Fine-grained General Entity Typing in German using GermaNet

Sabine Weber
University of Edinburgh
s.weber@sms.ed.ac.uk

Mark Steedman
University of Edinburgh
steedman@inf.ed.ac.uk

Abstract

Fine-grained entity typing is important to tasks like relation extraction and knowledge base construction. We find however, that fine-grained entity typing systems perform poorly on general entities (e.g. "ex-president") as compared to named entities (e.g. "Barack Obama"). This is due to a lack of general entities in existing training data sets. We show that this problem can be mitigated by automatically generating training data from WordNets. We use a German WordNet equivalent, GermaNet, to automatically generate training data for German general entity typing. We use this data to supplement named entity data to train a neural fine-grained entity typing system. This leads to a 10% improvement in accuracy of the prediction of level 1 FIGER types for German general entities, while decreasing named entity type prediction accuracy by only 1%.

1 Introduction

The task of fine-grained entity typing is to assign a semantic label (e.g. '/person/politician' or '/location/city') to an entity in a natural language sentence. In contrast to coarse grained entity typing it uses a larger set of types (e.g. 112 types in the FIGER ontology (Ling and Weld, 2012)), and a multilevel type hierarchy. An example of fine grained entity typing can be seen in Figure 1. Fine-grained entity typing is an important initial step in context sensitive tasks such as relation extraction (Kuang et al., 2020), question answering(Yavuz et al., 2016) and knowledge base construction (Hosseini et al., 2019).

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 are 17% lower on general entities than on named entities (see Table 1 and section 5). This is because the training data for these systems contains only named entities, but not general entities (e.g. Weber and Steedman (2021, under submission); Ling and Weld (2012)). This is the problem we address with our approach.

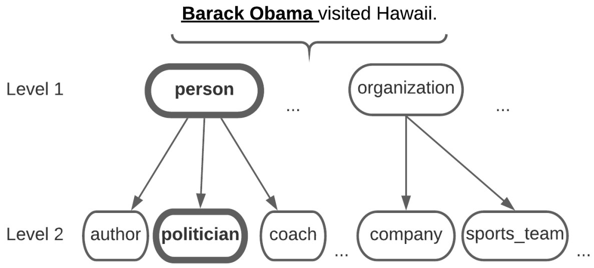

Figure 1: Fine-grained entity typing with the FIGER ontology in English. Correct types are highlighted.

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). 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. (2020), 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% improvement in accuracy of the prediction of level 1 FIGER types for German general entities, while decreasing named entity prediction accuracy by only 1%. 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.

2 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., 2020; Dai et al., 2019; López et al., 2019). By choosing to use the model of (Chen et al., 2020), 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 (2021, under submission) 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.

3 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.

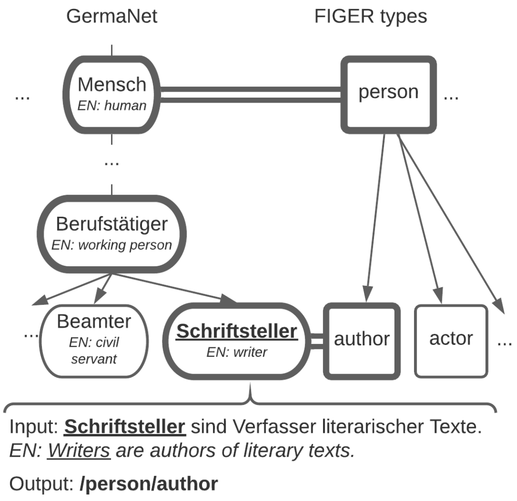

Figure 2: An example of FIGER type assignment using GermaNet. The manual mapping between GermaNet and FIGER is indicated by double lines. Whenever a word in the hypernym path of the input word is mapped to a FIGER type, the respective type gets assigned.

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 2 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. (2020). 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. (2020).

4 Experimental setup

4.1 Data sets

As a NE training set we use the German fine-grained entity typing corpus of Weber and Steedman (2021, under submission). This data set was generated from the WikiMatrix corpus by Schwenk et al. (2019) 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.

4.2 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 in Section 3 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 in Section 6. The resulting corpus consists of 200K sentences of German FIGER typed GE data [1].

4.3 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 (2021, under submission).

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 (2021, under submission) and train the hierarchical typing model on it. We shuffle the sentence order before training.

Baseline: We compare these two neural approaches against using only GermaNet. In this baseline we use the approach described in Section 3 and Figure 2 to type our test data.

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 (2021, under submission) 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.

5 Results

Table 1 shows the accuracy and F1 scores on the gold German test set. Additionally, development set results are presented in appendix A. We compare the performance of models trained with different amounts of GE data on the GE and NE test sets described in section 4.4.

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%. 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.

[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 1: 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% when 20K sentences of GE training data are added, while NE accuracy and F1 declines by only 1%.

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

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 contextualized word embeddings (XLM-RoBERTa) to encode entities and their context during training. While our GE data provides it with high confidence non-polysemous examples, it 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 by also GE performance. This is due to noise in the GE typing data. A manual error analysis of 100 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.

7 Conclusion and future work

In this paper we have shown that it is possible to improve the performance of a German fine-grained entity typing system using GermaNet. We create silver annotated general entity typing data for training a fine-grained entity typing model that builds upon contextualised word embeddings (in our case, XLM-RoBERTa). Our results 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-precison 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.

References

Tongfei Chen, Yunmo Chen, and Benjamin Van Durme. 2020. Hierarchical entity typing via multi-level 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.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, E. Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

Hongliang Dai, Donghong Du, Xin Li, and Yangqiu Song. 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.

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

Birgit Hamp and Helmut Feldweg. 1997. Germanet-a lexical-semantic net for german. In *Automatic information extraction and building of lexical semantic resources for NLP applications*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Mohammad Javad Hosseini, Shay B Cohen, Mark Johnson, and Mark Steedman. 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.

Jun Kuang, Yixin Cao, Jianbing Zheng, Xiangnan He, Ming Gao, and Aoying Zhou. 2020. Improving neural relation extraction with implicit mutual relations. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1021–1032. IEEE.

Elena Leitner, Georg Rehm, and Julián Moreno-Schneider. 2020. A dataset of german legal documents for named entity recognition. *arXiv preprint arXiv:2003.13016*.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *AAAI*, volume 12, pages 94–100.

Federico López, Benjamin Heinzerling, and Michael Strube. 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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Josef Ruppenhofer, Ines Rehbein, and Carolina Flinz. 2020. Fine-grained named entity annotations for german biographic interviews.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620

	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 2: 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% for Level 2 metrics.

language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

Sabine Weber and Mark Steedman. 2021, under submission. Fine-grained named entity typing beyond english using annotation projection.

Semih Yavuz, Izzeddin Gur, Yu Su, Mudhakar Srivatsa, and Xifeng Yan. 2016. Improving semantic parsing via answer type inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 149–159.

A Development set results

Development set results can be seen in Table 2. We use the development set to determine which amount of of added GEs achieves the best result. The exact amount of GEs necessary for an ideal result might vary depending on the fine-grained entity typing model and the NE data used. The development set enables the user to determine this amount for their individual application. Best development set performance aligns with best test set performance on Level 1 metrics, and is only off by 1% for Level 2 metrics.

B Reproducibility

In keeping with the NAACL reproducibility guildines we report the following implementation details of our model: We trained all models using a single GeForce RTX 2080 Ti GPU. Training each of the models took under an hour. The number of model parameters is 50484362. All hyperparameters of the model were taken from the implementation of Chen et al. (2020). All additional code used and all of our data sets are available on github.com/webersab/german_general_entity_typing.