the single components of our pipeline in turn. We present our own models for open domain relation extraction and fine-grained entity typing as part of the German entailment graph pipeline. The result of our pipeline is the German entailment graph. We discuss ways to evaluate entailment graphs and we look at the possibilities of explicit alignment of the

German and English entailment graph. We then offer a method for creating a supervised multilingual entailment detection model using entailment graph relations as training data. Finally we offer avenues for future work regarding German and multilingual entailment graphs.

The contributions of this work are the following:

- a German Open Domain Relation Extraction system
- a German Fine-Grained Entity Typing model for named an general entities
- three German Fine-Grained Entity Typing test sets for named entities
- a German Fine-Grained Entity Typing test set for general entities
- a German predicate entailment graph
- a German translation of the Levy-Holt and SherLIic (Schmitt and Schütze, 2019) test sets
- a Multilingual Predicate Entailment Detection model

Work described in this thesis appeared in the following publications:

- *Construction and Alignment of Multilingual Entailment Graphs for Semantic Inference* (Weber and Steedman, 2019), describing our first approach at creating the German entailment graph and aligning German and English entailment graphs across languages.

- *Zero-Shot Cross-Lingual Transfer is a Hard Baseline to Beat in German Fine-Grained Entity Typing* (Weber and Steedman, 2021b), describing our work on German named entity typing.

- *Fine-grained General Entity Typing in German using GermaNet* (Weber and Steedman, 2021a), describing our work on German general entity typing.

- The code relating to German Fine-Grained Entity Typing can be found under https://github.com/webersab/german_general_entity_typing.

- The code of the Relation Extraction Pipeline can be found under https://github.com/webersab/relationExtractionPipeline.

Moreover we collaborated on the following publications:

- *Multilingual Entailment Graph Alignment Augmented by Cross-graph Guided Interaction* (Wu et al., 2021). The paper takes a different approach at aligning multilingual entailment graphs, using additional information from Wikipedia for alignment. We contributed our data in form of German entailment graphs and manually evaluated the main authors preliminary results. Their approach was strongly informed by the results presented in chapter 6, that show that additional information beyond the information contained in the entailment graphs themselves is necessary to align them across languages.

- *Cross-lingual Inference with A Chinese Entailment Graph* (Li et al., 2022). The paper presents a Chinese entailment graph and evaluates the results across languages using machine translation. We contributed work on creating a typing system for named and general entities in Chinese.

Chapter 2

Background

2.1 Introduction

In life we are often reminded to not judge things by what they occur with: Don't judge a book by it's cover or a wrist by it's Rolex. Harris' Distributional Hypothesis suggests the opposite for words: Words that occur within similar contexts have similar meanings (Harris, 1954). For example, seeing the words "cat" and "dog" used in similar sentences shows us that a shared set of properties applies to them: Both are four legged animals that we like to keep as pets.

If words are represented by vectors in a high dimensional vector space, the features that represent context can either be learned, like in Word2Vec (Mikolov et al., 2013) or be a direct numerical representation of the occurrence of a certain context. While approaches like Word2Vec only focus on semantic similarity and use bidirectionally similarity measures between word vectors, Geffet and Dagan (2005) examine the relation of lexical entailment by looking at the overlap between word vectors and computing a directional similarity measure. They introduce the distributional inclusion hypothesis and quantify the extent to which it holds for English nouns. Berant et al. (2011) and Hosseini et al. (2018) take this work from nouns to entailment in English predicates, and this thesis applies it to German predicates. The following sections gives a more detailed look at the theoretical foundations this thesis builds on.

2.2 The Distributional Inclusion Hypothesis for Nouns

In their 2005 paper "The Distributional Inclusion Hypotheses and Lexical Entailment" Geffet and Dagan define the term lexical entailment as a relation between two words that holds when “there are some contexts in which one of the words can be substituted by the other, such that the meaning of the original word can be inferred from the new one”. They name synonyms, hyponyms and meronyms as categories that fit this description. Their hypothesis is that more general words (e.g. “animal”) appear in a wider variety of different contexts, while more specific words (e.g. “cat”) appear in only a few specific contexts. When all of the contexts of the more specific word are included within all of the contexts of the more general word, they assume that there is a lexical entailment relation between the two words (e.g. “cat” entails “animal”). Representing the contexts of a word as features in a vector, they formulate two hypotheses: 1) if a word v entails a word w then we can expect all features of the vector representation of v to appear in the vector representation of w and 2) if all features of the vector representation of v occur in the vector representation of w, we expect v to entail w.

To test these hypotheses Geffet and Dagan select 200 English nouns and create a feature vector for each noun representing the context the noun appears in. They extract the context information from the 18 million word large Reuters corpus. The authors stress that the context is not a window of words occurring around the target word, but rather the words that attach to it in a dependency parse, e.g. modifiers to the noun like “company” in the noun phrase “the profit of the company”.

One problem in this approach is the lack of negative evidence. If a certain feature is not observed in the training data, this might be a correct negative signal, or a something that just does not happen to occur in the training corpus, but could be observed outside of it. To overcome this problem, Geffet and Dagan perform a web search for each of the word pairs that make up a feature. Because this approach is time intensive, they don’t web search all of the missing features, but only randomly sample 20 of them. They retrieve up to 3000 additional sentences from the web for each triple of two nouns and a feature, and include the counts from these sentences into the feature vectors.

The value of a feature within a feature vector is determined by point-wise mutual information. They compare different metrics for the calculation of the overlap between vectors and propose a new weighting schema for the task. They then move on to test whether feature inclusion of the vectors correlates with lexical entailment judgements of humans. Their results show a strong correction between feature inclusion and human judgement. In 86% of all the hypernym pairs they tested there was an overlap between the context vectors and 70% of all vectors that had an overlap were actually hypernym pairs.

2.3 Application for Verbs

The experiments of Geffet and Dagan (2005) to prove the distributional inclusion hypothesis for English nouns have not been repeated for verbs. While Geffet and Dagan talk about words in general in their paper, verbs have different statistical properties from nouns. In English, they are much rarer than nouns (there are approximately 60 000 verbs in English and about 550 000 nouns), which means that they are more likely to occur in a wider range of different contexts than nouns. Moreover there are so called light verbs (e.g. give, take, have) that are very common and express very different concepts depending on the nouns they combine with. This has implications for the feature vectors used to represent verbs: While Geffet and Dagan could construct meaningful representations from 200 features, vectors for verbs have to include more context and are necessarily larger and sparser. It is also unclear how many lexical entailment relations exist between verbs. While WordNet classifies every noun as “entity” or “concept”, there are not such overarching categories for verbs, and often less of a clear hierarchy.

Berant et al. (2011) and Hosseini et al. (2018) apply the previous approach to the case of verbs, but their approach differs from Geffet and Dagan approach in several points:

1) Berant et al. (2011) and Hosseini et al. (2018) use higher dimensional vectors (1000 features as compared to the 200 features that Geffet and Dagan used for nouns) to represent verb contexts. They show that both verb hypernyms (e.g. “whisper” is a type of “speak”) and preconditions and consequences (e.g. “win” is an outcome of “play”) can be extracted from larger corpora of text. The latter type of lexical entailment does not occur with nouns, that only have lexical entailments that are synonyms, hyponyms and meronyms.

2) Following Geffet and Dagan’s use of dependency parsing to determine a verbs context Berant et al. (2011) consider the subject and the object of a verb as the verb’s context. They stress the importance of this context for the detection of entailment relations, because one verb might entail another verb in one context, but contradict it in a different context. One example for this is the sentence “Antibiotics eradicate smallpox”. The reader concludes that in this case antibiotics cure smallpox. But when we read a sentence stating that antibiotics eradicate a person it’s the opposite of antibiotics curing them. Therefore only verbs that share the same context can be considered for entailment relations.

Considering only cases in which two verbs share exactly the same subject and object would drastically narrow down the number of sentences that could be considered. To mitigate this problem Berant et al. (2011) suggest determining a more general description for the specific subjects and objects, the so called type: e.g. antibiotics are a medicine, smallpox a disease and so on. Entailment relations only hold between verbs, if they occur with the same type pair.

3) Geffet and Dagan address the problem of lacking negative evidence by conducting a web search for pairs of nouns and features. This way they try to make sure that a lack of occurrence for a certain combination is genuine and not due to sparsity of the corpus. Neither Berant et al. (2011) nor Hosseini et al. (2018) do this additional step.

4) Unlike Geffet and Dagan neither Berant et al. (2011) nor Hosseini et al. (2018) quantify how strong the correlation between context inclusion and lexical entailment is for verbs. Therefore it remains unclear whether sparsity and noise are due to shortcomings of the tools used for parsing and data extraction or due to a weakness of correlation between feature inclusion and lexical entailment for verbs.

One way to formalize entailment relations between verbs is to represent the verbs as nodes and the entailment relations as the edges of a graphs. We refer to these graphs as *entailment graphs*. Because only verbs that go with the same type pairs entail each other, each type pair gets its own entailment graph. An example of the the entailment graph for the type pair “living thing” - “medicine” is shown in Figure 2.1.

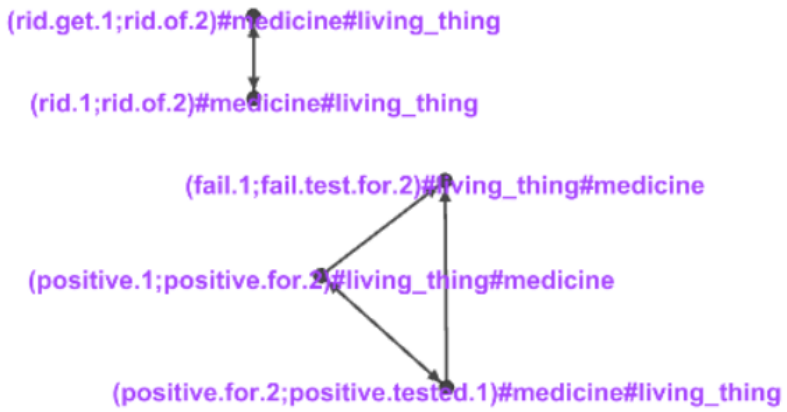

(rid.get.1;rid.of.2)#medicine#living_thing
(rid.1;rid.of.2)#medicine#living_thing
(fail.1;fail.test.for.2)#living_thing#medicine
(positive.1;positive.for.2)#living_thing#medicine
(positive.for.2;positive.tested.1)#medicine#living_thing

Figure 2.1: An example of the English entailment graph for the type pair “living thing” - “medicine”. Double-headed arrows indicate paraphrases, single-headed arrows indicate entailment. While the entailment graph building algorithm considers “being positive for” and “being tested positive for” as paraphrases, “fail a test for” is entailed by both.

Since the introduction of typed entailment graphs by Berant et al. (2011), Hosseini et al. (2018) advanced the state of the art by adapting the algorithm to process larger amounts of data and by adding global soft constraints to learn entailment relations across different typed entailment graphs. This way entailments that might have been observed for one type pair can be transferred to a different type pair, if contexts are similar, but no spurious new entailment links are introduced.

Recently, Schmitt and Schütze (2021a) reframed the problem of predicate entailment detection as a supervised learning problem. Schmitt and Schütze (2021a) use Hearst patterns to train a classification model that builds upon the contextualised word embeddings derived from the language model RoBERTa (Liu et al., 2019). We discuss the specific strengths and weaknesses of this approach in more detail in chapter 7.

2.4 Beyond English

So far there has been no work on predicate entailment in languages other than English. Lewis and Steedman (2013b) expand a similar distributional semantics approach as Berant et al. (2011) to English and French, but instead of building entailment graphs they only cluster similar predicates, creating a bidirectional representation. They then align the resulting clusters across languages.

In their approach Lewis and Steedman (2013b) construct vectors representing the predicates by extracting the subject and object a predicate occurs with, just as Hosseini et al. (2018) do for the construction of entailment graphs. They then link the subjects and objects of the predicate to a database. Because the named entities in different languages are linked to the same database, the predicate vectors are part of the same vector space and can be aligned using measures like cosine similarity. Because Lewis and Steedman (2013b) use bidirectional measures of similarity, both paraphrases and entailments are included in their paraphrase clusters.

In their work Lewis and Steedman (2013b) use Wikipedia articles in English and French describing the same topic. This can be considered a form of parallel text. Even though there is no alignment between sentences, the fact that the same topics are covered is helpful for the construction of paraphrase clusters, as translations of predicates are likely to occur in the text. They use the inter-language links between named entities provided in the Wikipedia articles and the types that were provided with the Wikipedia links. This way, they could work with perfect typing and alignment of named entities to derive the alignment of predicates. Working with data that is less annotated poses new challenges. Linking German named entities to an external data base like Freebase (Bollacker et al., 2008) or DBPedia (Lehmann et al., 2015) is often noisy and incomplete, and only a partial overlap between named entities in parallel text can be achieved. In preliminary experiments, only about 20% of all named entities in a German news corpus can be linked to DBPedia. Therefore, a different approach for alignment than the one used by Lewis and Steedman (2013b) is needed. In chapter 6 we discuss a method using the alignment of monolingual word embeddings, and offer an error analysis detailing the difficulties with this approach.

Another approach is to forgo explicit alignment of German and English entailment graphs, but instead use both German and English entailment pairs from monolingual entailment graphs as training data for a multilingual entailment detection model. We use this approach to construct a multilingual predicate entailment detection system in section 7, and talk about its background in the following section.

2.5 Language Model Based Approaches

The previous sections focused on the distributional inclusion hypothesis and predicate entailment detection as a standalone task to be performed by a model specifically created for this purpose. The recent emergence of contextualised word embeddings (also called foundational models) has opened up a different approach to NLP tasks. Rather than creating standalone systems or pipelines for a specific task, it has become common practice to use large self-supervised models that are trained on unstructured data and to fine-tune them on a proportionally small amount of task specific data. This approach has also been applied to the task of predicate entailment detection, and we employ it in chapter 7 to create a multilingual predicate entailment detection system. In this chapter we will cover the background of this approach.

Word Embeddings The count based vectors that we use to calculate entailment scores between predicates can be seen as a precursor to the word embeddings introduced by Mikolov et al. (2013). Rather than representing actual counts of specific contexts of a word, the features of the word vectors of Mikolov et al. (2013) are learned by the model using a word prediction task. The resulting vectors have proven to perform better in word similarity tasks than previous count based methods and have been successfully used in downstream applications. One of the shortcomings of this word representation is that each word is represented by a vector regardless of the context it appears in, leading to inaccurate representations for polysemous words.

The next step in the development of word vector representations are the so called contextualised word embeddings. Using them, a word receives a different representation based on the sentence context it appears in. The training of these models jointly conditions on both left and right context of any word. Contextualised word embeddings use a masked language model objective, where words from the input are randomly masked and the model has the task to predict the masked word. The second objective is the task of next sentence prediction. These models also use the Transformer architecture and large amounts of training data, profiting from the widespread availability of GPUs for computation.

The release of BERT (Devlin et al., 2019a), RoBERTa (Liu et al., 2019) and XML-RoBERTa (Conneau et al., 2019) marks a shift from language modelling as a sub-field of NLP to language modelling as a prerequisite for many different NLP tasks. While the pre-training of the large language models requires large data and computational resources, the subsequent fine-tuning can be done at a smaller scale. The transfer learning ability of large language model enables high performance on a variety of different tasks, with only relatively little annotated training data needed (e.g. the 100K sentence pairs of the SQuAD data set for fine-tuning, compared to all of English Wikipedia and the BookCorpus for pre-training (Devlin et al., 2019a)).

Prompt Based Learning The high performance and transfer learning ability of contextualised word embedding models like BERT and RoBERTa opened up the question whether linguistic knowledge and world knowledge is encoded in them, and if so, how to best access that knowledge. Subsequent approaches strive to make the intrinsic knowledge of the language model extrinsic by giving it the best suited fine-tuning examples. Early approaches used task-based fine-tuning data sets without alteration. In the case of sentence level entailment detection this meant concatenating the premise and hypothesis sentence and set the objective to predict the correct label (Devlin et al., 2019a). While not as data-hungry as pre-training, these approaches still required a relatively large amount of human-annotated task-specific data (e.g. the 112,500 sentence pairs of the XNLI data set (Conneau et al., 2018)). This opened up the field of few-shot-learning, where a model is only given few specially selected instances for fine-tuning (e.g. only 4500 sentences of the Levy Holt data set, as used by Schmitt and Schütze (2021a)).

At this point we need to distinguish between sentence level entailment detection (NLI) and the task of predicate entailment detection, which is at the center of this thesis. While NLI is used as a way to evaluate the quality of large contextualised

	Premise	Hypothesis	Label
NLI	At the other end of Pennsylvania Avenue, people began to line up for a White House tour.	People formed a line at the end of Pennsylvania Avenue.	entailment
Levy-Holt	Griseofulvin,is indicated in the treatment of,infections	Griseofulvin,kills, infections	entailment

Table 2.1: NLI contains long sentences, while the Levy-Holt data set contains minimal pairs. The sentences only differ in the predicate. Moeover, subject, predicate and object are annotated in the Levy-Holt data set. NLI examples were taken from Williams et al. (2018b).

word embeddings (e.g. as part of the GLUE benchmark (Wang et al., 2018)), the task of predicate entailment detection has so far received less attention. While NLI is a task at sentence level, predicate entailment detection operates at phrase level. Predicate entailment data sets contrast minimal sentence pairs that only differ in their predicate, while NLI data sets often have longer and complicated sentences with no clear distinction of which part of the sentence triggers the entailment. Examples can be seen in table 2.1. Lastly, while NLI data sets are available for different domains and languages (e.g. MNLI (Williams et al., 2018b), XNLI (Conneau et al., 2018) and QNLI (Wang et al., 2018)) up until the introduction of the SherLiIC data set (Schmitt and Schütze, 2019) the Levy Holt data set (Levy and Dagan, 2016a) was the only predicate data set available. Using machine translation we translated these data sets to German for our German and multilingual experiments. NLI data sets and predicate entailment data sets are also set apart by their size, with either of the two predicate entailment data sets (Levy-Holt and SherLiIC) considered to small to effectively fine-tune a large language model.

This is where prompt-based learning comes in. Rather than relying on examples alone to fine-tune a model, the approach attempts to linguistically encode the task in the examples as well (Radford et al., 2019). For the task of entailment prediction this can mean that rather than simply concatenating premise and hypothesis in the training data, the two sentences are connected by words that make the entailment relation explicit, e.g. “hypothesis *because* premise”. After first using a variant of this approach in for the task of text calssification (Schick and Schütze, 2021a), Schmitt and Schütze (2021a) apply this method to the task of predicate entailment detection, which they also call *lexical entailment in context*. In their paper they explicitly compare their approach to the unsupervised distributional inclusion approach of (Hosseini et al., 2018) and reach a large improvement. Owing to the supervised learning nature of the language model fine-tuning approach they use a small subset of the Levy-Holt data set as training data. They show strong results when testing on a different part of the Levy-Holt data set, but weaker performance when testing on a previously unseen data set. We address this shortcoming in chapter 7, where we train their system on data generated from the German and English entailment graphs and achieve strong results across two unseen data sets.

Criticism of Prompt Based Learning The prompt based approach has been used in a variety of different NLP tasks since (for a survey see for example Liu et al. (2021)). The approach has also been criticised by Webson and Pavlick (2021), stating that there is little evidence that prompts that encode the task verbally perform better than neutral or intentionally misleading prompts. They also find that prompt based models are overly sensitive to the targets that a judgement of entailment is mapped to, e.g. mapping two entailed sentences to the target “yes” and two non-entailed sentences to the target “no”. Their experimental setup does not translate well to the setup we use in chapter 7, because we are not mapping the judgement of entailment to verbalised targets but rather to a number between 0 and 1. It therefore is a avenue for future work to see how varying the prompt text would influence the models performance in our specific use-case.

Another criticism of the prompt-based approach was voiced by Perez et al. (2021). The authors question the performance of the approach in what they call a “true” few-shot scenario where no development set for fine-tuning is available. They stress the sensitivity of prompt based learning to hyperparameter fine-tuning, prompt phrasing, model architecture and decoding strategies. Our own findings partially support this criticism, showing high sensitivity to hyperparameters, which leads us to train 50 models with different hyperparameter configurations to achieve results within 5% of what Schmitt and Schütze (2021a) report.

Schick and Schütze (2021b) address this criticism by revisiting the task of text classification an the model presented in Schick and Schütze (2021a). Because of the different task and model it is unclear how well their results are applicable to the work we present in chapter 7. They focus mostly on the performance of prompts and show that using prompts leads to better performance than not using them, but they do not include neutral or intentionally misleading prompts, which hinders a comparison with Webson and Pavlick (2021). While they state that best performing prompts stay the same when switching language model, we find that hyperparameters do not translate well across different language models. Using the hyperparameters that work best with the English language model RoBERTa for training with the multilingual language model XLM-RoBERTa leads to a massive decrease in performance.

As a next step in the prompt base learning approach to predicate entailment Schmitt and Schütze (2021b) replace human generated prompts with prompts that are learned by the system. These prompts do not have a human understandable linguistic counterpart, but are rather vectors in the word embedding space that express the desired task relation. This builds upon the assumption that the relation of predicate entailment is implicitly contained within the language model, similar to the commonly cited and later criticised analogy tasks in non-contextualised Word2Vec style word embeddings (Rogers et al., 2017). While Schmitt and Schütze (2021b) improve over their earlier results with explicit prompts, they still train on a subset of the Levy-Holt data set which makes it likely that the model picks up annotation biases. It is this specific caveat that we address with our approach, which we will cover in more detail in chapter 7. Generating training data from the entailment graphs rather than relying on the small human-annotated Levy-Holt data set allows the model to learn entailment information without picking up annotation biases. Moreover this opens up possibilities for training on languages that have entailment graphs, but no human-annotated predicate entailment data sets.

2.6 Conclusion

In this section we laid out the background of the methods presented in this thesis. We explained the usage of the distributional inclusion hypothesis for entailment detection in nouns and the alterations taken to make this approach viable for predicates, with the final goal of predicate entailment graph construction. We then revisited the literature on predicate entailment approaches for languages other than English, finding that creating a multilingual predicate entailment representation from non-parallel text poses a unique set of challenges.

We then contrasted this unsupervised approach of entailment graph building with a supervised language model based approach for predicate entailment detection. While the task of sentence level entailment detection has received much attention as part of benchmark data sets for language models, the task of predicate entailment detection is only addressed by few and small data sets, which makes it more suited for the few-shot learning paradigm. We introduced the prompt based approach to few-shot learning, and laid out the criticism this approach received since its inception. While we find hyperparameter instability to be an important caveat of the approach, we manage to mitigate the instability introduced by small human-annotated training data sets. By replacing training on the human-annotated Levy-Holt data set with training on data generated from the English and German entailment graphs we achieve consistent results across two unseen data sets. We describe this approach in more detail in chapter 7.

Chapter 3

The Relation Extraction Pipeline

3.1 Introduction

Our goal is to find the entailments between predicates using the distributional inclusion hypothesis. The representation of predicates in our system depends on the context that they appear in. In contrast to Word2Vec style word embeddings (Mikolov et al., 2013) and contextualised word embeddings (Peters et al., 2018; Devlin et al., 2019b), our representation does not formalize context as words appearing next to a specific word, but instead formalizes context as the words that attach to it in a dependency parse. Berant et al. (2011) have shown that the subject and object of a predicate are useful features when using the distributional inclusion hypothesis for entailment detection.

Our work is concerned with the entailments between predicates. This means that we not only encounter predicates as single words, but also as larger constructs, e.g. light verb constructs ("Paul *has an appointment* with Dave."), modifier constructs ("Paul *failed to buy* the car.") and frozen metaphor ("Paul *held a speech* in front of the the audience."). We also consider negation ("Paul *did not meet* Dave."). This expands the amount of predicates the resulting entailment graphs have: For example, in addition to the ca 55 000 verbs of the English language we potentially have occurrences of the constructs shown above. The predicates "fail", "fail to buy" and "not fail to buy" are represented as different, independent instances and receive different vector representations.

The following chapter describes how subject-predicate-object triples are extracted from text, to later be used as features in the vector representation of predicates. Because our ultimate goal is a multilingual representation, we extract them from German and English text. While we rely on the work of Hosseini et al. (2018) for English relation extraction, we create our own system for German. The input to this part of our system is a natural language sentence like “Die CDU untersucht jetzt die Zahlungen und hat den Bundestag informiert .”[1], the output is the subject-predicate-object triple in the format “(untersuchen.1,untersuchen.2) #government#event ::CDU::Zahlungen |||(passive: False)”[2]. In the following chapters we will look into the different parts of the German relation extraction system in more detail.

3.2 Corpora

We extract the subject-predicate-object triples (form here on referred to as *relation triples*) from large corpora of news text. We make this choice for the following reasons: First, news text is widely available and can be easily scraped from the internet. Second, it covers the same topics (politics, economy, sports etc.) over a long period of time, which ensures that the parties discussed (e.g. politicians, countries and sports teams) occur multiple times across different documents. This makes it more likely that different predicates are mentioned in the same context, e.g. when a politician runs for office, gets elected and then acts in their governmental role. This enables us to find overlaps between contexts, which is important for our directional similarity metric to work (see chapter 2 for more details). Third, one event might be discussed by several different news outlets in varying degrees of detail, which constitutes a good source for paraphrases and entailments, e.g. when one news source describes a company takeover as one company buying the other, while a second news source might use the words “purchase” or “aquire”.

The work discussed in this thesis uses two different news corpora. Hosseini et al. (2018), who’s system we use for English relation extraction and entailment graph construction, use the NewsSpike corpus (Zhang and Weld, 2013). This corpus was collected from January 1 to February 22, 2013 by subscribing to newspaper RSS feeds and collecting RSS news seeds, which contain the title, time-stamp, and abstract of the news items. The authors then used the news titles to search for the news article on the internet. The authors do not state which newspapers they collected the data from. They collected approximately 500 thousand articles this way.

For the German and a larger English entailment graph we use the unpublished NewsCrawl corpus (a corpus of a similar name, but different contents was published as

[1] Tranlsation: The CDU examines the payments now and informed the Bundestag.

[2] Tranlsation: (examine.1,examine.2) #government#event ::CDU::payments |||(passive: False)

part of the WMT corpus by Barrault et al. (2019)). This corpus was gathered by the Machine Translation group at Edinburgh University since 2008 and contains a wide variety of languages beside English and German. The news sources were set up in two ways: 1) by manually subscribing to RSS feeds, 2) by automatically crawling for and subscribing to RSS feeds. Liane Guillou performed automatic language detection to filter out cases in which the language of the article was different from the language label of the news source. Thomas Kober performed deduplication to filter out the instances where the same article was published by different news sources.

The German part of the NewsCrawl corpus contains 276 different news sources, ranging from nationwide publications like “Bild” and “Zeit” to local newspapers like “Ahlener Zeitung”. Notably, it also includes German publications from outside of Germany, e.g. from Austria, Luxembourg and Liechtenstein. While these news sources might introduce variations in spelling and word choice, we do not examine the implications of these variations in the context of entailment graphs.

There are a few important differences between the English and the German parts of the NewsCrawl corpus that lead to complications in the construction of entailment graphs. First, there are only 62 English news sources, compared to the 276 German ones. Second, the English sources span a larger geographical area, covering the US, UK and Australia and English language outlets of newspapers in China, Russia and Bulgaria. Most importantly, the English news sources are large newspapers and press agencies like Reuters and AP, while smaller publications are completely missing, whereas in the German part of the corpus smaller publications make up the bulk of the news sources. This leads to very different events and entities being discussed in the German and English part of the same corpus: While both parts cover world politic and economy, there is a large amount of German text that deals with local events. We shall return to this point when discussing problems with named entity recognition and typing in German, and problems with aligning German and English entailment graphs.

In the following sections we will first look at related work in English relation extraction before outlining Hosseini et al. (2018)’s approach. We then look at work relating to German relation extraction and our own system.

3.3 English Relation Extraction

3.3.1 Related Work

The problem of extracting subject-predicate-object triples can be framed as open domain relation extraction (also referred to as open information extraction or *openIE*). While closed domain relation extraction aims to match a relation in a preexisting data base schema to its expression in text, open domain relation extraction does not operate with a preset number of relations. Rather, the relations are defined by their expression in the text that they are extracted from.

Previous work treats the task as an unsupervised learning problem. Early systems were rule based and operated on dependency parses (Stanovsky et al., 2016). In recent years neural systems have advanced the field, using supervised learning approaches and building on contextualised word embeddings (a more detailed time line of open domain relation extraction systems can be found in the paper by Zhan and Zhao (2020)).

Currently there are two main neural approaches to the problem: either labelling sequences of the text as subject, relation and object, or generating the extraction one word at a time. Stanovsky and Dagan (2016) use the labelling approach, using a semantic role labeling corpus adapted for OpenIE as training data. Cui et al. (2018) apply the generation approach and use the output of older, rule based open domain relation extraction systems as training data. While the labelling approach is relatively fast, it is less accurate than the generation approach. The generation approach re-encodes the previous output when generating extractions, and therefore captures dependencies between extractions. This leads to less redundant output than the labelling approach, but also makes it slower.

Kolluru et al. (2020)'s state of the art system combines both approaches. They set a maximum number of extractions per sentence and model the sentence as a gird of the size token number times maximum extraction length. In this grid the system can label tokens as either subject, relation, object or none. The authors use a set of linguistically motivated constraints to limit what their grid labeling approach is allowed to produce, e.g. making sure that all nouns and verbs are part of a prediction, that each relation contains only one verb, and that each verb must be part of one extraction.

Despite these advances, the discussed models are ill suited for the task of relation extraction with the goal of entailment graph building. First, these models do not distinguish between nouns, noun phrases and whole sub-clauses as the subjects and objects of predicates. Our directional similarity metric relies on different predicates

Sentence	First Boston incurred millions of dollars of losses on Campeau securities it owned as well as on special securities it couldn't sell.
OpenIE extraction	First Boston; *incurred*; millions of dollars of losses on Campeau securities it owned as well as on special securities it couldn't sell it; *owned*; Campeau securities it; *couldn't sell*; special securities
entailment graph relation triple	First Boston; *incur*; losses First Boston; *incur_on*; securities

Table 3.1: While the extractions of an open domain relation extraction system retain more information and are more easily readable for humans, the triples required for entailment graph building are constrained to only one or two nouns per subject and object. Also, the predicate is lemmatized to create a unified representation for predicates independent of tense and number. To make typing possible, triples with pronouns as subjects or objects are discarded.

occurring with the same subject and object, which is why too much variation in subjects and objects is counterproductive. This variation also complicates the typing of subjects and objects, which is essential for our metric. Second, the output of state-of-the-art open relation extraction systems does not contain any explicit information about negation or modifiers, which are treated as part of the relation by the systems. Obtaining this information, which is necessary for our entailment detection approach, would require substantial post-processing. An example taken from the Re-OIE2016 corpus (Zhan and Zhao, 2020) illustrating these differences can be seen in Table 3.1. This is why we choose to follow a rule-based approach instead.

3.3.2 Our Pipeline

For English relation extraction we use the system of Hosseini et al. (2018). It consists of several steps: annotating named entities, parsing the sentences that contain these named entities, finding predicates in the parses that are connected to the named entities and finally linking named entities to a database and deriving their type from that link.

For named entity recognition, Hosseini et al. (2018) use the Stanford NER component (Finkel et al., 2005). Whenever there are two named entities in a sentence they parse it using the CCG semantic parser of Reddy et al. (2014). They only consider binary relations (e.g. “Obama visits Hawaii”, but not “ Obama visits Hawaii on Monday”). If a ternary case like this occurs in the text it will be split up into two relation triples: “Obama; visit; Hawaii”, “Obama; visit_on; Monday”. The parse also contains information about passivization, negation and the use of modifiers. The predicate is lemmatized and event modifiers are attached to the main verb (e.g. the predicate in “Obama planned to visit” will be extracted as “visit_plan”).

The next step in the pipeline is typing, for which Hosseini et al. (2018) use AIDA-light (Nguyen et al., 2014). AIDA-light links a named entity to its Wikipedia entry. Then, the URL of the Wikipedia entry is mapped to its freebase entity. The type of the freebase entity is then mapped to FIGER types (Ling and Weld, 2012). Only the first level if the FIGER type hierarchy is used, a total of 49 of 112 types. Because entailment graphs can only be constructed for type pairs, the usage of 49 distinct types keeps the graphs from becoming too small, but important semantic distinctions are still preserved. Types are, for example, “person” or “location”. If an entity is not found in Freebase, the type “thing” is assigned.

3.4 German Relation Extraction

3.4.1 Related Work

In recent years, there have been few approaches to open domain relation extraction in German. There is no freely available annotated data set for open domain relation extraction in German, which limits the research in that area. Current approaches use parsing and static rules. Falke et al. (2016) transfer the rule set of an English rule based open domain relation extraction system to German. Due to the lack of a human annotated German test set, they evaluate their results by manually annotating errors in the system output. Bergmann et al. (2016) use a similar rule-based approach by adding German-specific rules to the ReVerb system (Fader et al., 2011), but they offer no evaluation or error analysis.

There have been several approaches treating open domain relation extraction as a supervised learning problem in a multilingual context, trying to leverage the knowledge a model learns from an English training data to other languages where only little or no training data is available. Harting et al. (2020) use a word2vec style multilingual embedding space to do open domain relation extraction in English and Dutch. Ro et al.

(2020) take a similar route for English, Spanish and Portuguese but use contextualised multilingual word embedding instead. Both approaches create their own test corpora either via manual annotation or via machine translation of existing English corpora. Soares et al. (2019) offer an interesting approach to English relation extraction that does not rely on an annotated training corpus, but only on linked named entities, that could be obtained from an automated NER and named entity linking component. One shortcoming of their approach is the assumption there is only one possible relation between any two named entities. Nevertheless, this approach could also be adapted for German.

The same shortcomings mentioned earlier in context of English open domain relation extraction apply here as well: The output of the mentioned systems differs too much from the format needed for our entailment metric. Additionally, multilingual system perform worse than their English counterparts, which would add noise to the necessary post-processing steps. This is why we choose to build a German rule-based system that is tailored to the requirements of our use case.

3.4.2 Our Pipeline

The German relation extraction pipeline mimics the English relation extraction system of Hosseini et al. (2018). This ensures that output of the German relation extraction system is as similar as possible to the output of the English relation extraction, which enables us to use Hosseini et al. (2018)'s code for the construction of entailment graphs from relation triples for both English and German. Most of the work on the German relation extraction pipeline was done in cooperation with Liane Guillou.

The first step of the German relation extraction pipeline is named entity recognition using the German NER model of FLAIR (Akbik et al., 2018). Preliminary experiments have shown that considering only named entities for the relation triples leads to too few extractions to build a meaningful representation of predicates (we will discuss this in more detail in section 3.4.3). This is why we decided to not only extract named entities (like "Angela Merkel"), but any noun that might occur with a predicate (like "chancellor"). We call these *general entities* as opposed to the *named entities* detected by a named entity detection system.

The English relation extraction pipeline uses linking to Wikipedia as a way to type named entities and general entities. Named entity linkers like DBpedia-Spotlight (Daiber et al., 2013) and AGDISTIS (Moussallem et al., 2017) come with the option to link named entities from German text, but not general entities. Linking poses a major bottleneck of the German pipeline, because AGDISTIS is slow and introduces noise in the form of wrong links. Instead of using these tools, we develop a typing system for German named and general entities, which we discuss in more detail in chapter 4.

Due to the lack of a German CCG parser we use the Stanza dependency parser (Qi et al., 2020) and a set of handcrafted rules. One way to extract a predicate that expresses the relation between two entities is to find the shortest path between the entities in the dependency tree and extract the verbs that lie on this path (see the example in Figure 3.1). Using dependency parses of German poses additional difficulties that are addressed by hand crafted rules, e.g. discarding the triple if the path is too long or if the path contains illegal labels.

The extraction of a predicate for any given set of two entities consists of the following steps, where at each step the pair of entities is discarded if the criteria are not fulfilled: First we check the dependency links attached to the entities. The subject entity must have one of the dependency parse links ‘nsubj’, ‘nsubj:pass’ or ‘dep’, while the object entity must have either ‘obj’, ‘obl’ or ‘dep’. Next we check if both entities are connected to the same head, which becomes the candidate for the predicate. At this point we check if the candidate predicate is part of a light verb construction, by looking up the entities and the candidate predicate in a list of German light verb expressions (Krenn, 2000). We additionally check for constructions with the light verbs ‘sein’ (tranlsation: to be) and ‘haben’ (translation: to have), like for example in the sentence “John has a meeting with Paul”, where the desired relation would be “to have a meeting with” between the entities “John” and “Paul”. After this we check the dependency parse for negation and modifiers for the predicate. Because negation of the noun is more common in German than in English (e.g. “John has no meeting” as opposed to “John doesn’t have a meeting”) we check for that, too. After we collected this information, we assemble the extracted entities and predicates to the format shown earlier (chapter 3.1). A major shortcoming of this approach is the lack of a development data set to fine-tune the rules and a test data set to evaluate the results. We address these points in more detail in the following section.

3.4.3 Error Analysis

The architecture of the German relation extraction pipeline builds on the assumption that parsing the German input with an universal dependency parser and applying a set

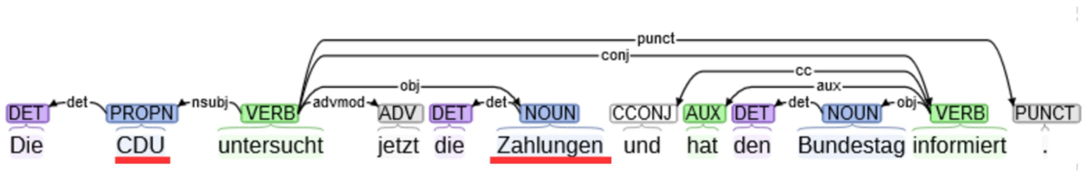


Figure 3.1: The figure shows the dependency parse tree of the sentence "Die CDU untersucht jetzt die Zahlungen und hat den Bundestag informiert." (*Translation: The CDU examines the payments now and informed the Bundestag.*) The shortest path between the two highlighted entities "CDU" and "Zahlungen" leads trough the predicate "untersucht". Our system therefore extracts the relation triple *CDU; untersuchen; Zahlungen*

of fixed rules in post-processing will yield German relations comparable to the ones yielded by the English system. This assumptions has been proven wrong to some degree: Running the English and the German relation extraction systems on parallel text shows that the relations yielded are both quantitatively and qualitatively different.

A close error analysis of 5 randomly selected parallel documents from the News-Commentary corpus (Freitag et al., 2021) (English new articles translated to German) shows that the German relation extraction, if using only named entities as subjects and objects, retrieves on average 15 relation triples per document, whereas the English relation extraction retrieves on average 85 relation triples per document. Only when the German relation extraction is taking general entities into account as well, the number of extracted relations becomes more similar. This under-prediction can be attributed to several factors: While the German NER system might be weaker than the English one due to less available German training data, different sentence structure in German might make it harder to find predicates on the shortest path between to entities in a dependency parse tree.

In addition to this quantitative difference, the assumption that English and German named entity recognition is of similar quality holds only in part. The different systems detect different named entity boundaries. While the English system might detect the named entity "Jeff Bezos", the German might just detect "Jeff" which leads to problems in typing the entities.

There are certain types of relations that are only extracted in English, but never in German e.g. "of", "for", "with", "'s" and time expressions. Moreover, these types of non-predicate relations make up the majority of the 40 most common extracted predicates in English. This is partly due to the usage of a CCG parser in English and a universal dependency parser in German, but also due to the differences in language.

For example, the equivalent of the English possessive "'s" and "of" is translated into German as a genitive construct and therefore harder to detect in a dependency parse. The extraction of time expressions is an additional problem that could only be handled by adding a specific time expression recognizing component to the pipeline (e.g. Heideltime by Strötgen and Gertz (2010)), which in turn might add noise.

There is a wide gap between the performance of the English and German relation extraction: a manual analysis shows that 34% of the English retrieved relations are correct, whereas only 16% of the retrieved German relations are correct. This in and of itself is problematic, as it points to the overall low quality of the relations that make up the basis of entailment graphs. In addition to the implications for the quality of German entailment graphs, these shortcomings have consequences for the construction of multilingual entailment graphs, that should ideally combine relations from the German and English entailment graphs into one representations. We address this in more detail in chapter 6.

3.5 Conclusion

In this chapter we presented the German relation extraction pipeline. To the best of our knowledge there is no German relation extraction system that fits the demands of downstream entailment graph construction, which is why we created our own. It is one of the main contributions of this thesis. We follow a pipeline approach that combines parsing, NER and entity typing and then uses static rules and a data base of common light verb expressions to extract relation triples from a German sentence. We find that while this approach succeeds in providing the building material for the later steps of entailment graph construction, there are several caveats to it.

This is why we offer a detailed error analysis of our system in comparison to the English relation extraction system used by Hosseini et al. (2018) as a precursor to English entailment graph building. We find that in both systems only a small part of all extractions are correct. Specific types of relations (prepositions, time expressions) are missing from the German system and adding them in could create another source of noise. This points to an open question in German open domain relation extraction: On the one hand, the rule-based pipeline approach is prone to accumulate errors from pipeline components, but on the other hand, an end-to-end supervised approach would need an annotated data set to learn from, the creation of which might be costly and time-intensive. The usage of zero-shot cross-lingual transfer or the usage of the output of rule-based systems as training data for supervised systems might be ways to address these problems. We will discuss this in detail in section 9.2.

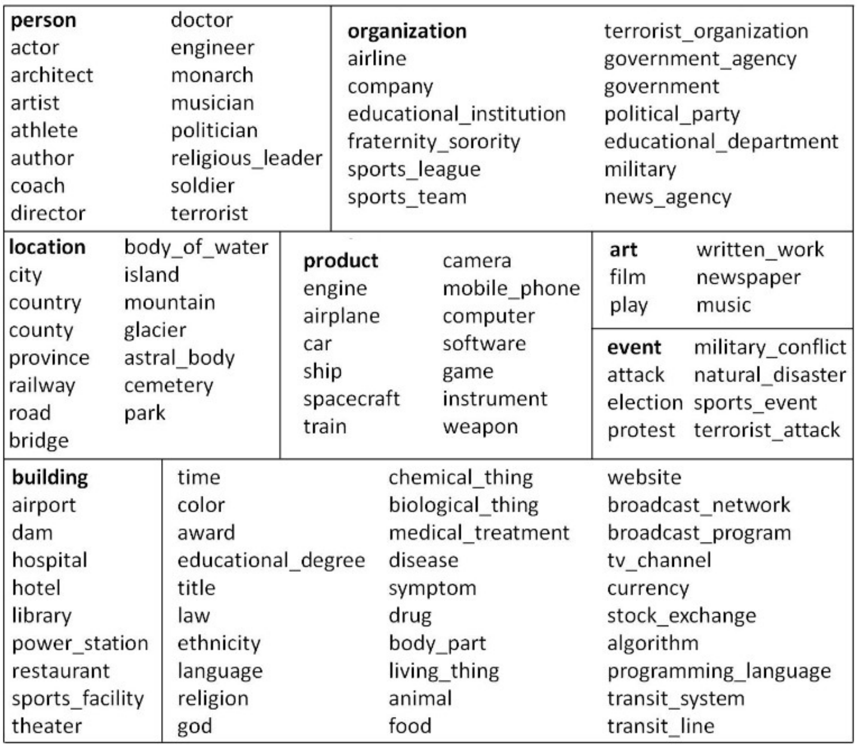
Chapter 4

Fine-Grained Entity Typing

4.1 Introduction

As discussed in chapter 2, the typing of nouns plays an important role in the construction of entailment graphs. The meaning of a predicate is highly context dependent, and therefore different entailments apply if the same predicate is seen in different contexts. Take for example the verbs “eradicate” and “cure”. In the case of a medicine eradicating a disease, this entails the medicine curing the disease. But if a cat eradicates all the mice in the house its the opposite of the cat curing the mice. This leaves us with the task of defining what *same context* and *different context* mean for our use case. If we constrain entailment relations only to apply between predicates that occur with exactly the same subjects and objects, we will have too few examples to construct meaningful feature vectors representing the predicates. On the other hand, if we don’t constrain what we consider as same context, we will predict entailments between predicates where the context is different, like in earlier stated example.

To navigate this, we choose to assign a type to the nouns that appear as subject and object to the predicate. The type of a noun is a more general way to describe the noun, e.g. “Berlin” could be assigned the types “location” and “city”. The simplest type system is the one used by named entity detection systems. The NLP software Stanza (Qi et al., 2020) for example assigns 4 different types to words that are recognized as named entities: ‘person’, ‘location’, ‘organisation’, and ‘misc’. Preliminary experiments have shown, that this granularity is too coarse for our approach at recognising entailments between predicates. To keep our work compatible with the English entailment graphs by Hosseini et al. (2018) we choose the same type system that they use, the FIGER system introduced by Ling and Weld (2012).



person
actor
architect
artist
athlete
author
coach
director
doctor
engineer
monarch
musician
politician
religious_leader
soldier
terrorist

organization
airline
company
educational_institution
fraternity_sorority
sports_league
sports_team
terrorist_organization
government_agency
government
political_party
educational_department
military
news_agency

location
city
country
county
province
railway
road
bridge
body_of_water
island
mountain
glacier
astral_body
cemetery
park

product
engine
airplane
car
ship
spacecraft
train
camera
mobile_phone
computer
software
game
instrument
weapon

art
film
play
written_work
newspaper
music

event
attack
election
protest
military_conflict
natural_disaster
sports_event
terrorist_attack

building
airport
dam
hospital
hotel
library
power_station
restaurant
sports_facility
theater

time
color
award
educational_degree
title
law
ethnicity
language
religion
god

chemical_thing
biological_thing
medical_treatment
disease
symptom
drug
body_part
living_thing
animal
food

website
broadcast_network
broadcast_program
tv_channel
currency
stock_exchange
algorithm
programming_language
transit_system
transit_line

Figure 4.1: All FIGER types, taken from Ling and Weld (2012). These are the types used in English and German entailment graph construction.

4.2 Typing via Linking to a Data Base

Typing is part of the relation extraction pipeline, that extracts subject-predicate-object triples (also called relation triples) from a large corpus of text, with the goal of using the subject and object of a predicate to construct a feature vector that represents the predicate. In their English relation extraction pipeline Hosseini et al. (2018) type entities via linking them to an data base: AIDA-light (Nguyen et al., 2014) links a named entity to its Wikipedia entry. Then, the URL of the Wikipedia entry is mapped to its freebase entity (Bollacker et al., 2008). The type of the freebase entity is then mapped to FIGER types (Ling and Weld, 2012). Only the first, coarser grained, level of the FIGER type hierarchy is used, which is made up of 37 types (all types can be seen in 4.1). Types are, for example, “product” and “religion”. If an entity is not found in Freebase, the type “thing” is assigned.

When approaching the task of typing for the German relation extraction pipeline, one possible approach is to copy Hosseini et al. (2018)’s method for the typing of German entities. But the assumption that German named entity linking is of similar quality than English named entity linking is problematic. It has proven very difficult to run a German named entity linker with the amount of data necessary for large entailment graphs. The quality of the named entity linking is low. Additionally, the German relation extraction pipeline needs to include general entities to extract a sufficiently large amount of relation triples. But so far there is no German entity linker that is able to handle both named entities (e.g. “Barack Obama”) and general entities (e.g. “ex-president”).

Preliminary experiments with a coarser grained type set (person, location, organization, event, misc) showed that these types are too coarse grained to produce useful entailment graphs. This lead us to the the decision to create a German fine-grained entity typing system that can handle both named and general entities. The following sub-chapters describe our work conducted on the topics of fine-grained named and general entity typing, which was published at the Workshop on Insights from Negative Results 2021 (Weber and Steedman, 2021b) and at the TextGraphs Workshop 2021 respectively (Weber and Steedman, 2021a).

4.3 Named Entity Typing, Annotation Projection vs. Zero-Shot

The task of fine-grained entity typing (FET) is to assign a semantic label to a span in a text. The task is distinct from coarse-grained entity typing as done by named entity recognition systems because these systems are restricted to a small set of labels like ‘person’, ‘organization’ and ‘location’ which are not helpful for tasks that require more precise information about the entities. For example, FET assigns the label ‘/location/city’ to the named entity ‘*Berlin*’ in the sentence *‘From 1997 to 2000, it had a permanent exhibition in Berlin.’*

Fine-grained entity typing uses a high number of types in a multilevel hierarchy, which can be seen in the level 2 label ‘/location/city’ (see Figure 4.2). In this work we use the FIGER type hierarchy which consists of two levels with 112 types in total (37 level 1, 75 level 2). FIGER types are derived from the knowledge graph Freebase (Bollacker et al., 2008). They are both interpretable by humans and useful in NLP applications such as relation extraction (Kuang et al., 2020).

There are systems for named entity recognition and coarse-grained entity typing in languages other than English (e.g. Stanza (Qi et al., 2020)), but systems for FET with FIGER types are only available in English, due to the lack of FIGER annotated data in other languages. Because manual annotation is time consuming and expensive,

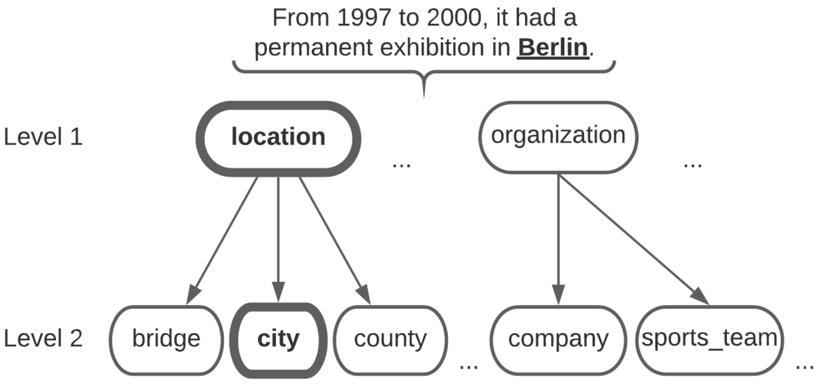


Figure 4.2: An example of fine-grained entity typing with the FIGER ontology. Correct types are highlighted.

various methods have been proposed to expand NLP models to other languages without additional manual annotation. The method of **annotation projection** (Yarowsky and Ngai, 2001) uses parallel text to automatically create annotated corpora. Annotations from the resource-rich language are transferred to the resource-poor language using word alignment between translated sentences.

Annotation projection has been used successfully for the task of coarse-grained named entity typing in conjunction with named entity recognition (Agerri et al., 2018; Li et al., 2021; Ni et al., 2017). We follow these examples by using a parallel English-German corpus, automatic named entity recognition and a state of the art English FET model (Chen et al., 2020a) to assign FIGER type labels on the English side for transfer. We then project the labels onto the German half of the corpus. The output of this process is a German corpus annotated with FIGER types, which we use to train a German FET model.

Another approach to the same problem is **zero-shot cross-lingual transfer**, in which a model built on multilingual word embeddings and trained on high-resource language data is applied to test data in a different language. Because the English FET model used in this work (Chen et al., 2020a) relies on contextualised multilingual word embeddings (XLM-RoBERTa Conneau et al. (2020)) it is possible to train it on English data and to test it on German data.

We compare the two approaches and show that the annotation projection approach amplifies the model's tendency to under-predict level 2 types, which lowers model performance. We also introduce three new test sets for German FET on which zero-shot cross-lingual transfer performs better than models trained with German or a mix of

German and English data.

4.3.1 Related Work

To the best of our knowledge there is no work that compares annotation projection directly against zero-shot cross-lingual transfer. While annotation projection has been used in a variety of tasks, there has not been a study of a case where this approach fails. Authors admit that the quality of the annotating system plays role (e.g. Ehrmann et al. (2011); Ni et al. (2017)), but they don't specify model properties that are necessary for the approach to work, instead focusing on ways to mitigate noise. Agerri et al. (2018) focus the alignment of annotations and Ni et al. (2017) train a system to assess the combined quality of the source annotation system and the alignments. Our work is different because we show how a certain model bias (in our case under-prediction of level 2 labels) is amplified by the approach. Additionally, neither of them compares directly against zero-shot cross-lingual transfer.

Pires et al. (2019); Hsu et al. (2019) and Artetxe and Schwenk (2019) show the strengths of zero-shot cross-lingual transfer on a variety of different NLP tasks, but they do not address fine-grained entity typing. Zhao et al. (2021) conclude that zero-shot performance can be improved by choosing a small amount of high quality training data from the target language. We test their approach for the FET scenario, but arrive at unclear results.

Beyond annotation projection and zero-shot cross-lingual transfer there have been other approaches to bridging the gap between FET in English and other languages. Notably, Heinzerling (2019) use the inter-language links of Wikipedia to obtain training data for their typing system. In contrast to our approach, they do not collect sentences in which the entities occur which makes their training data unsuitable for training the FET system of Chen et al. (2020a). Therefor a comparison to their work is out of the scope of this chapter.

4.3.2 Method

In this work we use the **hierarchical typing model** of Chen et al. (2020a) trained on English gold data for the zero-shot approach and also to annotate the English side of the parallel text for annotation projection. We train the model with English silver data to show the amount of noise added by automatic annotation and finally we train it with German data which was produced by annotation projection.

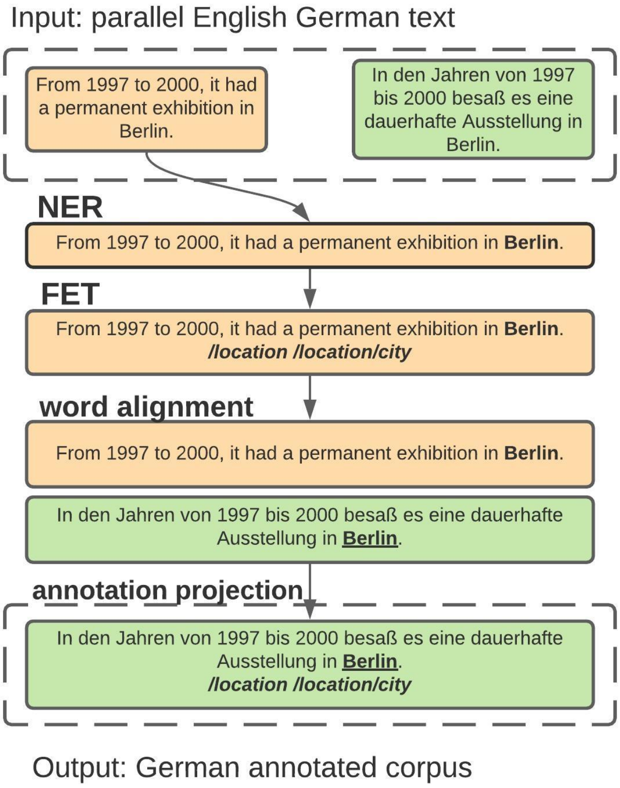

Input: parallel English German text
From 1997 to 2000, it had a permanent exhibition in Berlin.
In den Jahren von 1997 bis 2000 besaß es eine dauerhafte Ausstellung in Berlin.
NER
From 1997 to 2000, it had a permanent exhibition in Berlin.
FET
From 1997 to 2000, it had a permanent exhibition in Berlin.
/location /location/city
word alignment
From 1997 to 2000, it had a permanent exhibition in Berlin.
In den Jahren von 1997 bis 2000 besaß es eine dauerhafte Ausstellung in Berlin.
annotation projection
In den Jahren von 1997 bis 2000 besaß es eine dauerhafte Ausstellung in Berlin.
/location /location/city
Output: German annotated corpus

Figure 4.3: Our annotation projection setup uses parallel text and an automatic named entity recognition component to generate an annotated corpus in German.

In the hierarchical typing model the entity and its context are encoded using multilingual XLM-RoBERTa (Conneau et al., 2020). For each type in the FIGER ontology the model learns a type embedding. It passes a concatenated entity and context vector through a 2-layer feed-forward network that maps into the same space as the type embedding. The score is the inner product between the transformed entity and context vector and the type embedding. For further model details refer to Chen et al. (2020a).

We use the method of **annotation projection** to generate German training data. A diagram of our pipeline can be seen in figure 4.3. To annotate the English halves of our parallel corpora with FIGER types preprocessing is necessary. Due to its automatic creation the WikiMatrix corpus contains a small amount of German sentences in its English half and English sentences in its German half, the translations of which are assigned very high confidence. We remove these by discarding the 5000 highest-confidence sentences.

To enable annotation by the English FET system, we run a named entity recognition system over the English input sentences (see the second box of Figure 4.3). We used the

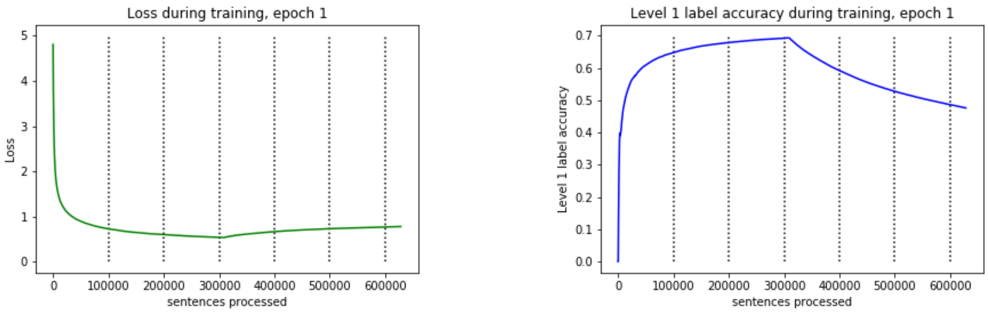


Figure 4.4: Loss rises and level 1 label accuracy deteriorates as the quality of samples gets worse towards the end of the automatically aligned and sorted corpus. The graphs show a cut-off point at approximately 300 thousand sentences. We use this information to select a high-quality slice of the corpus to train our system.

NER component of Stanza (Qi et al., 2020) for this task. We then use the English FET model to assign FIGER types to the named entities (see the third box of Figure 4.3). The FET model only annotates one entity per sentence. Sentences that contain more than one named entity occur multiple times in the English input, so that each entity receives an annotation.

We use ZAP (Zalando, 2020) to obtain a word alignment between the English and German halves of our parallel corpora (see the fourth box of Figure 4.3). While the similar tools fast_align (Dyer et al., 2013) and Giza++ (Och and Ney, 2003) are language agnostic, ZAP's model for English-German word alignment uses probabilities computed from large parallel corpora. We then use our own code to project the fine grained entity type labels from the annotated English text to its German translation. We use static rules to filter out misalignments, e.g. discarding all cases where not all words of an entity were aligned.

We then use the resulting German FET annotated corpus to train our FET model. Because of the ordering by alignment quality in the machine-aligned WikiMatrix corpus, we introduce a preprocessing epoch to the training to mitigate noisy input. During training the model receives the sentences in exactly the order that they occur in the corpus. In the WikiMatrix corpus the sentences are sorted by the confidence of the alignment algorithm. This means that the sentences towards the bottom of the corpus are more likely to be incorrectly aligned. Incorrectly aligned sentences are more likely to have incorrectly projected labels. Therefor the quality of FIGER type annotations in the resulting German data is higher towards the beginning of the corpus and lower towards its end.

During the first epoch of training this drop in quality can be observed in the change of learning rate and the accuracy of predictions after approximately 300,000 sentences (see Figure 4.4). These curves give us important information about what portion of the data is clean enough to be used in the following epochs of training. It gives us a possible cut off point for our data set at 300,000 sentences, so that in the next epochs we only train on a slice of the corpus before this point.

To show the effect of increasing training data size we select for our experiments 3 slices of data that were processed before the cut off point: the first 100K, 200K and 300K sentences of the corpus.

4.3.3 Experimental Setup

Training Data To contrast the zero-shot cross-lingual transfer approach with models trained on automatically annotated and projected data we use three sources of training data. We use the 2M sentences English FIGER corpus as described by Ling and Weld (2012) as a source of **English human annotated data**, which we will refer to as *EN gold*. The data set consists of English Wikipedia articles and we use it to train the *zero-shot gold* model.

Second, we use **English machine annotated data** (*EN automatic*). Annotating English data using a model is the first step of the annotation projection. We use this data to train the *zero-shot automatic* model to examine the amount of noise added by automatic annotation. We generate *EN automatic* from English sentences from the WikiMatrix corpus Schwenk et al. (2021), using the hierarchical typing model trained on 2 M sentences *EN gold*. Lastly, we use **German annotation projected data** (*DE projected*) that was generated by projecting the labels from the *EN automatic* onto German. We use this data to train the *annotation projected* model.

We portion each training corpus into slices of 100, 200, 300 and 400 K sentences to compare the influence of data size. For *DE projected* only 300 K sentences are available, because only part of the parallel sentences in the WikiMatrix corpus are of high enough quality for annotation projection.

An important point for our experiments is the **label distribution** in the training corpora (see table 4.1). The hierarchical typing model has the tendency to underpredict the finer-grained level 2 labels (e.g. /person/actor, as opposed to level 1 label /person), which leads to a different distribution of labels in *EN gold* and the other corpora. Compared to approximately 100 K level 2 labels per 100 K sentences in the gold data,

	EN gold	EN automatic	DE projected
Level 1 labels	60 %	78 %	77 %
Levvl 2 labels	40 %	22 %	23 %
Level 1 labels	155679	148571	150166
Level 2 labels	104807	42008	43604

Table 4.1: Percentage and total numbers of level 1 and level 2 labels in 100 K sentences of the training corpora. Data created by annotation with the hierarchical-typing model contains fewer level 2 labels than human annotated gold data.

we only see about 50 K level 2 labels in the silver data. This tendency does not depend on the different input data: If we use a model trained on 100 K *EN gold* to predict labels on an unseen portion of *EN gold*, only 25 % of the resulting annotations are level 2 labels.

Metrics Following previous FET literature we evaluate the results of our model using strict accuracy (Acc). The strict accuracy is the ratio of instances where the predicted type set is exactly the same as the gold type set. We also evaluate per hierarchy level.

Test sets We compare performance using the following test corpora: **1)** a German machine translation of the test split of the English FIGER corpus Ling and Weld (2012), which was manually corrected to eliminate translation and labelling errors (*DE-FIGER*); **2)** 500 manually annotated German sentences from the WikiMatrix corpus (*DE-Wiki*), which we consider to be more challenging than DE-FIGER, because it contains a wider range of type labels; **3)** a small challenge set of 135 sentences taken from DE-Wiki, in which we replaced entities with close string matches to English (e.g. 'Präsident Nixon') with specifically German entities of the same type (e.g. 'Bundeskanzler Kohl'), which we call *DE-GermEnt*; and **4)** for experiments where we mix German and English data, we also compare against test split of the English FIGER corpus (Ling and Weld, 2012) (*EN-FIGER*). Data set statistics can be seen in table 4.2.

4.3.4 Results

Monolingual training Figure 4.6 compares the performance of the models *zero-shot gold*, *zero-shot automatic* and *annotation projected* at different training data sizes on the *DE-FIGER* and the *DE-Wiki* test sets. *Zero-shot gold* outperforms *zero-shot automatic* and *annotation projected* on both test sets and in all training data sizes. *Zero-shot gold* trained on the full *EN gold* data set of 2 M sentences performs only 1 percent point

	size	unique lables	total lables
EN-FIGER	563 sentences	42	624
DE-FIGER	563 sentences	42	624
DE-Wiki	500 sentences	57	771
DE-GermEnt	135 sentences	34	213

Table 4.2: Statistics of the different test sets listing size, number of unique labels and total number of labels. While DE-FIGER is parallel to the commonly used English FIGER test set, DE-Wiki contains more unique labels and more labels in total.

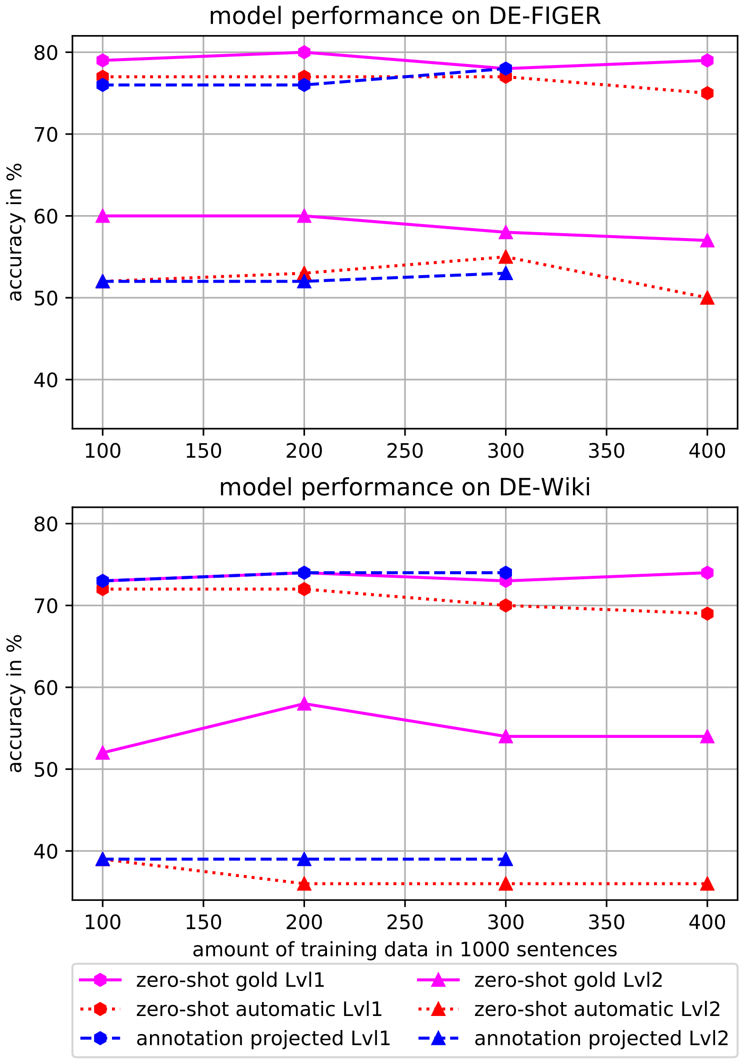


Figure 4.5: Zero-shot cross-lingual transfer performs best on both German data sets. *EN automatic* and *DE projected* perform similar on both data sets, with a wider gap in level 2 performance on *DE-Wiki*

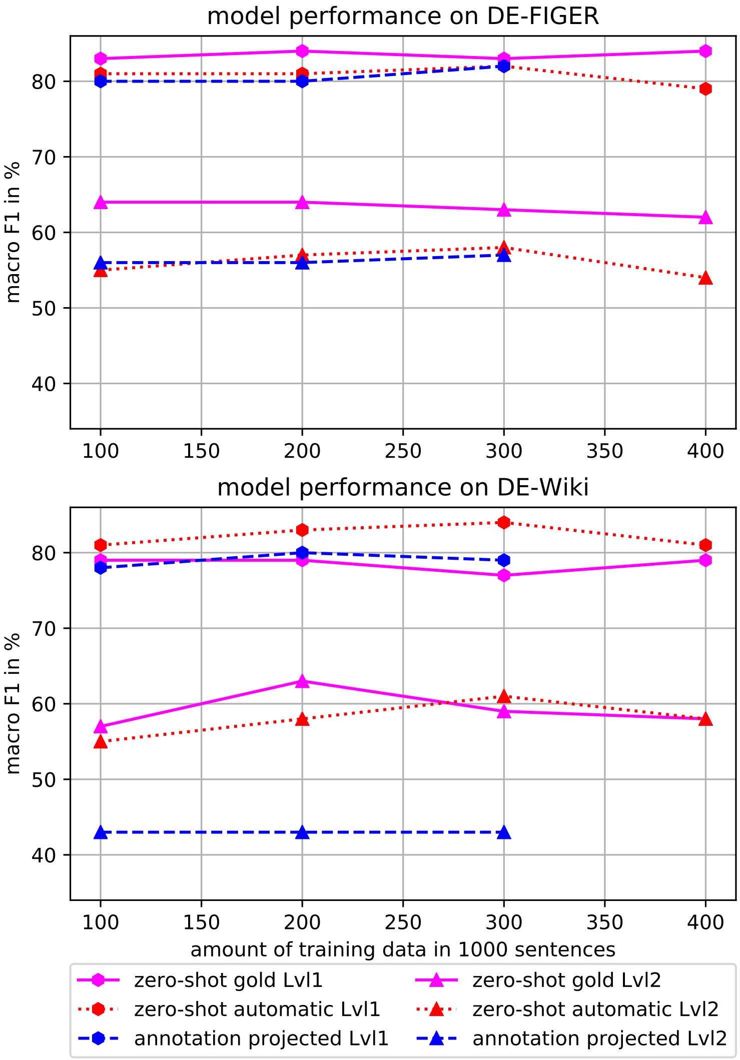


Figure 4.6: In terms of macro F1 score the zero-shot approach outperforms all other approaches on DE-FIGER, being slightly outperformed on L1 label types on the DE-Wiki test set.

better on level 1 labels and 3 percent point better on level 2 labels than a model trained with 400 K sentences, which shows that smaller data slices are sufficient to reach most of the possible performance with this data set.

While for level 1 type labels *annotation projected* gets close to the performance of *zero-shot gold* on both test sets, on level 2 type labels the system falls behind *zero-shot gold*, with a wider gap on *DE-Wiki*. The comparison between *zero-shot automatic* and *annotation projected* is less clear. On the *DE-Wiki* test set *annotation projected* consistently outperforms *zero-shot automatic*, while on *DE-FIGER* both systems perform very similarly.

The high performance of *zero-shot gold* and the noisier *zero-shot automatic* might be due to the quality of English and German embeddings in XLM-RoBERTa, as both are high resource languages from the same language family. This confirms Lauscher et al. (2020) who show that this method works especially well for close, high resource language pairs and low level semantic tasks. The noise introduced by the annotation projection approach affects level 2 label performance the most (see figure 4.7 and table 4.1). But the amount of level 2 labels in the training data can not be the only reason for this. The total number of labels in the silver corpora (see table 4.1) shows that 200 K of silver training data contain approximately the same amount of level 2 labels as 100 K of gold data. Nevertheless, the level 2 performance of systems trained on 200 K of silver data lies behind the model trained on 100 K of *EN gold*. This points towards the possibility, that not only the amount of level 2 labels in the training data, but also their quality and their proportion with level 1 labels plays a role for the hierarchical-typing model.

		DE-FIGER		EN-FIGER	
name, size	metric	Level 1	Level 2	Level 1	Level 2
DE projected, 200K	Acc	75 %	52 %	79 %	54 %
DE projected, 200K	maF1	80 %	56 %	82 %	57 %
DE proj.+EN aut., 200K	Acc	**77 %**	**54 %**	78 %	54 %
DE proj.+EN aut., 200K	maF1	**81 %**	**58 %**	82%	57 %
EN automatic, 200K	Acc	76 %	53 %	**79 %**	**55 %**
EN automatic, 200K	maF1	81 %	57 %	**83 %**	**58 %**

Table 4.3: **Accuracy and macro F1 score in %** of a model trained with both English and German data, in comparison with monolingual data tested on parallel test sets. While performance in German is best with mixed data, performance in English is best with only English training data.

Multilingual training The underlying XLM-RoBERTa embeddings allow to train a model with both German and English data. For this we combine slices from *DE projected* with *EN automatic*, because these data sets have the same distribution of labels. Table 4.4 shows the performance of a model trained with evenly mixed data (*EN+DE*) in comparison with monolingually trained models of the same size tested on *DE-FIGER* and *EN-FIGER*. German performance benefits from using both German and English training data, while performance in English is best with only English data. The mixed model does not outperform *zero-shot gold* on these test sets.

The low performance in the data mixing scenario compared to *zero-shot gold* can

		DE-FIGER		EN-FIGER	
name, size	metric	Level 1	Level 2	Level 1	Level 2
DE projected, 200K	maP	82 %	57 %	85 %	59 %
DE projected, 200K	maR	79 %	55 %	85 %	56 %
DE proj.+EN aut., 200K	maP	**83 %**	**59 %**	84 %	58 %
DE proj.+EN aut., 200K	maR	**81 %**	**57 %**	81%	56 %
EN automatic, 200K	maP	83 %	59 %	**85 %**	**60 %**
EN automatic, 200K	maR	80 %	56 %	**82 %**	**57 %**

Table 4.4: **Macro Pracision and Macro Recall %** of a model trained with both English and German data, in comparison with monolingual data tested on parallel test sets.

be explained with the distribution of labels in the silver corpora. Due to the noise added when labels are projected from English to German, the mixed model tested in German profits from the addition of higher quality English data, but not vice versa.

Few-shot training Zhao et al. (2021) suggest that few-shot learning improves zero-shot performance. To test this we take a model trained on 100 K sentences *EN gold* and fine-tune it by training on the 135 sentence manually annotated *DE-GermEnt* data set. We evaluate the resulting model's performance on *DE-FIGER*. In comparison with the model trained on 100 K *EN gold* only, the performance of the resulting model is 10 percent point lower in accuracy of level 1 labels and 12 percent point lower on level 2 labels. We did not specifically select which sentences to use like Zhao et al. (2021), which is an avenue for future work. The low performance of the few-shot model could be due to the high number of different labels, only a few of which can be observed during few-shot training, but further work is needed to confirm this.

German entities To challenge *zero-shot gold*, we test a model trained 2 M sentences *EN gold* on the test set *DE-GermEnt*. Surprisingly, we find that the model performs better on *DE-GermEnt* than on its English entity counter part, with 1 percent point higher performance on level 1 labels and 3 percent point higher performance on level 2 labels. It is unclear why *zero-shot gold* behaves this way, and examining this with larger challenge data sets it an avenue for future work.

Other sources of noise One other source of noise, in addition to automatic FET annotation and annotation projection is added by our usage of a named entity recognition system as a pre-processing step. A closer examination of the noise that is added by this component is an avenue for future work.

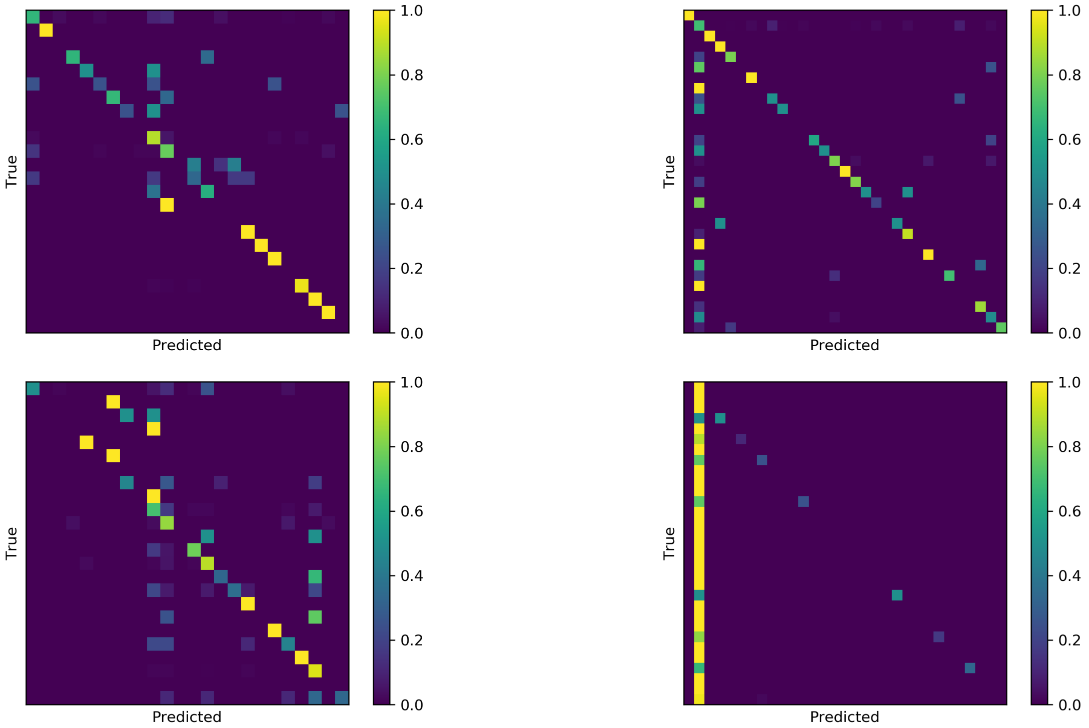


Figure 4.7: The upper two matrices show level 1 and level 2 performance of EN gold 2 M, the lower two matrices show the same for DE silver 200K. While the English model has a slight tendency to predict no label for level 2 label, this tendency is stronger in the German model. The yellow vertical line shows this effect.

4.3.5 Discussion

Our results show that zero-shot cross-lingual transfer building upon XLM-RoBERTa is a strong baseline for the task of FET and the language pair of English and German. It outperforms annotation projection on three new test sets. We also show that in our specific scenario annotation projection using the hierarchical typing model amplifies the models tendency to underpredict level 2 types. This happens, because we use data generated by this model in English to train the same model in German. Figure 4.7 shows the confusion matrices for level 1 and level 2 labels for EN gold 2 Mil and DE silver 200K. While for level 1 labels in EN gold there is no dominant class that labels are misclassified to, the most common misprediction for level 2 labels is to assign no label at all, which can be seen as the dotted vertical line in the second upper confusion matrix. When comparing the upper confusion matrices to the lower ones, it becomes clear that this trend to under-predict level 2 labels is even stronger in the German model. The German model sees less level 2 labels in its training data and therefore doesn't learn to predict them.

One way to mitigate these shortcomings would be to sample level 1 and level 2 labels in a training corpus so that they have the same distribution as in the gold data, although this would not control for data quality. Another way could be to machine translate the manually annotated English corpus into German and then use annotation projection, as suggested by Ehrmann et al. (2011). This way the label distribution of the human annotated data could be preserved as well. Improving the few-shot approach and designing more challenging test sets are other avenues to explore. Lastly, this approach only concentrates on names entities. These make up only a part of all the entities in text, which is why in the following sub-chapter we examine the task of typing general entities.

4.4 General Entity Typing, Using GermaNet for Training Data Generation

Entities can appear in text in many forms. In the sentences 'Barack Obama visited Hawaii. The ex-president enjoyed the fine weather.' both 'Barack Obama' and 'ex-president' should be assigned the type '/person/politician' by a fine-grained entity typing system. While the typing of the **named entity** (NE) 'Barack Obama' can be performed by state of the art entity typing systems, it is unclear how well these systems perform on **general entities** (GEs) like 'ex-president'. We find that accuracy and F1 score of a state-of-the-art German fine-grained entity typing system (the system described earlier in this chapter, Weber and Steedman (2021b)) are 17 percent point lower on general entities than on named entities. This is because the training data for these systems contains only named entities, but not general entities (e.g. Weber and Steedman (2021b); Ling and Weld (2012)). This is the problem we address with our approach.

Because manual annotation of training data is costly and time intensive we propose an approach that uses existing resources to create silver annotated GE typing data. For this we use German text taken from Wikipedia, GermaNet (a German WordNet equivalent, Hamp and Feldweg (1997)) and the FIGER type ontology (Ling and Weld, 2012). We use GermaNet version 8.0, which contains 111361 lemmata. The resulting data can be added to existing NE typing data for the training of a neural entity typing system. In our approach we use the hierarchical typing model of Chen et al. (2020a), which builds upon contextualized word embeddings. It has shown good performance on public benchmarks and is freely available.

We compare our approach against using only NE data for training and a rule-based approach and achieve 10 percent point improvement in accuracy of the prediction of level 1 FIGER types for German general entities, while decreasing named entity prediction accuracy by only 1 percent point. Our approach can be seen as a proof of concept and a blueprint for the use of existing WordNet resources to improve entity typing quality in other languages and domains.

4.4.1 Related work

The problem of GE typing performance has not been examined specifically before, nor has it been addressed for the case of German. Choi et al. (2018) create a fine-grained entity typing system that is capable of typing both GE and NE in English by integrating GEs into their training data. Their approach relies on large amounts of manually annotated data, and is therefore not feasible for our case. Moreover they propose a new type hierarchy, while we stick to the widely used FIGER type hierarchy, to make the output of our system consistent with that of other systems for tasks like multilingual knowledge graph construction.

Recent advances in typing NE in English have harnessed the power of contextualized word embeddings (Peters et al., 2018; Conneau et al., 2020) to encode entities and their context. These approaches use the AIDA, BNN, OntoNotes and FIGER ontologies, which come with their own human annotated data sets (Chen et al., 2020a; Dai et al., 2019; López et al., 2019). By choosing to use the model of (Chen et al., 2020a), we build upon their strengths to enable GE typing in German.

German NE typing suffers from a lack of manually annotated resources. Two recent approaches by by Ruppenhofer et al. (2020) and Leitner et al. (2020) use manually annotated data from biographic interviews and court proceedings. Owing to the specific domains, the authors modify existing type onthologies (OntoNotes in the case of biographic interviews) or come up with their own type ontology (in the case of court proceedings). This limits the way their models can be applied to other domains or used for multilingual tasks. Weber and Steedman (2021b) use annotation projection to create a training data set of Wikipedia text annotated with FIGER types. We build upon their data set to create a German model that types both NEs and GEs.

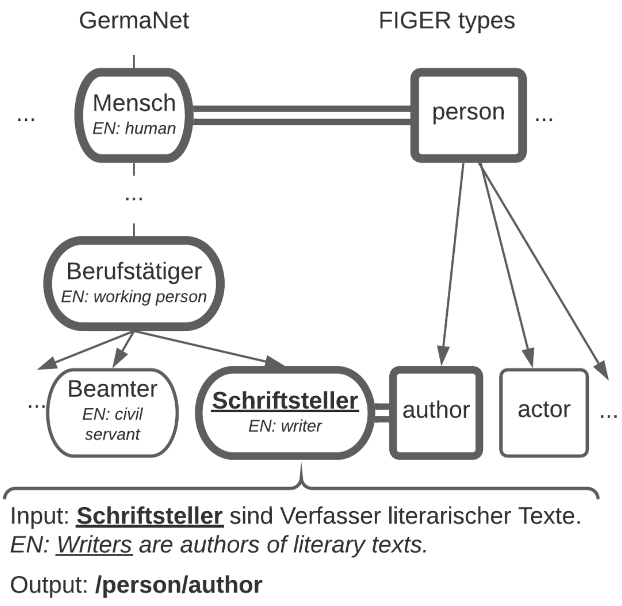


Figure 4.8: An example of FIGER type assignment using GermaNet. The manual mapping between GermaNet and FIGER is indicated by double lines. Whenever a word in the hypernym path of the input word is mapped to a FIGER type, the respective type gets assigned.

4.4.2 Method

GermaNet (Hamp and Feldweg, 1997) is a broad-coverage lexical-semantic net for German which contains 16.000 words and is modelled after the English WordNet (Fellbaum, 2010). The net contains links that connect nouns to their hyponyms and hypernyms. This way GermaNet implicitly contains a fine-grained ontology of nouns. Although some NE are contained in GermaNet, the vast majority of nouns are GEs.

We manually map the 112 FIGER types to nouns in GermaNet. Starting from a German translation of the type name (e.g. the type ‘person’ translates to ‘Mensch’) we add terms that best describe the FIGER type. This mapping enables us to look up a word in GermaNet and check if any of its hypernyms are mapped to a FIGER type. If this is the case, we can assign the corresponding FIGER type to the word in question. Figure 4.8 illustrates this method. We use this method to generate German GE training data and as our rule-based baseline.

We use this GE training data in addition to German NE typing data to train the **hierarchical typing model** of Chen et al. (2020a). In this model the entity and its context are encoded using XLM-RoBERTa (Conneau et al., 2020). For each type in the

FIGER ontology the model learns a type embedding. We pass the concatenated entity and context vector trough a 2-layer feed-forward network that maps into the same space as the type embedding. The score is an inner product between the transformed entity and context vector and the type embedding. For further model details refer to Chen et al. (2020a).

4.4.3 Experimental setup

Data sets As a NE training set we use the German fine-grained entity typing corpus of Weber and Steedman (2021b). This data set was generated from the WikiMatrix corpus by Schwenk et al. (2021) using annotation projection.

To create the GE training data, we use the German portion of the WikiMatrix corpus. By using the same genre we make sure that no additional noise is added by domain differences. Moreover, the original English FIGER data set was created from Wikipedia text, so we can assume that all FIGER types are well represented in the WikiMatrix data.

GE training data generation To generate GE training data we take the following steps: First, we split off 100 K sentences from the top of the German part of the WikiMatrix corpus. We use spaCy (Honnibal et al., 2020) for part of speech tagging. Every word tagged as a noun is looked up in GermaNet. We use the method described earlier to assign FIGER types to the noun.

This lookup in GermaNet is not context-aware, so polysemous words are assigned multiple contradicting types. We only include words in our training data that have less than two level 1 types and not more than one level 2 type. This filter discards about 41 % of all input words. We discuss the implications of this filter later in this chapter. The resulting corpus consists of 200K sentences of German FIGER typed GE data [1].

Training set up In our experiments we compare six different training setups against a rule-based baseline using only GermaNet.

Only NE data: In this setup we train the hierarchical typing model on 200K sentences taken from the German fine-grained NE typing corpus by Weber and Steedman (2021b).

Mixing NE and GE data: In this setup we add either 20K, 40K, 60K, 80K or 100K sentences of automatically generated GE training data to 200K sentences taken from the corpus of Weber and Steedman (2021b) and train the hierarchical typing model on it. We shuffle the sentence order before training.

[1] The generation code and generated data can be found here: https://github.com/webersab/german_general_entity_typing

	Acc L1		F1 L1		Acc L2		F1 L2	
Model	NE	GE	NE	GE	NE	GE	NE	GE
200K (only NE)	0.74	0.57	0.79	0.62	0.39	0.25	0.44	0.30
220K	0.73	0.66	0.78	0.71	0.37	0.29	0.42	0.34
240K	**0.73**	**0.67**	**0.77**	**0.72**	0.38	0.29	0.43	0.34
260K	0.72	0.66	0.77	0.70	**0.39**	**0.30**	**0.44**	**0.35**
280K	0.72	0.66	0.77	0.71	0.37	0.30	0.42	0.35
300K	0.70	0.64	0.75	0.68	0.37	0.30	0.42	0.34
GermaNet BL	0.10	0.48	0.10	0.48	0.27	0.08	0.27	0.08

Table 4.5: **Test Set Results** Accuracy and micro F1 score based on training input, tested on 500 NE annotated sentences and 300 GE annotated sentences. GE Level 1 accuracy and Level 1 F1 rises by 9 percent point when 20K sentences of GE training data are added, while NE accuracy and F1 declines by only 1 percent point.

Baseline: We compare these two neural approaches against using only GermaNet. In this baseline we use the approach described in the method section and Figure 4.8 to type our test data.

4.4.4 Evaluation

Metrics Following previous fine-grained entity typing literature we evaluate the results of our model using strict accuracy (Acc) and micro F1 score. The strict accuracy is the ratio of instances where the predicted type set is exactly the same as the gold type set. The micro F1 score computes F1 score biased by class frequency. We also evaluate per hierarchy level accuracy (level 1 type labels being more coarse grained and level 2 labels more fine grained).

Test sets We use the German NE typing test set of Weber and Steedman (2021b) for testing the performance of our systems on the task of NE typing. The test set consists of 500 manually annotated sentences. We create our GE typing data sets by taking that same test set and manually replacing the named entities in it with plausible general entities (e.g. swapping ‘Barack Obama’ for ‘ex-president’). Where this was not possible, we chose another noun from the sentence and manually added the correct type. In all other cases we removed the sentence from the data set. The resulting GE data set consists of 400 sentences, which we split into a 100 sentence development set and a 300 sentence test set.

	Acc L1		F1 L1		Acc L2		F1 L2	
	dev	test	dev	test	dev	test	dev	test
200K	0.62	0.57	0.64	0.62	0.32	0.25	0.35	0.30
220K	0.62	0.66	0.73	0.71	0.34	0.29	0.36	0.34
240K	**0.73**	**0.67**	**0.75**	**0.72**	**0.36**	0.29	**0.38**	0.34
260K	0.71	0.66	0.73	0.70	0.35	**0.30**	0.37	**0.35**
280K	0.73	0.66	0.73	0.71	0.36	0.30	0.38	0.35
300K	0.69	0.64	0.71	0.68	0.35	0.30	0.37	0.34

Table 4.6: **Development Set Results** We report development set and test set performance of the fine-grained entity typing model trained with different amounts of general entity training data. Best development set performance aligns with best test set performance on Level 1 metrics, and is only off by 1 percent point for Level 2 metrics.

4.4.5 Results

Table 4.5 shows the accuracy and F1 scores on the gold German test set. Development set results can be seen in Table 4.6. We use the development set to determine which amount of of added GEs achieves the best result. Best development set performance aligns with best test set performance on Level 1 metrics, and is only off by 1 percent point for Level 2 metrics.

We compare the performance of models trained with different amounts of GE data on the GE and NE test sets described earlier. The test set performance on NE is best when no GE data is added, but GE performance is at its lowest. After adding 20K sentences of GE training data the level 1 accuracy and F1 score on the GE test set rises by 9 percent point. Increasing the amount of GE training data to 40K improves the GE test set performance further with best level 1 results at 40K sentences GE data and best level 2 results at 60K sentences GE data. Adding more GE data beyond these points decreases GE performance.

Although NE performance is worsened by adding GE training data, the decrease in level 1 performance in both accuracy and F1 is only 1 percent point for 20K and 40K GE sentences, with a maximum decrease of 3 percent point when 100K GE sentences are added.

Adding GE training data has a smaller effect on level 2 performance than on level 1 performance, with level 2 accuracy and F1 on the GE test set increasing by 5 percent point when 60K sentences of GE data are added. Adding GE training data initially decreases performance on NE level 2 types, but at 60K sentences of GE data is just as good as without them.

Adding more than 60K sentences of GE data does not improve GE test set performance, but decreases both NE and GE test set performance in accuracy and F1 score. We can also see that the GermaNet baseline is outperformed by all systems, although its performance on level 2 GE types is close to our best models. We will discuss possible explanations in the next section.

4.4.6 Discussion

The results show that the models' performance on GE typing can be improved using a simple data augmentation method using WordNet, while only lightly impacting the performance on NE typing.

All neural models outperform the GermaNet baseline. This raises the question why the neural systems were able to perform better than GermaNet on GE, although the training data was generated from GermaNet. We speculate that the hierarchical typing model is very context sensitive because of its usage of XLM-RoBERTa to encode entities and their context during training. Because our GE training data provides it with high confidence non-polysemous examples, the model is able to learn which context goes with which type. At test time this awareness of context enables the neural systems to disambiguate polysemous cases, even though it has not observed these cases at training time. This intuition is supported by our test results: For the best performing model (240K) 40 % of the general entities that occur in our test set are never seen in the training data.

A second reason why the neural models outperform GermaNet is that GermaNet does not represent every German noun. A certain word might not be part of GermaNet and therefor no type can be assigned. This is the case for 23 % of words seen during training data generation. The neural models do not have this problem because our vocabulary is larger than the 16.000 words contained in GermaNet and because the neural models assign type labels to out of vocabulary words on the basis of the language model XML-RoBERTa.

Despite these factors the neural models' performance is closely matched by the GermaNet baseline on level 2 labels. Level 2 types are underrepresented in the data, because their prevalence follows their occurrence in the Wikipedia data. This leads to some low-level types being very rare: a signal that is too weak to be learned sufficiently by a neural model. On the other hand, a lookup of words in a preexisting data base like GermaNet is not affected by this issue. While the neural models offer high recall at low precision, GermaNet has higher precision at low recall.

The results also show that 20K sentences of GE data produce the highest increase of GE performance while impacting NE performance least. Adding GE data beyond 60K sentences does not only worsen NE performance but also GE performance. This is due to noise in the GE training data. A manual error analysis of 100 sentences GE training data sentences shows that 35 % have incorrect type assignments. With more GE training data the model starts to overfit to this noise, which leads to decreasing test set performance, affecting NE performance slightly more than GE performance.

Our approach can be taken as a blueprint for improving fine-grained entity typing performance in other languages and domains, as there are WordNets for over 40 different languages. Moreover, the manual mapping we introduced could be replaced by machine-translating English type labels into the language of the WordNet, which would require less resources for human annotation than a manual mapping.

Avenues for future work could be a combination between high-precision but low recall WordNets and neural models, e.g. through incorporating the models' prediction confidence to make a decision whether a WordNet look-up should be trusted over the models' own prediction.

The problem of general entity typing could also be viewed through the lens of coreference resolution: The type of a general entity could be inferred from a named entity that the general entity refers to. However, there might be cases in which no named entity referent exists, or domains and languages where coreference resolution systems are unavailable. In all of these cases combining our method with existing approaches opens new possibilities.

4.5 Conclusion

In this chapter we discussed different approaches to building a German fine-grained entity typing system. We first compared the approaches of annotation projection and zero-shot cross-lingual transfer and found that surprisingly, a zero-shot approach outperforms annotation projection. This means that a system trained on high-quality English data performs better on a German test set than a system trained on noisy German data. This is due to the high transfer learning ability of the underlying contextualised word embedding model XML-RoBERTa, which performs especially well for closely related high-resource languages and low-level semantic tasks.

We then moved on to the task of typing *general* entities (e.g. “ex-president” as opposed to the named entity “Barack Obama”). We showed that it is possible to improve the general entity typing performance of a German named entity typing system using the German WordNet equivalent GermaNet. To do so we created an automatically annotated general entity typing corpus from GermaNet and trained a fine-grained entity typing model that builds upon contextualised word embeddings on it, achieving state-of-the-art results.

At the end of this work we need to make a choice with system to integrate into the German relation extraction pipeline: The zero-shot named entity typing system, or the general entity typing system trained on German silver data. While the zero-shot system performs better on named entities than than the system trained on German data, it performs much worse on general entities. Considering that the majority of the entities extracted by the German relation extraction pipeline are general entities, we choose to integrate the system that performs best on them, i.e. the system described in the previous sub-chapter. Additionally, we follow Hosseini et al. (2018) in using only the first level types of the FIGER hierarchy, where the accuracy of the German general entity typing model on named entities is only 7 percent point worse than the performance of the zero-shot named entity typing model.

This German entity typing model enables us to complete the German relation extraction pipeline. The extracted relations serve as building material for the next step, the calculation of entailment scores between different predicates and the construction of entailment graph. We cover this in the following chapter.

Chapter 5

Entailment Graph Construction

5.1 Introduction

In the previous chapters we covered relation extraction and typing, which provides us with the materials needed to construct feature vectors that represent predicates. The predicates extracted in the previous step can span a wide range from single word predicates (e.g. “visit”) over modifier constructions (e.g. “fail to visit”) to frozen metaphors (e.g. “hold a speech”), also including negated predicates, if they occur in the input data. We created a German relation extraction pipeline including a German entity typing system to extract German relation triples from a large corpus of news text. We kept the German relation extraction pipeline analogous to Hosseini et al. (2018)’s English pipeline so we can use the language agnostic graph construction part of their system to build German entailment graphs.

Because this part of the system is language agnostic, we don’t modify it for German, but instead use the codebase of Hosseini et al. (2018). After extracting subject-predicate-object triples from text, the next step consists of constructing feature vectors that represent the predicates based on the subjects and objects the predicates occur with. We then measure the overlap between the predicate vectors to compute a directional similarity measure, the so called entailment score (details on the theoretical background of this can be found in chapter 2). The predicates with their respective entailment scores are written to output files. Lastly, we evaluate the quality of the entailment scores by comparing them to human judgements using the Levy-Holt data set (Levy and Dagan, 2016a). An example of the input and output of this step can be seen in Table 5.1

Input	(betreiben.seit.1,betreiben.seit.2)#person#event::Paul_Gauselmann::Jahrzehnten (betreiben.1,betreiben.2)#person#organization::Paul_Gauselmann::Parteispendensystem (untersuchen.1,untersuchen.2)#government#event::CDU::Zahlungen ...
Output	predicate:(müssen.1,müssen.zurückgreifen.auf.2)#person#event (müssen.1,müssen.zurückgreifen.auf.2)#person#event 1.0 (leben.1,leben.von.2)#person#event 0.15598375 (anweisen_sein_auf.1,anweisen_sein_auf.2)#person#event 0.09213747 (entdecken.1,entdecken.2)#person#event 0.076194815 (erhalten.1,erhalten.2)#person#event 0.032182068 ...

Table 5.1: An example from the input and output files of the entailment graph building algorithm. The input consists of subject-predicate-object triples with the respective types of the subject and object. Only predicates that occur with the same types of subject and object can entail each other. The output matches a predicate to a list of predicates with their respective entailment scores. Each predicate entails itself (entailment score 1), the other predicates are presented in descending order of entailment score. A high entailment score means that one predicate entails the other.

5.2 Method

Because we are using Hosseini et al. (2018)'s code base without any modification, this section will cover the method in less detail than the original paper and focus more on the implication that the method has for the specific use case of German entailment. The graph building algorithm constructs the predicate vectors by counting the occurrences of specific entity pairs with a predicate. A toy example for this can be seen in figure 5.2. Let's assume our corpus contains only sentences about people and locations, and the relation extraction pipeline extracted the triples "(visit)::Obama::Hawaii", "(visit)::Bush::Washington" and so on. The columns of the table are the vectors constructed using the extracted triples, where each number stands for how often the predicate and the entity pair occur together. We assume a high entailment score between two predicates when one predicate vector is a subset of the other, like in the case of "visit" and "arrive_at". This assumption is directly building upon the distributional inclusion hypothesis explained in chapter 2. Note that the vector representing "born_in" does not form a subset of any of the other vectors. There would therefor be a low entailment score between "born_in" and any of the other predicates.

	visit	arrive_at	born_in
Obama, Hawaii	5	2	4
Bush, Washington	4	1	0
Kafka, Prague	0	0	3

Table 5.2: A toy example exemplifying how the vectors that represent predicates are constructed. "visit" entails "arrive at", while "born in" has no entailment relation with the other predicates.

The actual construction of the representing vectors differs from this toy example in several points. Rather than using the raw counts as features for the vector, the value of each feature is the point-wise mutual information between the predicate and the feature, to balance out the influence of highly common pairs. The vectors are then used to calculate the Balanced Inclusion score (or BInc) (Szpektor and Dagan, 2008). This score is a combination of the symmetric Lin score and the directional Weeds precision score. According to the authors, BInc recognizes directional similarity using the directional measure but penalises infrequent occurrences using the symmetrical measure:

$$BInc(u,v) = \sqrt{Lin(u,v) * WeedsPrecision(u,v)}$$

$$Lin(u,v) = \frac{\sum_{f \in F_u \cap F_v} [w_u(f) + w_v(f)]}{\sum_{f \in F_u} w_u(f) + \sum_{f \in F_v} w_v(f)}$$

$$WeedsPrecision(u,v) = \frac{\sum_{f \in F_u \cap F_v} w_u(f)}{\sum_{f \in F_u} w_u(f)}$$

Here u and v are two predicates, F_u the set of features comprising the representation of u, and $w_u(f)$ is the feature weight for $f \in F_u$. For a certain BInc score at the threshold value λ entailment graphs are defined as $G_\lambda(t_1,t_2) = (V(t_1,t_2), E_\lambda(t_1,t_2))$, where $V(t_1,t_2)$ are the nodes and $E_\lambda(t_1,t_2)$ are the edges of the graph, with t_1 and t_2 being the respective type pair of the graph.

Hosseini et al. (2018) acknowledge the problem of sparsity in the entailment graphs. This is caused in part by the constraint that entailments can only exists between predicates that occur with the same type pair. For example, the predicate “visit” might only occur in the entailment graph of the type pair #person#location, even though a person might visit another person, but we didn’t observe “visit” with the #person#person type pair in the input corpus. This way, a high entailment score might be calculated for a pair of predicates with one type pair, while the same pair of predicates might have not been observed with other types, even though an entailment between them might exist for other types as well. Hosseini et al. (2018) alleviate this problem by introducing a global learning algorithm, that learns so called global entailment graphs (in contrast to the ’local’ graphs that use only predicates with the same type pairs). The algorithm sets three soft constraints: 1) global scores should not diverge too far from local scores 2) a predicate’s entailments should be similar across type pairs, if the predicate’s neighbours are similar and 3) predicates that are paraphrases should have similar entailments. Hosseini et al. (2018) report that this method outperforms all other metrics they tried.

5.3 Experimental Setup and Evaluation

The goal of our experiment is to determine the quality of the German entailment graph in comparison to the English entailment graph of Hosseini et al. (2018). We will present two different English entailment graphs here, one constructed from the NewsSpike corpus (Zhang and Weld, 2013) and one from the larger NewsCrawl corpus. The comparison between English and German entailment graphs is a difficult one, because the two entailment graphs stand on an unequal footing from the beginning. The corpora that the English and German entailment graphs are constructed from are different in content, with the German corpus containing more local news (for more details on the corpora, see chapter 3.2). This might lead to sparser feature vectors for German, because a predicate might occur with a wider variety of locally specific entities. As described in chapter 3, the German relation extraction pipeline also suffers from shortcomings in terms of parsing, named entity recognition and entity typing. To extract a similar amount of relation triples per document as the English relation extraction pipeline, the German relation extraction pipeline includes general entities, which have different occurrence statistics, and might therefore lead to feature vectors that are principally different from their English counterparts. Another difference between the English and the German entailment graphs is the amount of data that each was constructed from. While the NewsSpike corpus is 20 million sentences large and the English portion of NewsCrawl contains 80 million sentences, the German entailment graph was constructed from only 430 thousand sentences, due to the constraints on time and computing resources available to us.

Test Data Set For evaluation of entailment graphs Hosseini et al. (2018) use the data set proposed by Levy and Dagan (2016a) and modified by Ricketts Holt et al. (2018). Levy and Dagan (2016a) approach collecting human annotations of entailment relations between predicates as a question answering problem. If a sentence is considered as the answer to a question by the annotators, Levy and Dagan (2016a) assume that the relation expressed in the answer entails the relation expressed in the question. This way they collected a data set of 16 K sentences. Ricketts Holt et al. (2018) later refined the data set by re-annotating and removing wrong entailments. The Levy Holt data set consists to 20 percent of sentences that entail each other and 80 percent sentences that don't entail each other (see examples from the original English data set and its German translation in Table 5.1).

If we compare the format of the test set in Table 5.1 with the output format of the graph building algorithm in Table 5.1 one problem becomes obvious: The Levy Holt data set contains a binary judgement of two predicates being entailed or non-entailed, while our output provides us with entailment scores. Our output does not provide us with a threshold value to determine which entailment score is high enough to count two predicates as entailed. This is why we (following Hosseini et al. (2018)) report our results as a precision recall curve. We plot the respective precision and recall of

Premise	Hypothesis	Entailment
Abyssinia, exports, coffee.	Coffee, is a native of, Abyssinia.	n
Nigeria, exports, oil.	Nigeria, is the supplier of, oil.	y
Abessinien, exportiert, Kaffee.	Kaffee, stammt aus, Abessinien.	n
Nigeria, exportiert, Öl.	Nigeria, ist der Lieferant von, Öl.	y

Figure 5.1: Example sentences from the English Levy Holt data set and its German translation. Subject, predicate and object are separated by commas.

the entailment graph, varying the threshold parameter λ. A high λ value means that only predicates with high entailment scores are considered as entailed. A lower λ value relaxes this assumption. Graphs where we discard all entailments below a high λ value tend to be sparse but contain less false entailments and graphs with a low λ value are noisier. We also report the area under the precision recall curve.

Experimental Setup At test time we proceed in the following way: Each sentence contains one relation between two entities. We use the relation extraction pipeline described earlier to recognize and type the entities and extract the relation between them. The fact that the Levy Holt data set separates subject, predicate and object by commas helps us, if the relation extraction pipeline is unable to extract this information (e.g. because of parsing errors). At the end of this step we have two relations and the type pair these relations occur with. We then retrieve the output file of the type pair, in the case of our example (see Table 5.1), the #location#product graph. The output file contains pairs of relations with their respective entailment scores. If the value is higher than the λ threshold value we set, we score them as entailments. If the value is lower than the threshold we consider them as non entailed. If one or both of the relations are not in the graph, we can make no claim about their entailment or non-entailment. Nevertheless, in this case we score the pair as non-entailed.

This testing procedure does not differ for English and German entailment graphs. The difficulty of evaluating German entailment stems from the lack of a human annotated German predicate entailment data set. Existing work has either not yet explored multilinguality (e.g.the Levy Holt data set Levy and Dagan (2016a)) or is not focused on our specific use case of predicate entailment, like the XNLI data set (Conneau et al., 2018). Because of this we created a machine translated version of the Levy data set (Levy and Dagan, 2016a) using the DeepL API. We used Stanza dependency parsing (Qi et al., 2020) to induce the separation into subject, predicate and object seen in the

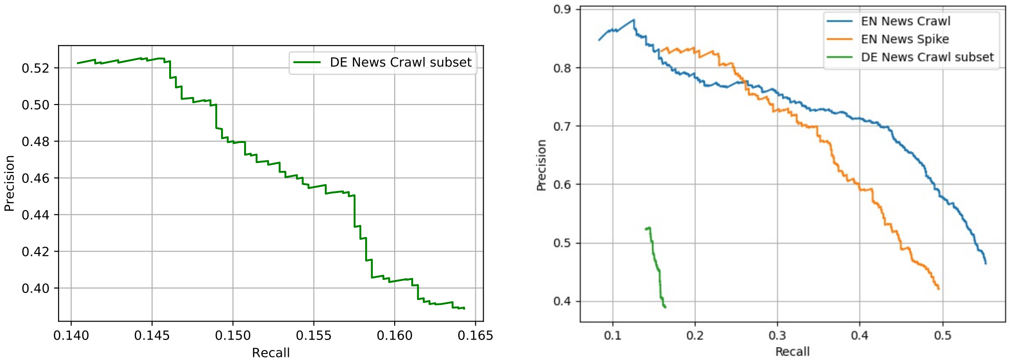


Figure 5.2: Precision Recall curves for the German entailment graph and the German and English entailment graphs in comparison, tested on the English Levy Holt data set and a German translation of that data set. The German entailment graph performs much worse than the English entailment graphs.

original corpus. Both of these steps introduce additional noise, which may influence the evaluation results, e.g. via incorrect translations. Only a manual translation or the manual creation of a German data set can alleviate this problem.

Results The first part of Figure 5.2 shows the precision recall curve for German entailment graph tested on a German translation of the Levy Holt data set. We can see that higher precision values occur with low recall values and vice versa. We can also observe that both in terms of precision and recall, values are quite low. The second part of the figure show the same precision and recall curve alongside the precision recall curves of the two English entailment graphs mentioned earlier, tested on the English Levy Holt test set. It is pretty obvious that the English entailment graphs perform much better than the German one. The area under the curve (AUC) is 0.2070 for English NewsSpike, 0.3346 for English NewsCrawl and 0.0044 for German.

The comparison between the German and English entailment graphs is not necessarily a fair one, because the German entailment graph was created from less data than the English entailment graphs (430 thousand sentences in German as compared to 20 and 80 Million sentences in English). This results in different amounts of created sub-graphs and different amounts of predicates covered. The exact numbers can be found in Table 5.3. The table distinguishes between singleton predicates (e.g. ‘come’), predicate compounds (e.g. ‘come with’ or ‘try to come with’) and noun constructs (e.g. ‘give a speech’). What stands out is that there are much less predicate compounds in German than in English, which is because concepts that are expressed in English by

	DE	EN News Spike	EN News Crawl
sentences	430,000	20,000,000	80,000,000
graphs	708,396	7,428,988	16,269,813
predicates total	145,040	323,958	1,049,078
predicates singletons	120,735	50,348	184,940
predicates compounds	15,654	174,294	496,486
noun constructions	2,286	80,295	237,718

Table 5.3: Statistical properties of the different entailment graphs.

a predicate and a preposition (e.g. ‘come with’) are expressed in German as a single verb (in this case ‘mitkommen’). A closer examination of the differences between the German and English graphs can be found in the following chapter.

Considering the shortcomings discussed earlier, this result is not necessarily surprising. It points to some general problems that working with entailment graphs pose. First, the pipeline and parsing approach to relation extraction involves many independent parts (NER, parsing, FET) and errors are propagated trough the system and accumulate. Second, relations are evaluated by a string match lookup in the graph building output files. This means that if a relation is not seen sufficiently often in the relation extraction step, the relation can not be found at test time. Moreover, similar words or synonyms are not taken into account, because the evaluation only considers exact string matches. This property is limiting for downstream applications of entailment graphs. Additionally, it is unclear whether the German translation Levy Holt data set is a fair test set for the German entailment graph. First, the automatic translation and automatic parsing introduce noise to the data set. Second, while both the Levy Holt data set and the German entailment graph are constructed from news text, the German data set used for constructing the German entailment graph consists of mainly local news and some global news. This creates a domain mismatch with the Levy Holt data set, because the overlap in the covered topics is small.

Therefore we need to consider other ways to evaluate the quality of the German entailment graph. One way is the **manual evaluation** of the relations that are contained in the graph. By manually scoring the relations that exist in the German entailment graph as either true or false we are not hampered by the sparsity of the graph. On the other hand, this approach comes with several difficulties. The output of the entailment graph building system provides us with a list of entailment scores, but with no information about which score is high enough to be considered an entailment. As mentioned earlier, different entailment graph thresholds lead to different graphs, with high thresholds leading to high precision and low recall results, and low thresholds to low precision and high recall. One possibility is to score the entailments in the graph at different thresholds, and creating a precision recall curve similar to the ones shown in Figure 5.2. But doing this exhaustively would take a lot of time. Instead we employ an abbreviated version of this procedure.

For our manual evaluation we choose to consider only entailments with an entailment score higher than 0.7 (for an example of the graph building output, see 5.1). Each type pair (e.g. person and location) has their own entailment graph. We randomly choose 300 examples of true entailments from all of the entailment graphs. We then manually annotate whether an entailment that is judged to be true by the entailment graph is correct. We find that this is the case for 43.6% of the predicate pairs. This is in line with the precision recall curve we report in Figure 5.2. We can not evaluate recall numbers, because we only evaluate entailments that are present in the graph. We admit that is can only offer a snapshot of the German entailment graphs performance.

Another way to address the evaluation of the German entailment graph would be the creation of a new German test data set from the same data that use to build the German entailment graph. In this scenario we take the output of the relation extraction stage (relation triples) and let human annotators annotate entailments between triples. We then compare the annotations of humans to the entailments scores of the graph building algorithm. This approach comes with pros and cons: While we could make sure that there is overlap between entailment graphs and the test set and can calculate both precision and recall numbers, this is time and cost intensive and can only be done for a limited amount of examples. This would also limit our ability to experiment with different input data sets.

An easier way to show the merit of the German entailment graph is through its **usefulness in downstream applications**. We can combine the entailment graph with contextualised word embeddings to return an entailment label for words that don't have an exact string match in the graph. We explore this approach in more detail in chapter 7, where we train a predicate entailment detection model on training data generated from the German entailment graph. We achieve a big improvement with regard to precision and recall in comparison to using only the German entailment graph, which can be seen in figure 5.3.

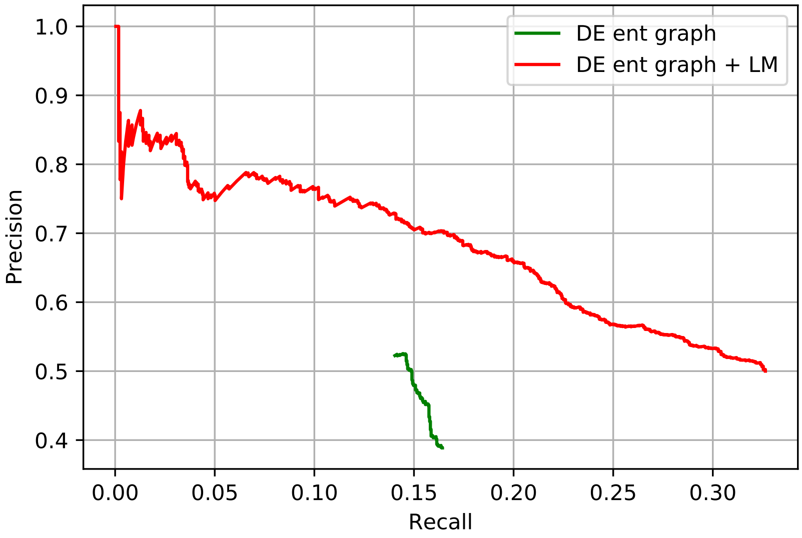


Figure 5.3: Precision and recall curves of the German entailment graph (green) and a predicate entailment detection model trained on German entailment graph data (red), tested on the German translation of the Levy Holt data set. While the German entailment graph by itself performs quite weakly, training a model on data from it leads to a big improvement

5.4 Conclusion

In this chapter we examined how entailment scores between predicates are calculated using the relations extracted by the earlier relation extraction steps. Depending on how we choose the entailment score threshold, the result of this process are more or less densely connected entailment graphs. We can then evaluate the quality of the entailment graphs using the Levy Holt data set. This data set contains sentence pairs and a label that tells whether the two sentences are entailed or not. The sentences only differ in their predicate. This feature makes the test set ideal for our use case, because it allows us to look up the predicates in the entailment graph and check if the entailment graph considers the two predicates entailed or not. We perform this testing procedure for different entailment score threshold values and report our results as precision recall curves.

To perform this test on the German entailment graph we use a German translation of the Levy Holt data set. We find that performance of the German entailment graph is quite low in comparison to the English entailment graph. This could be due to translation noise in the data set. It could also be due to the fact that the German entailment graph is created from less data and the data the the German entailment graph is created from contains more local news, which constitutes a domain mismatch with the Levy Holt data set. We manually score a set of 300 examples and find that 43.6% of them are correct (131 examples in total), which is in line with the precision recall curve we report. Moving forward from this weak result we find that the low performance of the German entailment graph can be improved by combining it with a model that uses contextualised word embeddings, leading to 26 percent points improvement in area under the precision recall curve. We explore this approach in more detail in chapter 7.

Chapter 6

Explicit Alignment of German and English Entailment Graphs

6.1 Introduction

The existence of a German and English entailment graph opens the question whether it is possible to combine these two monolingual resources into one multilingual resource. A multilingual entailment graph, in which English predicates are aligned with their German translations would for example enable us to search for answers to an English question in both English and German documents. We would not only be able to look up the entailed predicates in English, but also paraphrases and entailed predicates in German. A multilingual representation could also introduce new links into the entailment graph. Entailment graphs suffer from sparsity, and the German entailment graph might contain entailments that are missing from the English graph because they have only been seen in the German text.

This chapter is most of all a presentation of negative results. The previous chapters have shown that the German entailment graph is of lower quality than the English one and that the relations contained in the German one are very different from the ones contained in the English graph. This poses difficulties for the alignment of the two graphs. There is an additional shortcoming that stems from the difference between the reading order of this thesis and the timeline of conducting the presented experiments: While we cover German fine-grained entity typing in a previous chapter, part of the work described here was conducted *before* the German fine-grained entity typing system was completed. We therefore separate the chapter into parts: First, we cover our approach at alignment of the German and English entailment graph with mismatched typing systems (part of this has been published in Weber and Steedman (2019)), and we then present an in-depth manual analysis of the possibility of aligning entailment graphs with an improved German relation extraction system that does provide types that match the English ones. We arrive at the conclusion that for the goal of a multilingual entailment graph a different approach is needed.

6.2 Previous Work

Work on entailment graphs has previously only focused on English (Hosseini et al., 2018). The approach most similar to multilingual entailment graphs has been taken by Lewis and Steedman (2013b), who construct a predicate representation in the multilingual domain, working on English and French. Instead of building entailment graphs they cluster predicates monolingually and align the resulting clusters across languages. The authors construct predicate vectors the same way we do, by first extracting subject-predicate-object triples and creating feature vectors representing predicates. They link the named entities occurring with a predicate to a database. Because the named entities in different languages are linked to the same database, the vectors representing English and French predicates are part of the same vector space. Predicates that are paraphrases or translations of each other have similar vectors, because they occur with the same named entities in both languages. The vectors that represent them can be aligned using cosine similarity. Because Lewis and Steedman (2013a) use this bidirectional measure of similarity, both paraphrases and entailments are included in their paraphrase clusters.

In their work Lewis and Steedman (2013a) use Wikipedia articles in English and French describing the same topic. This can be considered as a form of parallel text. Even though there is no alignment between sentences, the fact that the same topics are covered is helpful for the construction of paraphrase clusters. Lewis and Steedman (2013a) use the inter-language links between named entities provided in the Wikipedia articles to align predicates in different languages. They also use the types that were provided with the Wikipedia links. This way they could work with a perfect alignment of named entities to derive the alignment of predicates. Because the linking of German named entities to an external data base like Freebase (Bollacker et al., 2008) or DBPedia (Lehmann et al., 2015) is slow and incomplete, only a partial alignment between named entities in our English and German relation triples can be achieved. Therefore, a different approach for alignment of predicates is needed in our scenario.

Lastly, Lewis and Steedman (2013a) evaluate the quality of their paraphrase clusters on the task of reranking the translation candidates of a statistic machine translation system. The advent of neural machine translation systems has rendered this task obsolete. Following Hosseini et al. (2018)'s work on entailment graphs, we use the Levy Holt data set (Levy and Dagan, 2016a) for evaluation.

6.3 Methods

The input to the system are two monolingual entailment graphs, the construction of which we described in the earlier chapters. Due to the lack of a German fine-grained entity typing system at the time that these experiments were conducted, we instead used the 4 types provided by the Stanza named entity recognition component and a catch-all type for unrecognized entities (person, location, organisation, event and thing). We project the 35 different types used in the English entailment graphs onto the 5 types of the German graph, combining for example English graphs of type pairs like #organisation#product and #organisation#art into a single graph of with the type pair #organisation#thing. The output of our system is a multilingual entailment graph, in which predicates that are translations of each other are aligned.

For this we use the method of multilingual vector space alignment with Word2Vec embeddings trained with fastText (Bojanowski et al., 2017). As training data for the English word embeddings we use the English NewsSpike corpus (Zhang and Weld, 2013), and for the German embeddings the German part of the NewsCrawl corpus (for more information, see chapter 3.2). We use the data that the respective entailment graphs were constructed from as training data for the word embeddings to ensure that the vocabulary of the entailment graphs is covered by the embeddings.

We used the bilingual dictionary induction algorithm MUSE (Lample et al., 2018) to align the two monolingual word embeddings into one multilingual word embedding. This approach builds upon the assumption that there is a linear mapping between word embedding spaces in different languages. Starting from a small dictionary of anchor translations the system uses a generative adversarial approach to map the source embeddings (in our case English) into the vector space of the target embeddings (in our case German). This way both semantically similar words in the source language and their translations in the target language are located close to each other in the resulting vector space. We ran preliminary experiments with different seed dictionaries and found string matches between English and German (mostly named entities) to produce the best alignment.

At the end of this step we have a vector representation of our German and English input vocabulary in which words that are semantically similar are close together in vector space. In the next step, we need to decide which predicates in the German graph align with which predicates in the English graph. We do this by traversing all predicate nodes of the German graph and determining the k nearest English neighbours of the German predicate using the multilingual vector space we created earlier. A node can contain more than one predicate, with all predicates within one node being paraphrases of each other. We merge the German node with every English node that contains one of the k nearest neighbours. This merging process eliminates all other links between predicates within one node. The result of this process is the multilingual entailment graph.

6.4 Experimental Setup, Evaluation and Results

Multilingual entailment graphs do not differ in format from monolingual entailment graphs. This enables us to use the same evaluation corpora and evaluation code as for monolingual graphs, i.e. the German translation of the Levy Holt data set (Levy and Dagan, 2016a) (for more information on the evaluation, please consult chapter 5.3.). We compare the performance of three different entailment graphs: The monolingual German entailment graph, the aligned multilingual entailment graph and a German translation of the English entailment graph. For this translation of the English entailment graph we used the same aligned multilingual vector space described earlier, and assigned each predicate in the English graph its nearest German neighbour.

Entailment Graph	Precision	Recall
Translated max λ	0.3912	0.3038
Translated min λ	0.2842	0.5501
German max λ	0.5437	0.2382
German min λ	0.2526	0.7617
Aligned max λ	**0.5709**	**0.2404**
Aligned min λ	0.2527	0.7647

Figure 6.1: precision and recall values for different test scenarios. The aligned entailment graph performs best at the maximal threshold value, albeit only by a slight margin.

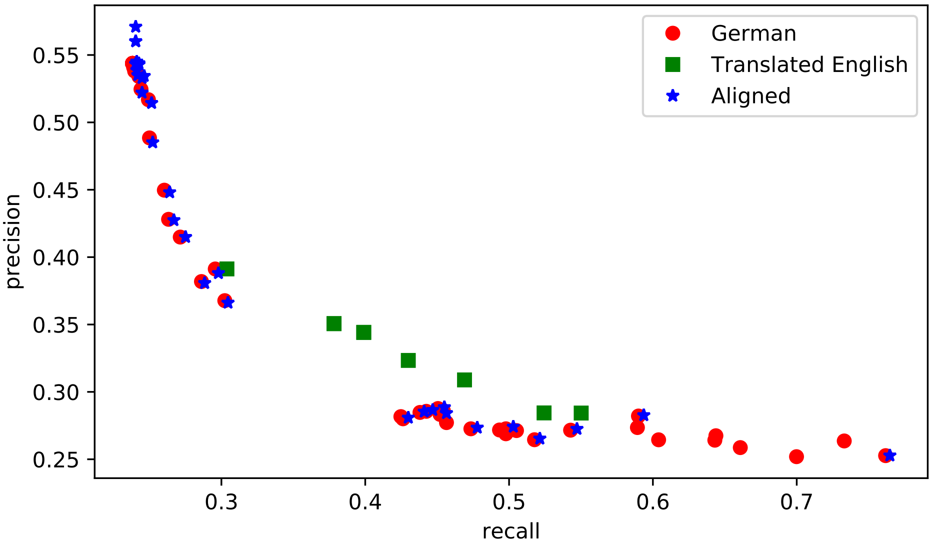


Figure 6.2: Precision Recall graph on the German data set across different confidence parameters λ. The aligned entailment graph performs only slightly better than the German graph.

Because the output of the entailment graph building algorithm only provides us with entailment scores between predicates but not with a threshold value for when to consider an entailment score high enough, we report the precision and recall at different threshold values. Figure 6.1 shows the precision and recall values of the three tested graphs at the highest possible and the lowest possible threshold value λ. We can see that the multilingual entailment graph performs up to 3 percent in precision over the German entailment graph. Figure 6.2 shows the precision recall curve at different λ values. There we can see that the multilingual entailment graph consistently achieves a small amount more precision and recall than the German graph, especially achieving more precision than the German entailment graph in a low recall scenario. The translation of the English entailment graph has more recall and precision in low precision scenarios, but never reaches the precision of the monolingual German or the multilingual Graph. When comparing testing the aligned multilingual entailment graph on the English Levy Holt data set, the multilingual graph does not perform as well as monolingual English graph. For a precision recall curve of the English graph, see Figure 5.2.

We want to again stress the specific setup that leads us to these results: We use only coarse grained entity typing and only named entities. This is an important point to keep in mind and we will elaborate on this point further in the following section.

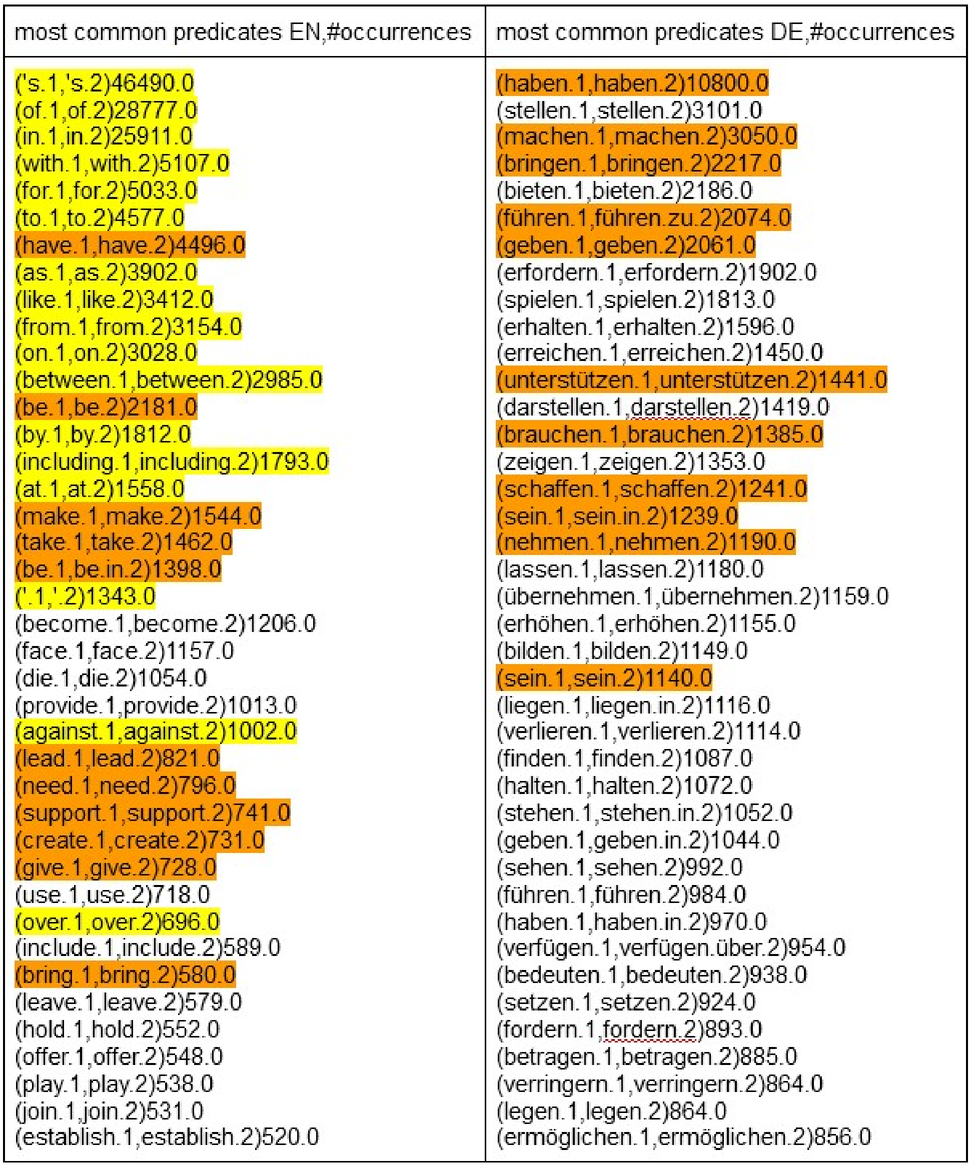
6.5 Manual Analysis of Predicate Alignment

The results presented here are quite weak. Moreover they do not support the hypothesis that combining the English and German entailment graph leads to a multilingual graph that is of higher quality than either monolingual one. While the multilingual graph performs slightly better than the German one on German test data, it performs much worse than the English graph on English data (see chapter 5.4). This opens up the questions of why this happens and how to proceed further.

There are many possible answers for the question of why: Previous chapters have covered the shortcomings of the German relation extraction, typing and evaluation. The results presented here come with the additional caveat of using an ill-fitted type system. The method of multilingual vector space alignment (also called binary dictionary induction) first presented by Artetxe et al. (2017) has since been criticised for the assumption that there always exists a linear mapping between to word embedding spaces (Vulić et al., 2020). Moreover, since this work has been conducted, contextualised multilingual word embeddings have become widely available, functionally replacing Word2Vec style embeddings in many high resource languages like English and German. And there is another point of criticism of the method: We know by now that the relations in the German and English entailment graphs are very different, but our alignment algorithm assigns every node in the German graph an aligned node from the English entailment graph, even though it seems more likely that in many cases a corresponding node does not exist.

This leads us to the question of how to move forward towards the creation of a multilingual entailment graph. Word2Vec style embeddings could be replaced with contextualised ones, and the alignment algorithm could be refined to keep certain nodes unaligned. But one important question that needs to be answered is the following: Is it even possible to successfully align entailment graphs? To answer this question we conduct a manual analysis of the German and English entailment. Approach differs from the one presented in the previous chapter in two ways: 1) Instead of using coarse grained entity typing and only named entities, we now use our system for the typing named and general entities. 2) The entailment graphs that we tried to align in the previous experiments started from input data that was different in size and content, which makes it difficult to judge how much of our results are due to the input and how much due other factors. To start the graphs on an equal footing we take the news commentary part of the WMT corpus (Mathur et al., 2020). It consists of 9K financial news commentary articles that were translated from English to German, resulting in parallel sentences.

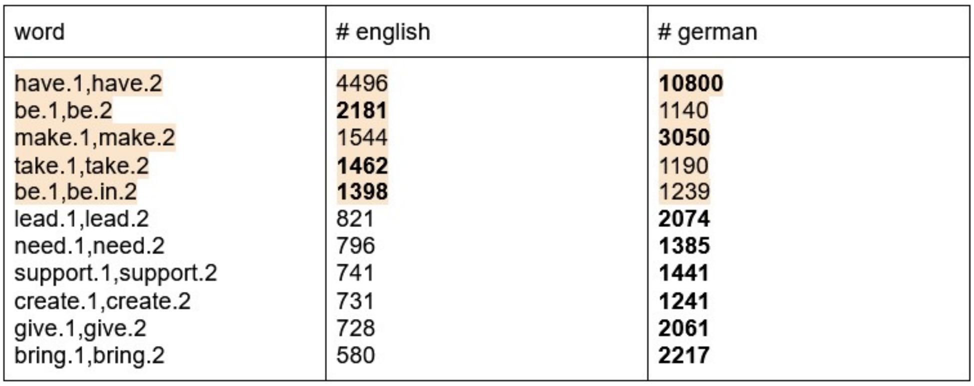
We process the English half with the English relation extraction pipeline and the German part with the German relation extraction pipeline (which now includes fine-grained entity typing). We then use (Hosseini et al., 2018)'s graph building algorithm to construct entailment graphs in German and English.



most common predicates EN,#occurrences	most common predicates DE,#occurrences
('s.1,'s.2)46490.0	(haben.1,haben.2)10800.0
(of.1,of.2)28777.0	(stellen.1,stellen.2)3101.0
(in.1,in.2)25911.0	(machen.1,machen.2)3050.0
(with.1,with.2)5107.0	(bringen.1,bringen.2)2217.0
(for.1,for.2)5033.0	(bieten.1,bieten.2)2186.0
(to.1,to.2)4577.0	(führen.1,führen.zu.2)2074.0
(have.1,have.2)4496.0	(geben.1,geben.2)2061.0
(as.1,as.2)3902.0	(erfordern.1,erfordern.2)1902.0
(like.1,like.2)3412.0	(spielen.1,spielen.2)1813.0
(from.1,from.2)3154.0	(erhalten.1,erhalten.2)1596.0
(on.1,on.2)3028.0	(erreichen.1,erreichen.2)1450.0
(between.1,between.2)2985.0	(unterstützen.1,unterstützen.2)1441.0
(be.1,be.2)2181.0	(darstellen.1,darstellen.2)1419.0
(by.1,by.2)1812.0	(brauchen.1,brauchen.2)1385.0
(including.1,including.2)1793.0	(zeigen.1,zeigen.2)1353.0
(at.1,at.2)1558.0	(schaffen.1,schaffen.2)1241.0
(make.1,make.2)1544.0	(sein.1,sein.in.2)1239.0
(take.1,take.2)1462.0	(nehmen.1,nehmen.2)1190.0
(be.1,be.in.2)1398.0	(lassen.1,lassen.2)1180.0
('.1,'.2)1343.0	(übernehmen.1,übernehmen.2)1159.0
(become.1,become.2)1206.0	(erhöhen.1,erhöhen.2)1155.0
(face.1,face.2)1157.0	(bilden.1,bilden.2)1149.0
(die.1,die.2)1054.0	(sein.1,sein.2)1140.0
(provide.1,provide.2)1013.0	(liegen.1,liegen.in.2)1116.0
(against.1,against.2)1002.0	(verlieren.1,verlieren.2)1114.0
(lead.1,lead.2)821.0	(finden.1,finden.2)1087.0
(need.1,need.2)796.0	(halten.1,halten.2)1072.0
(support.1,support.2)741.0	(stehen.1,stehen.in.2)1052.0
(create.1,create.2)731.0	(geben.1,geben.in.2)1044.0
(give.1,give.2)728.0	(sehen.1,sehen.2)992.0
(use.1,use.2)718.0	(führen.1,führen.2)984.0
(over.1,over.2)696.0	(haben.1,haben.in.2)970.0
(include.1,include.2)589.0	(verfügen.1,verfügen.über.2)954.0
(bring.1,bring.2)580.0	(bedeuten.1,bedeuten.2)938.0
(leave.1,leave.2)579.0	(setzen.1,setzen.2)924.0
(hold.1,hold.2)552.0	(fordern.1,fordern.2)893.0
(offer.1,offer.2)548.0	(betragen.1,betragen.2)885.0
(play.1,play.2)538.0	(verringern.1,verringern.2)864.0
(join.1,join.2)531.0	(legen.1,legen.2)864.0
(establish.1,establish.2)520.0	(ermöglichen.1,ermöglichen.2)856.0

Figure 6.3: the 40 most common extracted relations in English and German. Non-predicate relations are highlighted in yellow. Relations that are translations of each other are highlighted in orange.

While the input to these steps is parallel text in English and German, the extracted relation triples and entailment graphs look very different. This first shows up in the



word	# english	# german
have.1,have.2	4496	**10800**
be.1,be.2	**2181**	1140
make.1,make.2	1544	**3050**
take.1,take.2	**1462**	1190
be.1,be.in.2	**1398**	1239
lead.1,lead.2	821	**2074**
need.1,need.2	796	**1385**
support.1,support.2	741	**1441**
create.1,create.2	731	**1241**
give.1,give.2	728	**2061**
bring.1,bring.2	580	**2217**

Figure 6.4: Comparison of predicate extractions from parallel English German data. While there is no clear trend for light verbs, non-light verbs are extracted more often in German than in English.

statistics of the output of the relation extraction step: in English about 100 K relations more are extracted than in German (EN 558 476, DE 438 318). There are also qualitative differences between extracted relations: Table 6.3 shows the 40 most common relations in English and German. In English the most common relations are not predicates but prepositions (highlighted in yellow). While these prepositions also show up as part of compound expressions and predicates in both languages, it is worth noting that the prepositions listed here do not fall into that category, but are singleton occurrences that are not part of any overarching constructions. This high number of prepositions is due to the CCG parser that is part of the English relation extraction pipeline. Because we use a dependency parser in the German pipeline, these are not extracted in German (for more details on the German relation extraction pipeline, see chapter 3). Instead, in the German output prepositions only occur as part of predicates that contain at least one verb and never as singletons. This means that in the case of an alignment between English and German, the most common English extractions must stay unaligned to the German graph.

The most common actual predicates in both languages are light verbs. Again, while light verbs occur as part of larger constructs, the light verbs listed here have been extracted by the respective pipelines as singletons and not as parts of larger constructs. We can see that only for some of the light verbs in English there are translations in German within the 40 most common predicates (translations are highlighted in orange). This is surprising because these relations were extracted from parallel text, which lets us assume that all predicates in English should have translations in German, and that English predicates and their translations should occur the same amount of times. But

Table 6.4 shows that the number of verb occurrences is different. While for the light verbs *have*, *be* and *make* there is no clear trend, the proper verbs *lead*, *need*, *support*, *create* and *bring* are extracted more often in German (often more than twice as much).

The reasons in this mismatch can lie either in the difference in the German and English relation extraction pipeline, or in the translation of the English text into German. It is possible that the English pipeline fails to recognize when a light verb is part of a light verb construction and therefore only extracts a light verb singleton, while the usage of a light verb construction database (Krenn, 2000) in the German pipeline leads to better recognition of light verb phrases in German, and in consequence less extractions of German singleton light verbs. Another possibility is that English light verb phrases are more often translated as proper verbs into German, leading to the higher amount of proper verbs in the German output. Both of those factors hamper a proper alignment between predicates in the graph.

After the relation extraction step, the resulting relation triples are passed to the language agnostic graph building algorithm. The graph building algorithm discards predicates that are extracted less than 3 times. If we apply this cut off, the amount of relations extracted in German and English is more similar, with 387 948 English and 350 659 German relations. This suggests that the English text has more rare relations than the German text. This may be due to less variability of predicates in the translations than in the English original. The resulting English and German entailment graphs vary in size a lot, e.g. the English location#event graph contains 23 predicates while the German location#event graph contains 486. This tendency is fairly consistent across graphs. This may be due to the different approach to typing in the German and English relation extraction pipeline. While both pipelines use FIGER types, the English pipeline assigns types via linking to a database. When a relation can not be found in the database, the additional type ‘thing’ is assigned. The German system uses an end to end model, that assigns FIGER types even to out of vocabulary words (for more details, see chapter 4). The additional catchall type ’thing’ in English leads to the creation of more English graphs which can not be aligned to any German graphs, and to smaller English graphs in the other types.

Judging from this manual analysis we can see that the differences in the relation extraction pipelines accumulate and lead to the problem that even graphs created from parallel text are very different both quantitatively and qualitatively. If an alignment of entailment graphs builds upon the assumption that similar predicates in different languages have similar occurrence statistics in the extracted relations and constructed entailment graphs (like in Lewis and Steedman (2013a)), then this manual analysis shows that assumption is incorrect. We therefore need to consider other sources of information for the alignment of English and German predicates. These can be found for example in the additional used of Wikipedia for alignment (Wu et al., 2021), or in the usage of multilingual contextualised word embeddings, which we will examine in chapter 7.

6.6 Conclusion

In this chapter we examined the possibility of aligning the English entailment graph with the German entailment graph using no external sources of knowledge, but only the entailment graphs themselves and the input data they were created from. We started by building monolingual Word2Vec style word embeddings from the same input corpora that the respective entailment graphs were created from, and combined these embeddings into a bilingual embedding space. Using adversarial generative training and a seed dictionary we made sure that translations of words would be close to each other in the bilingual vector space. We then used this vector space to determine which nodes of the German entailment graphs should be aligned with which nodes of the English entailment graphs.

We found that this approach leads to weak results. The assumption that there is a linear alignment between two monolingual word embedding spaces has since been proven as problematic. Moreover, our assumption that every node in the German graph can be aligned with a node in the English graph has proven false, too. We therefore conducted a in depth manual analysis of the extracted relations and subsequent entailment graphs constructed from parallel text. One conclusion of this manual analysis is that entailment graphs generated from parallel text are not very similar. Aligning them even in this best case scenario using only the information contained within them is not a trivial task. We uncovered several complicating factors, like differences in the relation extraction pipeline and typing. We can therefore assume that additional knowledge beyond the entailment graphs themselves is needed to create a multilingual representation.

We collaborated with Wu et al. (2021) to approach the task of creating a multilingual entailment graph in English and Chinese by employing additional data from Wikipedia. Wu et al. (2021) create a relation and an entity-centric graph that contains both relation triples extracted from news text by the relation extraction pipeline and relation triples from Wikipedia. They then use a cross-graph guided interaction mechanism to encourage interactions between the two graphs to obtain better predicate and entity representations for alignment, effectively using entities that appear both in English and Chinese as anchors. The resulting predicate vectors can then be aligned by a simple distance metric. This way they manage to create a multilingual entailment graph for English and Chinese, overcoming the difficulties brought on by relation extraction pipelines in different languages using Wikipedia data as an additional training signal.

We also collaborated with Li et al. (2022) on creating a Chinese entailment graph, using similar methods for fine-grained entity typing as the ones we employed for German. Li et al. (2022) evaluate the results of their Chinese entailment graph in combination with the English entailment graph. They do this by generating an aggregated entailment score that uses both graphs. They show that the aggregated score performs better than each of the monolingual scores. By aggregating scores they make use of the different relations in the different language graphs without explicitly aligning the two graphs. Their approach requires the test set to exist in both English and Chinese, so that a look-up in both graphs is possible, which is different from a use case where a query in one language could be answered with knowledge from multiple languages, as it is the case in Wu et al. (2021)'s approach. Rather than requiring an alignment on the side of the graph, the alignment is performed on the test side, using additional information in form of a machine translation system, translating the original English test set into Chinese.

Another way to look at this problem is from the application perspective: If our use case is the detection of predicate entailments in German and English, an explicit aligned representation of German and English entailment graphs might not be needed. We could instead use the German and English entailment graphs to generate training data for a supervised multilingual entailment detection system, that builds on contextualised multilingual word embeddings. Using pretrained multilingual contextualised word embeddings could provide us with the additional knowledge needed to enable transfer learning from English and German entailment graph data. It is this approach that we will discuss in more detail in the following chapter.

Chapter 7

Implicit alignment via training

7.1 Introduction

The previous chapter has shown the difficulties faced when attempting to create multilingual entailment graphs. But even monolingual entailment graphs come with the weakness of sparsity. Their low recall limits the options for downstream applications. In the past there have been several approaches to address the problem of entailment from a different angle. Building on the power of contextualised word embeddings, there have been approaches that framed entailment detection as a sentence-based supervised classification task. These models classify two sentences as either entailed, contradicting or neutral based on sentence representations derived from contextualised word embeddings. These approaches come with their own problems, like requiring a large amount of human annotation, low inter-annotator agreement (Pavlick and Kwiatkowski, 2019; Chen et al., 2020b), and models picking up annotation biases rather than learning useful information from the semantics of the text (McCoy et al., 2019).

Nevertheless, it might be worthwhile to think of a combination of these approaches with entailment graphs to mitigate the problem of entailment graph sparsity and low recall. A model based on contextualised word embedding can draw from pretraining on large corpora and produces an output even for out of vocabulary words. Representing the supervised learning approach to the problem, Schmitt and Schütze (2021a) train a model to classify two sentences as either entailed or non-entailed using Hearst patterns and the language model RoBERTa. The model achieves state-of-the-art results on the Levy-Holt data set (Levy and Dagan, 2016a) when training on a small subset of it. But training on a different data set has a strong negative impact on Levy-Holt test set performance (Schmitt, 2021), which suggests that this approach is quite brittle and might pick up annotation biases rather than acquiring useful knowledge.

We therefore train Schmitt and Schütze (2021a)'s model using high confidence examples from the entailment graph instead of a part of the Levy-Holt data set. Our hypothesis is that this new approach has several advantages over the earlier approaches. It has the potential to mitigate the sparsity problem of the entailment graphs, while also not training on the same data set that we are testing on. This way the model will not pick up annotation biases from the test set. Our previous work on fine-grained entity typing has shown that models building upon contextualised word embeddings are able to generalise well from automatically generated data. Provided with high-confidence examples taken from the entailment graphs, a model might learn not only a representation of the predicate but also a representation of the context the predicate appears in. If a specific predicate does not occur in the entailment graph and is therefore not seen at training time, the contextualised embedding of the sentence it appears in can nevertheless be telling enough for a neural model to make a correct entailment judgement at test time.

This approach also opens the door for multilingual applications. While Schmitt and Schütze (2021a) only consider entailments in English, the usage of XLM-RoBERTa (Conneau et al., 2019) as an underlying language model allows us to apply their method in a multilingual scenario, where we provide both German and English examples during training. Similar approaches have been shown to work well in the task of open domain relation extraction (e.g. Harting et al. (2020); Ro et al. (2020)). This multilingual approach is similar to our earlier attempt of aligning entailment graphs, but rather than performing an explicit alignment it uses the entailment information contained in German and English graphs implicitly. Multilingual training might enable the model to learn entailment information across languages without needing an explicit alignment of entailment graph relations.

We evaluate our results using both the English Levy-Holt data set and a translation of it into German. We also evaluate the robustness of our method by using the SherLIiC data set of Schmitt and Schütze (2021a) and a machine translated German version of it. This offers the benefit of not needing to spend time and resources on human annotation of a new test set. We compare this model against the German and English monolingual entailment graphs, and models trained using only Schmitt and Schütze (2021a)'s method.

7.2 Method

Model Architecture Schmitt and Schütze (2021a) present a model that builds on a large language model, i.e. the hugging-face implementations of RoBERTa-base and RoBERTa-large. They use a classification layer in combination with so called *patterns* that connect the premise and hypothesis provided in the training data. The model then classifies the two connected sentences as either entailed or non-entailed. The patterns are natural language words or phrases that encode entailment between two sentences e.g. "sentence 1 *because* sentence 2". Schmitt and Schütze (2021a) compare manually constructed patterns against patterns that they extract automatically from large corpora of text but find that the manual patterns perform better. This is why we only use the manually constructed patterns in our experiments. They also use antipatterns that encode the relation of two sentences not being entailed, e.g. "sentence 1. *This does not mean that* sentence 2".

Adaptation to German and Multilingual Data To adapt their system to the task of German and multilingual predicate entailment detection we need to exchange the monolingual English language model RoBERTa used by Schmitt and Schütze (2021a) for the multilingual XML-Roberta (Conneau et al., 2019). We also need to add German patterns in addition to the existing English patterns. We do this by manually translating the English patterns into German.

Training Data Generation While Schmitt and Schütze (2021a) train on splits from the Levy-Holt and SherLIiC data sets, this approach has been shown to produce models that perform well on the data set they were trained on, but that perform worse on test data from a different data set (Schmitt, 2021). Our hypothesis is that Schmitt and Schütze (2021a)'s training method overfits on the relatively small (about 4500 sentences large) data set. To address this problem we use the entailment graph to generate our training data. This way we create a model that does not see any data from the Levy-Holt data set at training time, and therefore can not pick up any annotation biases from it. We generate training data from the German entailment graph and the English entailment graph built from News Crawl data. We decide for the English News Crawl graph over the News Spike graph, because the News Crawl graph is generated from more data and shows better performance on the Levy-Holt test set (for more details, see chapter 5.4.) Generating training data from the entailment graphs requires us to make several decisions:

1. *Threshold value:* We need to decide how high the entailment score between

two predicates needs to be to consider two predicates as entailed. As stated earlier, high confidence examples have turned out to be very helpful in training models based on large language models, because they generalize well based on the context information contained in the example (for more details, see chapter 4). To better compare against training with the training split of the Levy-Holt data set (about 4500 sentences), we choose the confidence thresholds so that the amount of data matches this size, choosing slices of about 4500, 9000, 13500 and 18000 sentences. The number of sentences does not exactly conform to these numbers, because the entailment graphs at a certain entailment score threshold do not cut off at exactly these numbers. Nevertheless, the numbers are within 100 sentences of these targets.

2. *Entailment graph selection:* Each type pair has its own entailment graph (e.g. the graph for the type pair #person#location might contain different predicates and different entailment scores from the graph for #person#person). Due to the granularity of the types some graphs are rather large and others are rather small, containing as little as 4 or as much as 1000 different predicates. We choose to only use graphs with types that appear in the Levy-Holt data set. This way we make sure that no additional information is introduced by subjects and objects a predicate appears with, which should make our approach more comparable to Schmitt and Schütze (2021a)'s approach.

3. *Entity selection:* This question follows from the previous point. While the entailment graphs contain only types and predicates (e.g. buy#person#product entails own#person#product), the Levy-Holt data set contains sentences with entities as subject and object. The sentence pairs in the data set only differ in the predicate, with the subject and object staying the same in each of the sentences. To avoid adding named entities as an additional source of information for the model, we only use the named entities in our training data generation process that are contained in the Levy-Holt training data set. We obtain these entities by running the Stanza NER component (Qi et al., 2020) on the Levy-Holt data set.

4. *Inflection:* Because the entailment graphs contain only the lemmatized form of a predicate (e.g. *buy*#person#product), we need to inflect the lemmatized forms of predicates to from natural language sentences (e.g. John *buys* an apple). For English we are using the python library mlconjug and for German we use the winning system for the task of inflection at SIGMORPHON 2018 (Kementchedjhieva

et al., 2018).

Hyperparameter Finetuning We find that replicating the results that Schmitt and Schütze (2021a) report for English is not possible without significant fine-tuning of the learning rate, weight decay and gradient aggregation steps. For example, swapping the model's underlying language model from English RoBERTa to the multilingual XLM-RoBERTa and training with the same data and hyperparameters as described in their paper results in a 22% drop in F1 score on the Levy-Holt test set. To replicate Schmitt and Schütze (2021a)'s hyperparameter tuning process, we train 50 models with random configurations of the learning rate, weight decay and gradient aggregation steps within the thresholds that the authors name in their paper. We achieve results within 5 percent of their results using this method. The authors trained 500 models with random configurations for finetuning, but this method is outside of the scope of time and hardware that is available to us.

7.3 Experiments

In our experimental setup we train Schmitt and Schütze (2021a)'s model with different training data that is either human annotated (Levy-Holt, SherLIiC) or generated from the German and English entailment graphs. We choose the data so that we can answer the following questions about the model: Are the positive results of Schmitt and Schütze (2021a) due to overfitting on the small Levy-Holt and SherLIiC data sets? Does enhancing manually annotated data with automatically generated entailment graph data lead to increased performance? Is there a difference in model performance when these experiments are conducted with German instead of English data? How does the zero-shot cross-lingual performance of an English model compare to a model trained on German data? And lastly, does multilingual training increase performance?

Test Sets We follow Schmitt and Schütze (2021a)'s work and test the models on the test split of the Levy-Holt data set (Levy and Dagan, 2016b) (12,921 sentences) and the test split of the SherLIiC data set (Schmitt and Schütze, 2019) (2,990 sentences). More information about the Levy-Holt data set can be found in chapter 5.3. The SherLIic data set was created using a similar approach to Levy and Dagan (2016b). Schmitt and Schütze (2019) frame the annotation of predicate entailments as a question answering task, but they use Freebase and the entity-linked web corpus ClueWeb09 (Gabrilovich et al., 2013) instead of news text. Also, label distribution is different between the two data sets, with 20% positive labels in the Levy-Holt data set and 30% positive labels in the SherLIic data set. We also automatically translated these test sets to German, using the DeepL API and the Google Translate API.

Metrics Following previous work we evaluate the models' performance using the metrics Precision, Recall and F1 score. When comparing against the entailment graphs we use the area under the precision recall curve (for more details on this metric see chapter 5.4.).

7.3.1 Experiment 1: Robustness

This experiment compares the performance of a model trained and hyperparameter tuned on the train and dev splits of the English Levy-Holt data set with the performance of models trained with different amounts of data automatically generated from the English entailment graph. We test the models on the test split of the Levy-Holt data set and the test split of the SherLIiC data set. It is important to note that none of the models are trained on any data from the SherLIiC data set, which makes this data set a good example of what a model might perform like on unseen data, e.g. in a real world application scenario.

Models:

- LH : Trained on the 4388 lines long training split of the Levy-Holt data set. Each line of the training data contains a premise sentence, hypothesis sentence and a label indicating the sentences either as entailed (20% of all labels) or non-entailed (80% of all labels). We train 50 models with different configurations of 3 hyperparameters, choosing the model that performs best on the dev split of the Levy-Holt data set.
- EntGraph 4.5K : Trained on about 4500 lines automatically generated from English entailment graphs. The proportion of labels is the same as in the Levy-Holt train split. For details on the data generation, see chapter 7.3. We perform the same hyperparamether tuning procedure as for LH.
- EntGraph 9 K : Trained on about 9000 lines automatically generated from English entailment graphs, and using the same hyperparameter tuning procedure.
- EntGraph 13.5 K: Trained on about 13500 lines automatically generated from English entailment graphs, and using the same hyperparameter tuning procedure.
- EntGraph 18 K: Trained on about 18000 lines automatically generated from English entailment graphs, and using the same hyperparameter tuning procedure.

	LH	EntGr 4.5K	EntGr 9 K	EntGr 13.5 K	EntGr 18 K
F1 Levy	**78.39**	13.67	20.44	42.77	51.61
FI SherLIiC	21.06	12.70	30.49	52.87	**57.67**

Table 7.1: While performance for the model trained on Levy-Holt data (LH) is very different when testing on the Ley-Holt test split and the SherLIiC test split, the models trained on entailment graph data perform similarly on both test sets, with a model trained on 9 K lines of training data (EntGraph 9 K) outperforming LH on the SherLIiC test set. All results are given in %

Results As can be seen in Table 7.1 and Figure 7.1 the model trained and hyperparameter tuned on the Levy-Holt train and dev sets performs best on the Levy-Holt test split, but it performs over 50percent points worse in F1 score on the previously unseen SherLIiC test set. In contrast to that, the models trained on entailment graph data perform similarly across both data sets, with more training data improving model performance. While the EntGraph models do not outperform LH on the Levy-Holt test set, a model trained on only 9000 lines of data (EntGraph 9 K) outperforms LH on the SherLIiC test set. These findings support our hypothesis that the high performance of Schmitt and Schütze (2021a)'s model does not generalise to unseen data sets and might be due to the model picking up annotation biases of the Levy-Holt data.

Hyperparameter Stability Using the hyperparameter tuning process described earlier, we find that the hyperparameters that lead to best dev set performance in the models trained on entailment graph data are relatively stable across data set sizes. Models EntGraph 1, EntGraph 2 and EntGraph 4 share the same hyperparameter settings, and the number of gradient accumulation steps is the same across all models.

Ablations One reason why the model trained on Levy Holt data (LH) does not generalise could be that it over-fits to artefacts in the sentences of the Levy Holt data set. To gain more insight into the behaviour of LH we create an ablation test set, in which we delete the premise, and provide the model with only the hypothesis. The performance of the LH on this test set is very weak with 0.07% AUC, 0,14% F1 score, 100% Precision and 0.07% recall. This leads us to believe that the artifacts that the model over-fits to must be properties of both hypothesis and premise, because the model is not able to perform well by using the hypothesis alone.

Another reason for the big gap in performance of LH on the Levy Holt test set and the SherLIiC test set could be due to different amount of lexical overlap between the Levy Holt training set and the respective test sets. While there 983 predicates appear in

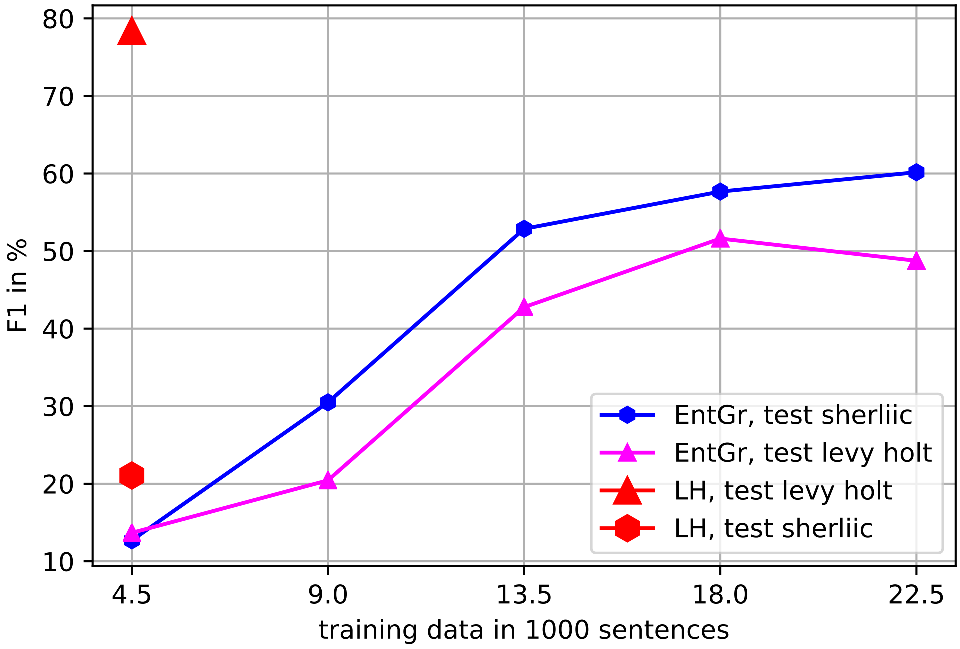


Figure 7.1: F1 scores of systems trained with different slices of entailment graph data (EntGr), tested on the SherLlic and Levy-Holt data sets. Performance increases with additional training data.

both the Levy Holt training set and the Levy Holt test set, only 20 of the predicates in the Levy Holt training set appear in the SherLIiC test set.

7.3.2 Experiment 2: Data Augmentation

While the previous experiment has shown that training with only the human annotated Levy-Holt train split leads to unsatisfactory results, this data set might still be useful when combined with data generated from entailment graphs. To test this hypothesis we train models on different proportions of automatically generated data and Levy-Holt data. We test these models on the Levy-Holt and the SherLIic test sets. We employ the same hyperparamerter tuning procedure as in the previous experiment.

Models:

- LH : Trained on the 4388 lines long training split of the Levy-Holt data set.
- Augmented 9 K: Trained on the Levy-Holt training split and approximately 4500 lines of automatically generated data. The data sets were concatenated and shuffled to make sure that the data from both sources is evenly distributed. We apply the same hyperparameter tuning procedure as before.

- Augmented 13.5 K : Trained on the Levy-Holt training split and 9 K lines of automatically generated training data.
- Augmented 18 K: Trained on the Levy-Holt training split and 13.5 K lines of automatically generated training data.

	LH	Augmented 9 K	Augmented 13.5 K	Augmented 18 K
F1 Levy	78.39	64.89	62.61	58.86
FI SherLIiC	21.06	29.95	44.53	51.19

Table 7.2: All models trained with a mix of Levy-Holt data and entailment graph data perform better on the Levy-Holt test set than on the SherLIiC test set. Adding entailment graph data increases performance on SherLIiC but decreases performance on Levy-Holt. All results are given in %

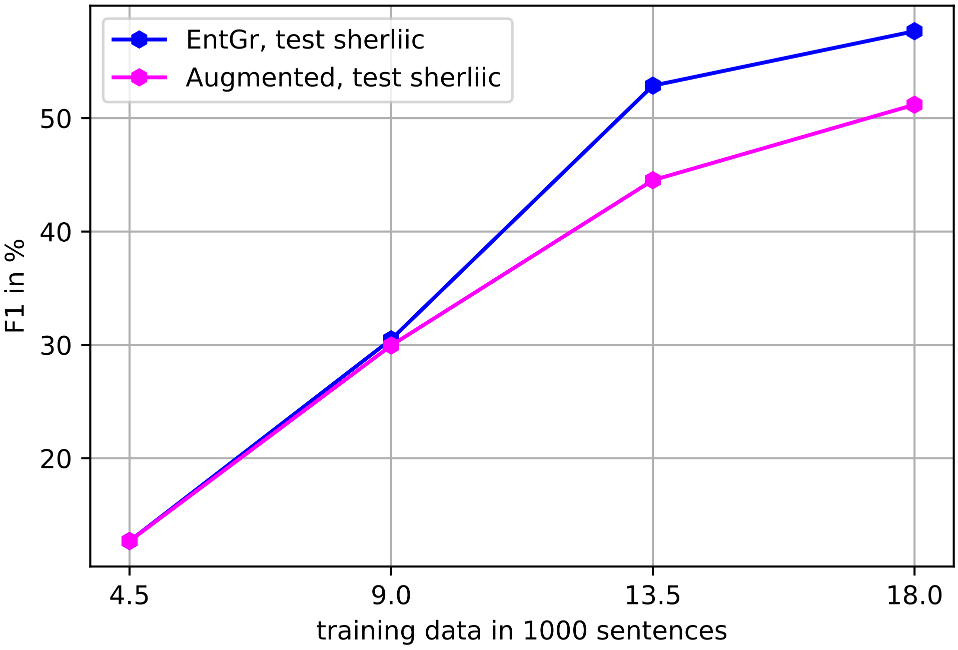


Figure 7.2: Models trained only on entailment graph data (EntGr) perform better than model trained on mixed data on the SherLIiC data set (Augmented).

Results As can be seen from Table 7.2, training with Levy-Holt data and entailment graph data together leads to overall higher performance on the Levy-Holt data set. While adding more entailment graph data decreases performance on the Levy-Holt test set, it increases performance on the SherLIiC test set. Contrary to our hypothesis, mixing entailment graph data with the Levy-Holt data set does not improve model performance on the SherLIiC data set. Figure 7.2 shows that using the same amount of pure entailment graph data performs better than mixed data. This is another pointer toward the possibility that the model picks up on annotation biases found in the Levy-Holt training data set, that later help the model to perform well on the Levy-Hot test set, but that are unhelpful when used on the formerly unseen SherLIiC test set.

7.3.3 Experiment 3: Zero-Shot Cross-Lingual Transfer vs. Training with German Data

Models building on multilingual word embeddings like XLM-RoBERTa can be trained in one language and applied to test data in another language. This is called zero-shot cross-lingual transfer. In chapter 4 we showed that for the task of fine-grained named entity typing in German a model trained on high-quality English data performs better than a model trained on noisy German data (Weber and Steedman, 2021b). While the hierarchical typing model by Chen et al. (2020a) performs well when used for zero-shot cross-lingual transfer, recent publications have examined the caveats of this approach for other tasks and models and come to the conclusion that it works best for closely related high-resource languages and for high level semantic tasks (Lauscher et al., 2020).

We want to examine the transfer learning ability of the Schmitt and Schütze (2021a)'s model by training it on German, English and mixed data and testing these models on the German translations of the Levy-Holt and SherLIiC test sets. While both data sets have been machine translated and therefore might contain noise, the results might nevertheless give us an estimate of how well different models perform.

Models

- LH-DE : Trained on the German translation of the 4388 lines long training split of the Levy-Holt data set.
- EntGraphDE 4.5 K, 9 K, 13.5 K and 18 K: Trained on the respective amount of lines generated from the German entailment graph
- EntGraphEN 4.5 K, 9 K, 13.5 K and 18 K: Trained on the respective amount of lines generated from the English entailment graph

Results Figure 7.3 is the German equivalent to Figure 7.1. It shows the performance of LH-DE on the Levy-Holt and SherLIiC test sets in comparison with the performance of the various EntGraphDE models. We can see a similar pattern to Figure 7.1: While the model that was trained on the translated Levy-Holt training data performs well on the Levy-Holt test set, it performs much worse on the SherLIiC test set. The models trained

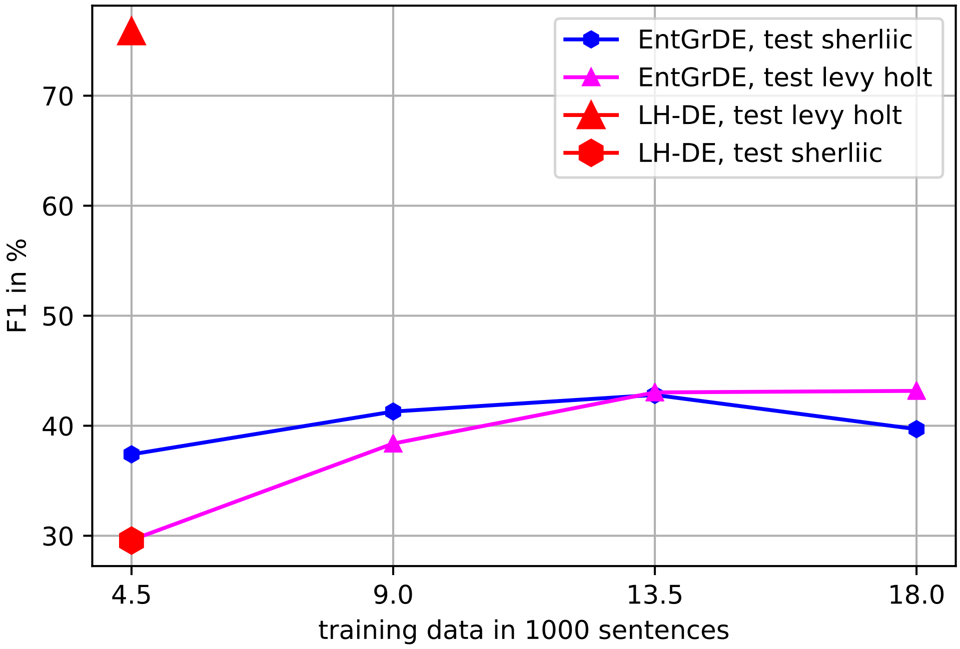


Figure 7.3: Performance of models trained with German entailment graph data (EntGrDE) and the German translation of the Levy-Holt data set. While the entailment graph model outperforms the Levy-Holt model (LH-DE) on the SherLIiC test set, overall performance of the German models is relatively low.

on entailment graph data perform similar on both test sets, outperforming LH-DE on SherLIiC at a training data size of 9 K sentences. Nevertheless, the overall performance of the EntGraphDE models is much lower than the performance of the EntGraphEN models on the English test data.

Figure 7.4 compares the performance of the EntGraphDE and EntGraphEN models on the German translation of the SherLIiC data set. We can see that even though the German models perform better at 4.5 K and 9 K sentences, the English models overtakes them at 13.5 K sentences, and performs nearly 10 percent points better in F1 score at 18 K sentences. This is surprising because the EntGraphEN models do not see any German data at training time. This high zero-shot cross-lingual transfer performance can be explained in two ways: First, the performance of the German models is quite weak, being about 10 percent points under the performance of the English model on English data. This is due to the lower quality of the German entailment graphs (for more details, refer to chapter 5) and due to possible weaknesses in the training data generation from German entailment graphs. These weaknesses stem from the difficulty of properly inflecting German predicates and more constraints in constructing grammatical sentences from lemmatized predicate pairs than in English. Second, as

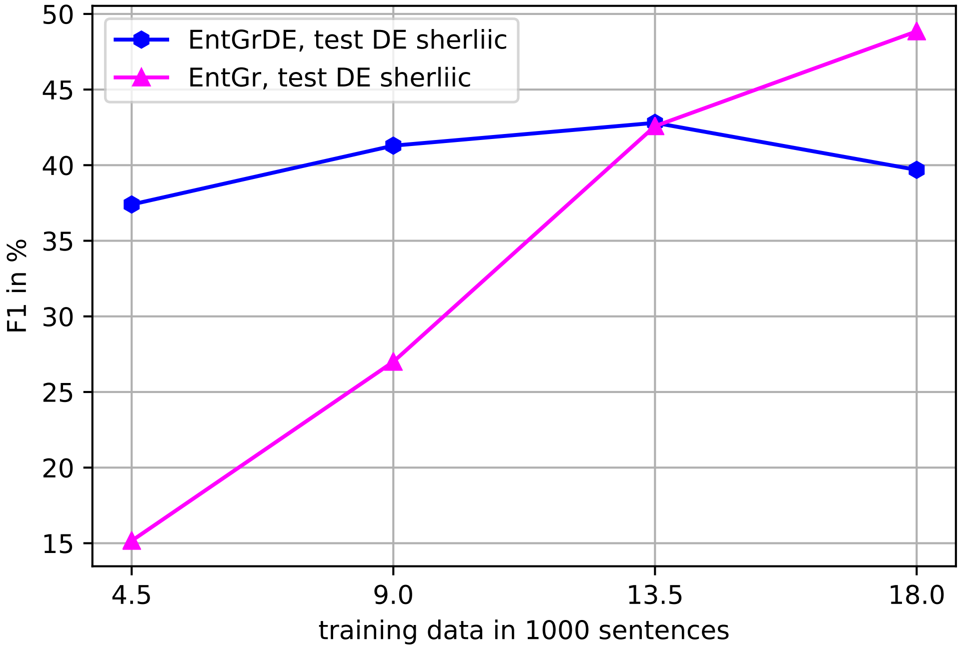


Figure 7.4: Performance of models trained on data from the English and the German entailment graph, tested on the German translation of the SherLIiC data set. The English models (EntGr) perform better than the German models (EntGrDE) even though they do not see any German data at training time.

we've shown in chapter 4, XLM-RoBERTa has very high quality embeddings for English and German, and therefore facilitates better transfer learning.

7.3.4 Experiment 4: Mixing German and English Data

The results of the previous experiment leave us with the question how to best involve German entailment graph data in the creation of a multilingual model. Discouragingly, a model trained on 18 K sentences of English entailment graph data performs better on a German test set than a model trained with the same amount of German data. The zero-shot model fulfills our goal of creating a multilingual predicate entailment detection system: The underlying language model XLM-RoBERTa enables us to process both German and English data at test time. But ideally, we would like our model to improve over this baseline by using data in German.

For this experiment we take a low resource scenario approach. We have shown that a model trained on 18 K sentences English entailment graph data has the best zero-shot cross-lingual performance. We now add different amounts of German training data to the 18 K sentences of English training data and train the model of Schmitt and Schütze

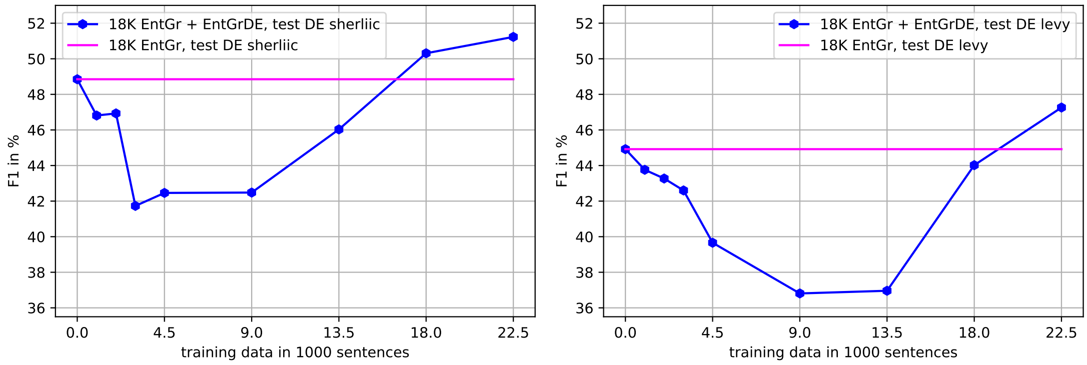


Figure 7.5: Performance of a model trained on 18 K sentences English data plus different amounts of German data (18K EntGr + EntGrDE), tested on German SherLlic (left) and Levy-Holt (right). While there is first a decrease of performance with the addition of German data, the mixed data model eventually performs better than zero-shot (18K EntGr).

(2021a) on that data. We do not do any additional hyperparamter tuning, but keep the hyperparameters of the best performing zero-shot model. This way we might be able to retain the high performance of the zero-shot model but also learn additional information from the German data. We evaluate on the German translations of the SherLIiC and the Levy-Holt data sets.

Results Figure 7.5 shows the comparison between a model trained only on 18 K sentences English data and a model trained on 18 K sentences English data and different amounts of German entailment graph data tested on the German SherLIiC and Levy-Holt data sets. The mixed model performs better than the zero-shot model at 18 K sentences of German data on the SherLIic data set and at 22.5 K German sentences on the Levy-Holt data set. Figure 7.6 shows the performance of the mixed models on English test data. Similar to the performance on the German test sets, the mixed models start perform better than the English only model at 18 K German sentences. This confirms our hypothesis that the model is learning different entailments from English and German, amounting to a higher number of learned entailments in total.

The mixed models beat the model trained with 18 K English sentences only by a small margin. This might be due to noise that exists either in the German entailment graph itself or in the German training data. Another possible explanation might be the lack of hyperparameter tuning in this experiment. Addressing these issues and exploring the models performance with regards to different hyperarameter configurations is an avenue for future work.

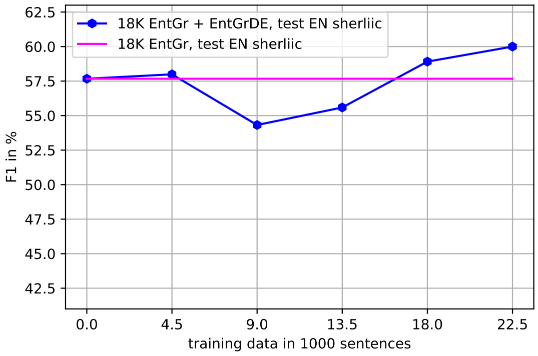

18K EntGr + EntGrDE, test EN sherliic
18K EntGr, test EN sherliic
18K EntGr + EntGrDE, test EN levy
EntGr, test EN levy
F1 in %
62.5
60.0
57.5
55.0
52.5
50.0
47.5
45.0
42.5
0.0
4.5
9.0
13.5
18.0
22.5
training data in 1000 sentences

Figure 7.6: Performance of a model trained on 18 K sentences English data plus different amounts of German data (18K EntGr + EntGrDE), tested on English SherLlic (left) and Levy-Holt (right). After an initial drop, the addition of German training data improves performance on the English test sets.

7.3.5 Experiment 5: The Directional Subset of the Levy-Holt Data set

In his Bachelors thesis Xavier Ricketts Holt et al. (2018) examined the data set by Levy and Dagan (2016a) and found a large amount of errors, which lead him to re-annotate the data set. This improved data set has been subsequently used by Hosseini et al. (2018), Schmitt and Schütze (2021a) and us under the name of 'Levy-Holt data set'. Another outcome of his work is the directional subset of the Levy-Holt data set. While the complete data set contains predicates that are paraphrases of each other as well as strictly directional predicate pairs, the subset only contains the directional pairs. The directional subset of the test split is accordingly smaller than the test split, containing only 1784 sentence pairs instead of 12921. We consider this data set to be more challenging than the full Levy-Holt data set, because the detection of paraphrases can be solved using semantic similarity, while the task of detecting directional entailment can not be solved that way. We therefore test the best performing English model (trained on 18 K sentences of entailment graph data) and the entailment graph that the training data was generated from on the directional subset of the Levy-Holt test split and compare it to the performance of the respective models on the full data set.

Results Table 7.3 shows the AUC, precision, recall and F1 values for the entailment graph and the trained model when tested on the full Levy-Holt data set and the directional subset. On the first glance the results seems quite counter-intuitive: when comparing the AUC values of the trained system to its F1 scores, the F1 score is lower

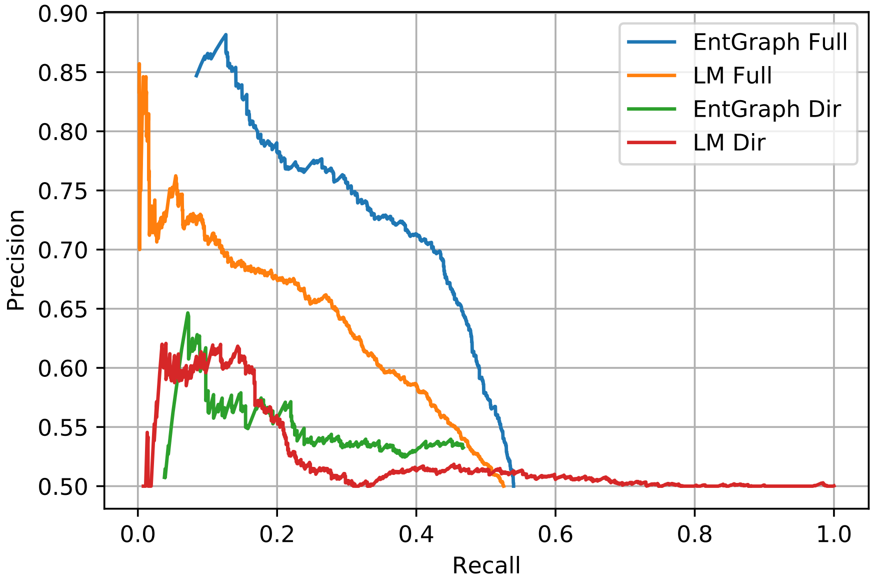


Figure 7.7: Precision recall curves of the English Entailment graph and the trained model when tested on the full Levy-Holt data set and the directional subset of the Levy-Holt data set. The curves for both the entailment graph and the trained model are relatively similar, with the curves for the directional subset reaching lower precision values.

for the directional subset than for the full data set, but the AUC is higher. This might be due to the threshold value that the trained model learns at training time. The model chooses the threshold with the best development set performance. The distribution of true and false labels in the training and development set it 20/80, while the label distribution on the directional subset is 50/50.

A clearer picture of the respective performance of the entailment graph and the trained model can be see in the precision recall curves in Figure 7.7. We can see here that the relatively high AUC for the trained model on the directional subset is due to the long tail of the precision recall curve, where the recall gets very high, but precision is close to random. The precision and recall graphs for both the entailment graph and the trained model on the directional subset look fairly similar. They do not reach the high precision values that are achieved on the full data set, with a gap of 20 percent points in precision for values around 40 percent points recall. Overall, there seems to be no clear winner in terms of performance on the directional subset and a closer examination of this problem is left for future work.

model	data set	AUC	Prec	Rec	F1
English entailment graph	Full	41.86	-	-	-
English entailment graph	Dir.	23.77	-	-	-
EntGraph 18 K	Full	33.67	49.39	54.04	51.61
EntGraph 18 K	Dir.	52.48	55.39	20.74	30.18

Table 7.3: A comparison between model performance on the full Levy-Holt data set and the directional subset of the Levy-Holt data set. Surprisingly, the trained model performs worse on the directional subset in F1 score, while the AUC is higher.

model	test lang.	AUC	Prec	Rec	F1
English entailment graph	EN	41.86	-	-	-
German entailment graph	DE	00.44	-	-	-
EntGraph 18 K	EN	33.67	49.39	54.04	51.61
EntGraphDE 18 K	DE	26.71	53.68	36.10	43.17

Table 7.4: Baselines and models tested on the Levy-Holt data set. The German trained model perform better than the entailment graph that was used to create the training data, but only the English model performs better than the majority class baseline.

7.3.6 Baselines

One factor that makes a comparison between Schmitt and Schütze (2021a)'s work and the entailment graphs difficult stems from solving the problem of predicate entailment with either a supervised or an unsupervised approach. As we described in more detail in chapter 5.4, Hosseini et al. (2018) report their results in precision recall curves and as the area under these curves (AUC), while Schmitt and Schütze (2021a) report precision, recall and F1. They also calculate an AUC value to compare against Hosseini et al. (2018) [1]. The AUC of the German and the English entailment graphs are one of the baselines that we compare against. We also present a majority class baseline. Because the Levy-Holt data set consists of 80% negative examples, predicting all sentences to be non-entailed constitutes quite a strong baseline in terms of precision, recall and F1 score. Sparsity is a big problem in entailment graphs, and our hypothesis is that a model that uses a large language model like XLM-RoBERTa might overcome this limitation.

[1] Preliminary experiments have shown that the AUC value given by the evaluation script of Schmitt (2021) is calculated using different cut offs than Hosseini et al. (2018), which is why we modified Schmitt (2021)'s code to output precision and recall values directly, and calculate the AUC from them.

Table 7.4 shows the performance of the best German and English model in comparison with the baselines. The English model trained on 18 K sentences automatically generated data does perform worse in AUC than the English entailment graph that it was created from. On the other hand, there is a big difference in the performance of the German model and the German entailment graph, with an increase by more than 26 percent points in AUC in the trained model. These results show two things: First, our hypothesis that using a model based on XLM-RoBERTa would improve performance over the entailment graphs has proven correct for a scenario where we have only a fairly small entailment graph, like in German. Second, despite the results that Schmitt and Schütze (2021a) present, the problem of predicate entailment is far from solved and further research is necessary to better combine the strengths of supervised models like Schmitt and Schütze (2021a)'s with unsupervised models like the one of Hosseini et al. (2018).

7.4 Conclusion

In this chapter we looked at using the entailment graphs as a source of training data for a supervised predicate entailment detection system. To do so we automatically created entailed sentence pairs from the predicates contained in the German and English entailment graphs. We used the model introduced by Schmitt and Schütze (2021a) and swapped out the monolingual contextualised word embeddings used by the authors for multilingual contextualised word embeddings to enable training with German and English data. The authors also use so called *patterns* that linguistically encode the entailment between two sentences during training (e.g. sentence 2 *because* sentence 1). To adapt the system to training with German data, we translated the English patterns to German.

Our experiments produced a number of interesting results. First, we confirm that the model of Schmitt and Schütze (2021a) does not generalise well to unseen data sets when trained on a subset of the human annotated Levy Holt data set. In contrast to that, training the same model with data generated from the entailment graph offers the model instances of predicate entailment that are not seen in the test sets, which prevents overfitting to annotation biases. A model trained with entailment graph data performs consistently across two unseen data sets.

We also show that the problem of predicate entailment is far from solved: Our best performing English model trained on 18 K sentences reaches under 52% in F1 score and many of the models trained on less data do not perform better than the majority class baseline (44% F1 score). On the other hand, the performance of the German model trained with entailment graph data is much higher than the performance of the respective entailment graph with approximately 26 percent points more AUC. This proves our hypothesis that using contextualised word embeddings alleviates the graph's sparsity problem for a scenario when only a small entailment graph is available.

When we look at our goal of creating a multilingual predicate entailment detection model, our experiments show a success: The zero-shot performance of a model trained on English entailment graph data is quite high, and improvements can be achieved on both German and English test sets by adding German data. This shows that the model is indeed able to learn different entailments from the different language graphs and arrive at a higher amount of total learned entailment relations, even though the quality of the German entailment graph is lower than the quality of the English entailment graph. The only caveat to this result is that the improvements over the English only model are quite low (about 3 percent points in all test cases). This might be due to the lower quality of the German entailment graph and noise added in the German training data generation. Nevertheless this leaves us with the optimistic outlook that these results might be superseded by the usage of higher quality entailment graphs, better data generating procedures, or by the addition of data from entailment graphs in other languages.

Since these experiments were conducted the authors of Schmitt and Schütze (2021a) released a new approach in which the lexical patterns connecting premise and hypothesis were learned by the model instead of being manually created. Due to time constraints, we do not repeat our experiments with the newer modification of this model. Drawing from the results shown earlier, we expect that even for the newer model training it on the Levy-Holt data set will lead to overfitting and lower performance on an unseen data set.

Chapter 8

Conclusion

8.1 Contributions

In this thesis we set out to create German predicate entailment graphs and use the information contained within them to create a bilingual predicate entailment representation with English. We also provide a detailed account of the challenges faced when adapting an approach from English to German and offer possible solutions. As in many a hero's journey, there were dead ends and the treasure is the friends found along the way: In our case, these are the German systems we have build and the knowledge we have gained about the possibilities and limitations of multilingual models.

The contributions of this thesis are the following: We constructed a **rule based German relation extraction system** and examined the success of our approach via a manual error analysis. We find that our system lacks many relations present in the English entailment graph because of differences in pipeline components and accumulating errors. We found entity typing to be a major bottleneck and we therefore created a **fine-grained entity typing model for German named and general entities**. This brought us two main insights: First, zero-shot cross-lingual transfer is a hard baseline to beat for German fine-grained entity typing. The parallel German-English data encoded in the contextualised word embedding XLM-RoBERTa enables a model trained on high quality English data to perform better on a German test set than a model trained on noisy German data. But second, a model for typing named entities can be enhanced to type both named and general entities by training it on data automatically generated from a German WordNet.

After extracting relation triples from German text, we used them to construct a **German entailment graph**, which we evaluated on a German translation of the Levy-

Holt data set (Levy and Dagan, 2016a). We aligned it with the English entailment graph in an explicit way by aligning predicate nodes with each other, but the performance of the resulting graph turned out to be quite weak. Each monolingual graph suffers from sparsity, and combining the graphs with our method does not alleviate this problem. We come to the conclusion that additional data is needed to create a multilingual predicate entailment representation. This lead us into two directions: First, we collaborated with Wu et al. (2021) to create an English Chinese entailment graph using additional information from Wikipedia. Second, we adapted Schmitt and Schütze (2021a)'s model that builds on contextualised word embeddings, making it trainable with data that we automatically generate from the German and English entailment graphs, creating a **supervised multilingual predicate entailment model**. We show that Schmitt and Schütze (2021a)'s original approach overfits to the relatively small human annotated Levy-Holt data set, and that training the model with entailment graph data leads to stable performance across two different unseen data sets.

This is in line with our earlier findings from German general entity typing: Using high confidence automatically generated examples as training data for a model that builds on contextualised word embeddings leads to good results because the model is able to learn from context and generalises well to unseen data. When training the model with multilingual data we find that once again zero-shot cross-lingual transfer constitutes a strong baseline: A model trained with 18 K English data performs nearly as well on a German test set as a model trained with 18 K English and 18 K German data, and better than the best model trained only with German data. Nevertheless we show that using Schmitt and Schütze (2021a)'s model in combination with German and English entailment graphs creates a multilingual entailment model that performs better on German and on English data than either German or English monolingual model. This confirms our hypothesis that a multilingual model can transfer the entailments learned from different language training data and this way acquire more entailments in total.

Our other contributions are the data sets that we make available for public usage. We created three novel German fine-grained named entity typing test sets and one German fine-grained general entity typing test set. We moreover translated the Levy-Holt and SherLIic test sets for predicate entailment into German.

8.2 Future Work

8.2.1 The Distributional Inclusion Hypothesis

Working on the components of the entailment graph building pipeline left us with directions for future work in many areas. In chapter 2 we cover the **distributional inclusion hypothesis** and point out that the experiments that quantified the distributional inclusion hypothesis for English nouns were never repeated for other languages or other parts of speech. Berant et al. (2011) and Hosseini et al. (2018)'s approach to lexical entailment in verbs falls short of Geffet and Dagan (2005)'s approach for nouns in two ways: they do not collect negative evidence via a web search for to make sure that missing examples are not due to sparsity in the corpus and they don't correlate the overlap of feature vectors with human judgements. The following points also complicate a replication of their experiments:

1. For their noun experiment Geffet and Dagan chose to use vectors of size 200 containing the most common features of their extracted nouns. This might work for the noun case, but our experience with verbs suggests that we need larger vectors to encode meaningful information (verb vectors have about 1000 features). Verbs are more sparse than nouns, so the range of subjects and objects they occur with might be bigger than the syntactic context that Geffet and Dagan considered for nouns. This is a key feature for the feasibility of the experiment: Geffet and Dagan did a web search for every combination of two nouns and a feature, to make sure whether presence or absence of the feature in the data was universal or only specific to the corpus, although they only sampled from this to keep the computational load low. They retrieved 3 000 additional sentences from the web for each triple of two nouns and a feature. It is up to future research to find an amount of features that is both informative and computationally feasible.

2. The English relation extraction pipeline extracts predicates, not verbs (see for example Figure 2.1). That means that we often have more than just one word, which makes our set of words quite large. Using only predicates that consist of a single verb might help us to keep the set of comparisons small. Geffet and Dagan prove their hypothesis using only 200 randomly chosen noun pairs, but it remains yet to be seen if 200 randomly chosen verb pairs will have enough entailment relations in them.

As a avenue for future work, we propose an experiment along the lines of the one performed by Geffet and Dagan (2005) using the entity-predicate triples extracted by Hosseini et al. (2018)'s English relation extraction pipeline. In this experiment, one would first filter the extracted predicates for the ones that consist of one verb only. One would then create large sparse feature vectors, where the features consist of the entity pair that the verb occurs with and a count of the frequency of that feature. One would apply point-wise mutual information at this point to balance high frequency entities and relations and take the top 1000 features of the vectors and perform the web search proposed by Geffet and Dagan (2005) with every triple of two verbs and a feature. The vectors would then be used to compute the correlation between feature overlap and human judgement. Both the web search and the manual annotation of human judgement are bottlenecks for the number of verbs and features to be considered in this experiment. If successful, this experiment could be expanded to verbs in German, using the German relation extraction pipeline.

8.2.2 German Open-Domain Relation Extraction

The shortcomings of rule-based **German open-domain relation extraction** pipeline presented earlier point to two possible directions for future work. The first one is the improvement of the rule-based system. Since the start of the project, a German CCG treebank for the Easy CCG parser has been released (Evang et al., 2019). A possible next step could be the replacement of dependency parses by German CCG parses, which would make this component of the German pipeline more similar to its English counterpart. It is unclear if the quality of German CCG parses would be comparable to the English parses used by Hosseini et al. (2018), and how much post-processing would be necessary. Additionally, the manual creation of a German gold data set for relation extraction with the use case of entailment detection would be necessary to better quantify the impact of different rules on the performance of the system. Overall, the quality of the English relation extraction pipeline is relatively poor (only 34% of extractions are correct), and applying its design to lower resource languages is likely lead to performance that is even worse.

Therefore, the second direction for future work is to step back from the pipeline approach and adapt the methods of open domain relation extraction to fit our requirements better than the currently existing systems discussed in section 3.3.1. Existing multilingual models use older neural approaches that do not specifically address relations that feature conjunctions or relations in sub-clauses. Because they forgo dependency parsing they produce erroneous output that could be prevented with simple linguistic constraints. The state of the art English model by Kolluru et al. (2020) addresses these problems in their model architecture.

One way forward would be to take the model architecture of Kolluru et al. (2020) and apply it to our use case, extending it to be trained with data in English and German. For this the English contextualized word embeddings in Kolluru et al. (2020)'s model need to be replaced with multilingual embeddings and a German part of speech tagger needs to be added. To generate training data we propose use the output from our own rule-based German and English relation extraction pipelines. This follows the approach of Cui et al. (2018), who use the output of a rule based open domain relation extraction system to generate training data for their neural system. Training on both English and German data would moreover enable transfer learning, possibly adding relations that are only seen in one language.

Numerous papers have stressed the difficulty of evaluation for the task of open domain relation extraction (Schneider et al., 2017; Lechelle et al., 2019; Han et al., 2020). There seems to be little consensus about data sets and metrics in the field, which makes it hard to compare systems to each other. Moreover, Lechelle et al. (2019) show that for some metrics a nonsensical baseline exploiting known weaknesses of the scoring function achieves higher results than any of the open domain relation extraction systems they tested. Because of this, it is important for our use case to step back from existing open domain relation extraction corpora and metrics and instead use data sets and metrics that are tailored to our requirements. We therefore propose to evaluate the German and English output by creating a manually annotated German and English test set that reflects the properties necessary for downstream processing of relation triples with the goal of constructing entailment graphs.

Another possible risk of this approach is the relatively low performance that state of the art systems achieve on their respective test sets and the big differences in performance on different test sets. The state of the art model of Kolluru et al. (2020) achieves accuracy scores between 26% and 48% and F1 scores 40% and 65% on the 4 data sets it was tested on. Although there is the possibility that adding training data in other languages might improve system performance, there is also the risk that adding another language might add more noise and therefore lead to even worse results. Nevertheless we believe that the proposed approach is one way to overcome the problems of the existing rule-based system.

8.2.3 German Fine-Grained Entity Typing

While **fine-grained named entity typing** in English is a well explored field, the lack of data sets makes this task less explored in other languages. Moreover, there is hardly any work on general entity typing. As an avenue for future work there is the application of existing models to a wider variety of languages and the further exploration of zero-shot cross-lingual transfer performance of different models. More details on future work there can be found in chapter 4.4.8.

8.2.4 Supervised Predicate Entailment Detection

With regards to training the **supervised predicate entailment detection** model of Schmitt and Schütze (2021a) a lot of the weak performance of the German model can be credited to the weakness of the German entailment graph and the German training data generation. But the big increase in AUC value of the German system as compared to using only the German entailment graphs shows the possibilities that open up when the strengths of entailment graphs are combined with the strengths of contextualised word embeddings. Future work could find an even better way of combining the two, creating entailment graph informed contextualised word embeddings, or using a combination of entailment graphs and contextualised word embeddings at test time to make a reliable judgement about entailment even if no exact match in the entailment graph is found.

The approach presented in section 7 can serve as a blueprint for the application of entailment graphs in other languages, e.g. by using the Chinese entailment graph in addition to the English and German one. There is still much to discover with regards to how different amounts of different language data influence the performance of the model. Another avenue for exploration is the zero-shot cross-lingual transfer. The best English model performs about 12 percent points worse in F1 score in German than in English, but we did not examine how well it performs in other languages that have high quality representations within XLM-RoBERTa, e.g. Spanish or French. These comparisons can be especially insightful for the questions of how predicates are represented within the model: While sentence based entailment data sets like XNLI don't annotate the part of the sentence which triggers the entailment, the data sets used here allow to specifically examine the distinct linguistic properties of predicates.

This, finally, leads us to the question of future work for entailment graphs in general. Entailment graphs allow for multiple kinds of uses: We can look up two words in them and check if they have a high entailment score. This is what we do at test time, when we evaluate the quality of entailment graphs on the Levy-Holt data set. We can use them to generate training data, as shown in chapter 7. Entailment graphs can also be used to enhance link prediction (Hosseini et al., 2019), or to enhance queries in a question answering scenario. Entailment graphs have been extended to include temporal information (Guillou et al., 2021) and information about unary relations (McKenna et al., 2021). One possible application for multilingual entailment graphs is translation quality estimation, where the aligned relations of a multilingual entailment graph could be used to check whether a machine translation is preserving the meaning of the predicate. Entailment graphs are an explainable and unsupervised way to represent predicate entailment, that eschews the problem of needing large amounts of human annotated data. It is up to future researchers to create data sets and tasks to show the entailment graphs' true strengths.

Chapter 9

Ethical considerations

The absence of human subject research in this thesis could easily lead to the assumption that no further examination of its ethical implications is necessary. Nevertheless, texts and textual artifacts like news articles and entailment graphs don't exist outside of their social context. Throughout their life cycles they intersect with different groups of people in different ways. Examining potential harms that could arise from the methods and results of NLP research has come into focus within the last years e.g. in Bender et al. (2021); Paullada et al. (2021); Hovy and Prabhumoye (2021) and many others. We examine the ethical implications of our work at two points: First, is harm done to build the described systems and to run the described experiments? And second, is harm likely to arise from the presented findings, e.g. if a government agency or a company was to use the findings of this thesis?

Ethics at Construction: At the very beginning of our pipeline stands the input data. We describe the exact sources of the news articles in more detail in chapter 3.2. While all of the news articles were publicly available, none of the authors explicitly agreed to their text being used to train machine learning models in general or our models specifically. Crawford (2021) criticises this approach of data mining. In her book she likens this practise to a seizing of the commons and points out that it obscures a connection to human subject research and therefore shields NLP research from the strict ethical oversight that is in place for medical and psychological research. We acknowledge this criticism, but we want to point out that unlike commercial actors who practice data mining from publicly available sources, the results of our usage of the data are publicly available as well, in form of published articles and code. Even though we do not have the explicit consent of the authors of the texts we use, news articles are different from texts like tweets, emails or Facebook posts in that they are not intended as personal communication, but as neutral statements for a wide audience with author names often only presented as initials or aliases to underscore the impersonal nature of the text.

Building the systems discussed in this thesis does not only require large amounts of data, but also large computational resources. Recently many NLP conferences added the requirement of reporting the computational resources necessary to conduct the presented research. This does not only facilitate easier reproducibility of results, but also highlights the energy usage and hardware requirements of the presented work. For the German entity typing system (see chapter 4) we trained all models using a single GeForce RTX 2080 Ti GPU. Training each of the models took under an hour. All hyperparameters of the model were taken from the implementation of Chen et al. (2020a).

We trained the predicate entailment detection model of Schmitt and Schütze (2021a) (see chapter 7) using the same hardware. Depending on data size, the training of one model takes up to 90 minutes. Because the hyperparameters need to be tuned whenever we change the make-up of the data (e.g. mixing human annotated data with automatically generated data, or German data with English) each of the models we present is one of 50 models we trained on the same data with random hyperparameters, which increases the energy use accordingly.

The construction of entailment graphs is an unsupervised learning problem, that can be divided into relation extraction and graph building. Extracting subject-predicate-object triples from 10 K sentences of German text using the hardware described above takes approximately 9 hours, making it 387 hours to extract all the triples we use to for the German entailment graph. Constructing the entailment graph from the triples using 12 CPUs takes approximately 3 hours. But listing these numbers remains a box ticking exercise if we don't acknowledge their meaning. Even if these numbers are relatively small in comparison with the energy consumption of the training of large language models, this energy consumption nevertheless contributes to climate change: The electricity used by the servers is generated mostly by burning gas (UoE, 2022), not to mention the emissions generated by the production and upkeep of the servers. Because this work concentrates on entailment detection in English and German, communities that are most likely to be impacted by climate change are not very likely to reap any benefits from the work presented in this thesis.

Ethics at Deployment: The code that transforms our input data into entailment graphs contains many components. Some of them were created by us (e.g. the German entity typing system described in chapter 4) but most of them are state of the art models used without modification to perform a defined task, e.g. dependency parsing, named entity recognition or entailment graph construction. While it is worth looking at the data these model were trained on and on the energetic cost it took to train them, we will leave this to further research like Strubell et al. (2019). Other parts of the model are more unique to our approach, which is what we will focus on here.

To create the German entity typing models we draw from several publicly available resources. We generate German named entity typing data by using the parallel corpus WikiMatrix (Schwenk et al., 2021). While Caswell et al. (2021) have pointed out weaknesses in the quality of this corpus, there has not been an analysis of harmful biases that might be contained in the corpus. The parallel sentences are taken from Wikipedia, which is written, moderated and edited by volunteers. The lack of diversity in the community of Wikipedia contributors is a well known problem that makes it likely that biases are encoded within Wikipedia (Ortega et al., 2008; Kessenides and Chafkin, 2016). To generate training data for the German general entity typing system we additionally use the German WordNet GermaNet (Hamp and Feldweg, 1997). Modelled after the English WordNet, it arranges words in synonym sets and links them to their hypernyms and hyponyms. To the best of out knowledge there is no analysis of biases within GermaNet. A manual search in GermaNet reveals existing errors. For example, the category of “sexual orientation” contains the hyponyms “bisexual” and “asexual”, but also “perverse”, “transsexual” and “intersex” which are not sexual orientations. The hyponyms of “gendered person” are “man”, “woman” and “hermaphrodite”, grouped with an intersex-phobic slur as its synonym. These incorrect and potentially harmful categorisations speak to the blind spots of the annotators.

This leads us to the question of how these biases in the data affect the finished system, and therefore potential users. Supervised learning models building on contextualised word embeddings like the one we use (Chen et al., 2020a) make it difficult to correlate the statistics of input data with undesired model output. It seems likely that biases found in Wikipedia and GermaNet are contained in the training data we generate and might show up in unexpected ways in our specific use case. Even the meaning of ’bias’ for different NLP tasks is a matter of ongoing discussion and change (Blodgett et al., 2020). Conducting a thorough analysis of the output of our German entity typing systems is unfortunately out of the scope of this thesis and has to be left to future work.

Unlike black box systems like the model of Chen et al. (2020a) the entailment graph building system of Hosseini et al. (2018) allows us to trace back the entailment judgements that our model makes to co-ocurrence statistics of the relation triples that we extract from the input text. Entailment graphs are human readable and this way easier to survey for possible biases. Incorrect entailments are especially common when a small data set is used to create the entailment graphs. In the course of our work we iterated through many different versions of German entailment graphs, which helped us to refine our approach and create the German entailment graph we presented in the previous chapters. When we created German entailment graphs from a small set of German news articles we found that common story lines of the articles reappeared as entailments. For example the person#location entailment graph stated that a person dying in a place entailed the person demonstrating at that place, and a person trying to reach a place entailed the person drowning. This mirrors the German news reporting on the so called Arab spring of the early 2010s and news of refugees dying in the Mediterranean sea in an attempt to reach to Europe. Entailments like these are not as common in entailment graphs constructed from more data. In the German entailment graph used in chapter 7, the predicate dying in the person#location graph has high entailment scores with paraphrases like "losing ones life" and preconditions like "being injured in" and "being treated in". This is only anecdotal evidence, and a thorough examination of the entailment graphs is an avenue for future work.

Is predicate entailment reasoning? The comet-like rise of large language models has raised the question how capable of human-like reasoning these models are. As Bender and Koller (2020) point out, language models can't acquire meaning if they are only trained on language surface form. This raises the question of how this work positions itself in this discourse. As we described earlier (see chapter 1) predicate entailments are grounded in extra-linguistic realities, yet in our method we only learn the entailments from surface form as expressed in news articles. We therefore want to make clear that our work does not claim that our model "understands" or "reasons" in a way that is similar to humans. It learns from associations in data and is therefore likely to replicate biases that are present in that data.

Bibliography

Agerri, R., Chung, Y., Aldabe, I., Aranberri, N., Labaka, G., and Rigau, G. (2018). Building named entity recognition taggers via parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Artetxe, M. and Schwenk, H. (2019). Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Barrault, L., Bojar, O., Costa-Jussa, M. R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., et al. (2019). Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Bender, E. M. and Koller, A. (2020). Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198.

Berant, J., Dagan, I., and Goldberger, J. (2011). Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 610–619. Association for Computational Linguistics.

Bergmann, T., Schmitz, M., Fader, T., and Balasubramanian, N. (2016). From text to facts: Relation extraction on german websites. `https:https://github.com/tabergma/relation-extraction`.

Blodgett, S. L., Barocas, S., Daumé III, H., and Wallach, H. (2020). Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Caswell, I., Kreutzer, J., Wang, L., Wahab, A., van Esch, D., Ulzii-Orshikh, N., Tapo, A., Subramani, N., Sokolov, A., Sikasote, C., et al. (2021). Quality at a glance: An audit of web-crawled multilingual datasets.

Chen, T., Chen, Y., and Van Durme, B. (2020a). Hierarchical entity typing via multilevel learning to rank. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8465–8475.

Chen, T., Jiang, Z. P., Poliak, A., Sakaguchi, K., and Van Durme, B. (2020b). Uncertain natural language inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8772–8779.

Choi, E., Levy, O., Choi, Y., and Zettlemoyer, L. (2018). Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *ACL*.

Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069.

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S. R., Schwenk, H., and Stoyanov, V. (2018). Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Crawford, K. (2021). *The Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence*. Yale University Press.

Cui, L., Wei, F., and Zhou, M. (2018). Neural open information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 407–413.

Dai, H., Du, D., Li, X., and Song, Y. (2019). Improving fine-grained entity typing with entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6211–6216.

Daiber, J., Jakob, M., Hokamp, C., and Mendes, P. N. (2013). Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019a). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019b). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Ehrmann, M., Turchi, M., and Steinberger, R. (2011). Building a multilingual named entity-annotated corpus using annotation projection. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 118–124.

Evang, K., Abzianidze, L., and Bos, J. (2019). Ccgweb: a new annotation tool and a first quadrilingual ccg treebank. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 37–42.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*, Edinburgh, Scotland, UK.

Falke, T., Stanovsky, G., Gurevych, I., and Dagan, I. (2016). Porting an open information extraction system from english to german. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 892–898.

Fellbaum, C. (2010). Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

Freitag, M., Rei, R., Mathur, N., kiu Lo, C., Stewart, C., Foster, G., Lavie, A., and Bojar, O. (2021). Results of the wmt21 metrics shared task: Evaluating metrics with expert-based human evaluations on ted and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, Online.

Gabrilovich, E., Ringgaard, M., and Subramanya, A. (2013). Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0). *Note: http://lemurproject. org/clueweb09/FACC1/Cited by*, 5:140.

Geffet, M. and Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114.

Guillou, L., De Vroe, S. B., Hosseini, M. J., Johnson, M., and Steedman, M. (2021). Incorporating temporal information in entailment graph mining.

Hamp, B. and Feldweg, H. (1997). Germanet-a lexical-semantic net for german. *Automatic information extraction and building of lexical semantic resources for NLP applications*.

Han, X., Gao, T., Lin, Y., Peng, H., Yang, Y., Xiao, C., Liu, Z., Li, P., Zhou, J., and Sun, M. (2020). More data, more relations, more context and more openness: A review and outlook for relation extraction. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 745–758.

Harris, Z. (1954). Distributional structure. *Word*, 10:146–162.

Harting, T., Mesbah, S., and Lofi, C. (2020). Lorem: Language-consistent open relation extraction from unstructured text. In *Proceedings of The Web Conference 2020*, pages 1830–1838.

Havasi, C., Speer, R., and Alonso, J. (2007). Conceptnet 3: A flexible, multilingual semantic network for common sense knowledge. *Proceedings of Recent Advances in Natural Language Processing*.

Heinzerling, B. (2019). *Aspects of Coherence for Entity Analysis*. PhD thesis.

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.

Hosseini, M. J., Chambers, N., Reddy, S., Holt, X., Cohen, S., Johnson, M., and Steedman, M. (2018). Learning typed entailment graphs with global soft constraints. *Transactions of the Association for Computational Linguistics*.

Hosseini, M. J., Cohen, S. B., Johnson, M., and Steedman, M. (2019). Duality of link prediction and entailment graph induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4736–4746.

Hovy, D. and Prabhumoye, S. (2021). Five sources of bias in natural language processing. *Language and Linguistics Compass*, 15(8):e12432.

Hsu, T.-Y., Liu, C.-L., and Lee, H.-y. (2019). Zero-shot reading comprehension by cross-lingual transfer learning with multi-lingual language representation model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5933–5940, Hong Kong, China. Association for Computational Linguistics.

Kementchedjhieva, Y., Bjerva, J., and Augenstein, I. (2018). Copenhagen at conll–sigmorphon 2018: Multilingual inflection in context with explicit morphosyntactic decoding.

Kessenides, D. and Chafkin, M. (2016). Is wikipedia woke?

Kolluru, K., Adlakha, V., and Aggarwal, S. (2020). Mausam, and soumen chakrabarti. openie6: Iterative grid labeling and coordination analysis for open information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3748–3761.

Krenn, B. (2000). The usual suspects: Data-oriented models for identification and representation of lexical collocations.

Kuang, J., Cao, Y., Zheng, J., He, X., Gao, M., and Zhou, A. (2020). Improving neural relation extraction with implicit mutual relations. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1021–1032. IEEE.

Lample, G., Conneau, A., Ranzato, M., Denoyer, L., and Jégou, H. (2018). Word translation without parallel data. In *International Conference on Learning Representations*.

Lauscher, A., Ravishankar, V., Vulić, I., and Glavaš, G. (2020). From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499.

Lechelle, W., Gotti, F., and Langlais, P. (2019). Wire57: A fine-grained benchmark for open information extraction. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 6–15.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Leitner, E., Rehm, G., and Schneider, J. M. (2020). A dataset of german legal documents for named entity recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4478–4485.

Lenat, D., Prakash, M., and Shepherd, M. (1986). Cyc: Using common sense knowledge to overcome brittleness and knowledge acquistion bottlenecks. *AI Mag.*, 6(4):65–85.

Levy, O. and Dagan, I. (2016a). Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 249–255.

Levy, O. and Dagan, I. (2016b). Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 249–255, Berlin, Germany. Association for Computational Linguistics.

Lewis, M. and Steedman, M. (2013a). Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Lewis, M. and Steedman, M. (2013b). Unsupervised induction of cross-lingual semantic relations. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 681–692.

Li, B., He, Y., and Xu, W. (2021). Cross-lingual named entity recognition using parallel corpus: A new approach using xlm-roberta alignment. *arXiv preprint arXiv:2101.11112*.

Li, T., Weber, S., Hosseini, M. J., Guillou, L., and Steedman, M. (2022). Cross-lingual inference with a chinese entailment graph. In *Proceedings of the Society for Computation in Linguistics*.

Ling, X. and Weld, D. S. (2012). Fine-grained entity recognition. In *AAAI*, volume 12, pages 94–100.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

López, F., Heinzerling, B., and Strube, M. (2019). Fine-grained entity typing in hyperbolic space. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 169–180, Florence, Italy. Association for Computational Linguistics.

Mathur, N., Wei, J., Freitag, M., Ma, Q., and Bojar, O. (2020). Results of the wmt20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725.

McCoy, R. T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference.

McKenna, N., Guillou, L., Hosseini, M. J., de Vroe, S. B., Johnson, M., and Steedman, M. (2021). Multivalent entailment graphs for question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10758–10768.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Moussallem, D., Usbeck, R., Röder, M., and Ngonga Ngomo, A.-C. (2017). MAG: A Multilingual, Knowledge-base Agnostic and Deterministic Entity Linking Approach. In *K-CAP 2017: Knowledge Capture Conference*, page 8. ACM.

Nguyen, D. B., Hoffart, J., Theobald, M., and Weikum, G. (2014). Aida-light: High-throughput named-entity disambiguation. *LDOW*, 1184.

Ni, J., Dinu, G., and Florian, R. (2017). Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1470–1480.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Ortega, F., Gonzalez-Barahona, J. M., and Robles, G. (2008). On the inequality of contributions to wikipedia. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pages 304–304. IEEE.

Paullada, A., Raji, I. D., Bender, E. M., Denton, E., and Hanna, A. (2021). Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11):100336.

Pavlick, E. and Kwiatkowski, T. (2019). Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Perez, E., Kiela, D., and Cho, K. (2021). True few-shot learning with language models. *Advances in Neural Information Processing Systems*, 34.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of* *the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners.

Reddy, S., Lapata, M., and Steedman, M. (2014). Large-scale semantic parsing without question-answer pairs. *Transactions of the Association of Computational Linguistics*, 2(1):377–392.

Ricketts Holt, X. et al. (2018). Probabilistic models of relational implication.

Ro, Y., Lee, Y., and Kang, P. (2020). Multiˆ2oie: Multilingual open information extraction based on multi-head attention with bert. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1107–1117.

Rogers, A., Drozd, A., and Li, B. (2017). The (too many) problems of analogical reasoning with word vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 135–148.

Ruppenhofer, J., Rehbein, I., and Flinz, C. (2020). Fine-grained named entity annotations for german biographic interviews.

Schick, T. and Schütze, H. (2021a). Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

Schick, T. and Schütze, H. (2021b). True few-shot learning with prompts–a real-world perspective. *arXiv preprint arXiv:2111.13440*.

Schmitt, M. (2021). Language models for lexical inference in context. `https://github.com/mnschmit/lm-lexical-inference`.

Schmitt, M. and Schütze, H. (2019). Sherliic: A typed event-focused lexical inference benchmark for evaluating natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 902–914.

Schmitt, M. and Schütze, H. (2021a). Language models for lexical inference in context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1267–1280.

Schmitt, M. and Schütze, H. (2021b). Continuous entailment patterns for lexical inference in context. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6952–6959.

Schneider, R., Oberhauser, T., Klatt, T., Gers, F. A., and Löser, A. (2017). Analysing errors of open information extraction systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 11–18.

Schwenk, H., Chaudhary, V., Sun, S., Gong, H., and Guzmán, F. (2021). Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361.

Soares, L. B., Fitzgerald, N., Ling, J., and Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

Stanovsky, G. and Dagan, I. (2016). Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305.

Stanovsky, G., Ficler, J., Dagan, I., and Goldberg, Y. (2016). Getting more out of syntax with props. *arXiv preprint arXiv:1603.01648*.

Strötgen, J. and Gertz, M. (2010). Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650.

Szpektor, I. and Dagan, I. (2008). Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 849–856.

UoE (2022). Low carbon and renewable energy.

Vulić, I., Ruder, S., and Søgaard, A. (2020). Are all good word vector spaces isomorphic? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3178–3192.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Weber, S. and Steedman, M. (2019). Construction and alignment of multilingual entailment graphs for semantic inference. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 77–79.

Weber, S. and Steedman, M. (2021a). Fine-grained general entity typing in german using germanet. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 138–143.

Weber, S. and Steedman, M. (2021b). Zero-shot cross-lingual transfer is a hard baseline to beat in German fine-grained entity typing. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 42–48, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Webson, A. and Pavlick, E. (2021). Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*.

Williams, A., Nangia, N., and Bowman, S. (2018a). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Williams, A., Nangia, N., and Bowman, S. (2018b). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Wu, Y., Hu, Y., Li, T., Feng, Y., Weber, S., Steedman, M., and Zhao, D. (2021). Multilingual entailment graph alignment augmented by cross-graph guided interaction. In *Under Submission*, pages 688–725.

Yarowsky, D. and Ngai, G. (2001). Inducing multilingual pos taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, pages XXX–XXX. ACL.

Zalando (2020). *ZAP - Multilingual Annotation Projection Framework*.

Zhan, J. and Zhao, H. (2020). Span model for open information extraction on accurate corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9523–9530.

Zhang, C. and Weld, D. S. (2013). Harvesting parallel news streams to generate paraphrases of event relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1776–1786.

Zhao, M., Zhu, Y., Shareghi, E., Vulić, I., Reichart, R., Korhonen, A., and Schütze, H. (2021). A closer look at few-shot crosslingual transfer: The choice of shots matters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5751–5767.