Automatic induction of named entity classes from legal text corpora*

Peter Bourgonje[2], Anna Breit[1][0000−0001−6553−4175], Maria Khvalchik[1], Victor Mireles[1][0000−0003−3264−3687], Julian Moreno-Schneider[2][0000−0003−1418−9935], Artem Revenko[1][0000−0001−6681−3328], and Georg Rehm[2][0000−0002−7800−1893]

[1] Semantic Web Company, Austria {first.lastname}@semantic-web.com
[2] DFKI GmbH, Germany,
[peter.bourgonje,julian.moreno_schneider,georg.rehm]@dfki.de

Abstract. Named Entity Recognition tools and datasets are widely used. The standard pre-trained models, however often do not cover specific application needs as these models are too generic. We introduce a methodology to automatically induce fine-grained classes of named entities for the legal domain. Specifically, given a corpus which has been annotated with instances of coarse entity classes, we show how to induce fine-grained, domain specific (sub-)classes. The method relies on predictions of the masked tokens generated by a pre-trained language model. These predictions are then collected and clustered. The clusters are then taken as the new candidate classes. We develop an implementation of the introduced method and experiment with a large legal corpus in German language that is manually annotated with almost 54,000 named entities.

Keywords: named entity recognition · ontology induction · knowledge discovery · deep learning · language model

1 Introduction and Problem Statement

The amount of available digital information, or that is currently being digitized, does not stop growing, and with it the mechanisms to carry out semantic processing on it [5]. In this sense, the recognition of named entities (NER) is one of the first steps to be carried out in semantic processing. NER systems recognize entities and classify them into different types (classes). Normally, these classes are very coarse-grained and only use abstract types like "Person", "Organization" and "Location". In specific domains, such as biomedical or legal, this broad classification limits the richness of the results, and more fine-grained classes are more appropriate which could be used to create a specific class hierarchy.

Unfortunately, creating domain-specific classes is not easy and requires vast input from domain experts. The main problems that arise are: (1) overpopulated

* The work presented in this paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 780602 (Lynx) and from the German Federal Ministry of Education and Research (BMBF) through the project QURATOR (Wachstumskern no. 03WKDA1A).

classes (as compared to the other classes), which should have been divided into more classes; and (2) underpopulated classes (as compared to the other classes), which should be reclassified within other classes. Automating this work, in part or whole, can be a drastic improvement when adapting NLP systems.

Having domain specific fine-grained (sub-)classes is not only useful to classify the different entity types that appear in the documents, but having this information can improve, for example, faceted search applications. Other applications include document clustering, as well as improvements in any NLP task that requires annotated entities as input, such as Relation Extraction, Event Detection, Fact Checking or Entity Linking.

Given a set of entities, there are many possible classifications of it, some of which might be more suitable than others for a particular application. One of such classification is implicit in the use of entities in a natural language corpus. Namely, when two entities are often found in similar contexts within the corpus, it is because the speakers behind the corpus, in a sense, ascribe to them a class in common. One way to capture the similarity of contexts between two entities e_1 and e_2 , is to train a predictive language model on the corpus, and then quantify how often the context in which e_1 is mentioned, is deemed by the model as a context for e_2. The precise definition of *context*, and the nature of the language model influence the resulting similarity, and this notion of similarity can be extended into an approximate partition of all entities into sub-classes by using decomposition methods. Finally, if an initial classification of entities is known, refinements can be obtained by the above described process on the entities belonging to one of the initial classes. The resulting class hierarchy attempts to capture the nuance in the usage patterns of entities in natural language.

The remainder of the paper is organized as follows. In Section 2 we give an overview of related work. In Section 3 we described the methodology and evaluate it in Section 4, followed by a discussion of the results in Section 5. We conclude and outline next steps in Section 6.

1.1 Lynx Project

The Lynx project[3] [16] focuses on the creation of a legal domain knowledge graph (Legal Knowledge Graph – LKG) and its use for the semantic analysis of documents in the legal domain. The three different use cases of the project operate in different languages and focus on different tasks:

- analysis of GeoThermal permits and best practices in Dutch,
- analysis of contracts and court decisions in German,
- question answering on top labor law in Spanish.

In order to analyze this multilingual legal textual data Lynx project develops various services like named entity recognition (NER), entity linking (EL), question answering (QA), etc. that can operate in different languages. The basis for many services are the annotations services that can identify and link or type

[3] `http://www.lynx-project.eu`

entities – NER and EL. Whereas EL relies on the various multilingual terminologies available in application domains, the NER service requires domain-specific training data or a pre-trained model. One such dataset reported in [11, 12] is a subject of this paper. However, for many potential domains fine-grained NER training datasets are not available and, therefore, the outcomes of this research would enable the domain and language adaptation for NER tools.

2 Related Work

As explained in Section 1, many NER systems are trained on data that distinguishes a small number of entity types [21, 22, 18]. Two prominent exceptions to this are the FIGER [13] and OntoNotes [9] data sets, which feature a larger collection of more fine-grained entity types. These data sets are used for training and evaluation by Shimaoka et al. [20] and Murty et al. [17], but in both papers the collection of entity types is directly taken from the training data and not extended upon. Our approach is more similar to Del Corro et al. [7], who equally attempt to extract new classes from a small set of annotated entity types. In contrast to [7], however, we exploit more modern language modeling techniques (i. e., DistilBERT [19]), which we expect to pick up on features such as the verb-based extraction rules from Del Corro et al. [7] automatically.

Specifically focusing on entities in the legal domain, Angelidis et al. [2] work with a Greek corpus annotated for Named Entities and target six different entity types, aimed specifically at legal texts (i. e., including *legislation reference* and *public document reference*) and introduce a corpus annotated for four different types of geographical landmarks (*local district*, *area*, *road* and *point of interest*), though further details on the annotation procedure or the corpus itself are, to the best of our knowledge, not published. Another contribution specifically targeted at the legal domain is represented by Leitner et al. [11, 12]. We use this corpus and refer to Section 4 for more details.

In addition to focusing on and discovering new, finer-grained entity types, we attempt to induce an ontology-like structure for these emerging types in the process, resembling the task of ontology creation. As a starting point, a simple knowledge graph can be created that groups synonyms of entities under one unequivocal identifier (e. g., *United States of America* and *US*). A more sophisticated option is to link entities via properties of an ontology, e. g., using external knowledge graphs [15] or relation extraction [10]. This can be extended further by inferring the emerging class hierarchy to build an actual ontology, a common task of ontology learning.

Ontology learning is the process of deriving an ontology, i. e., a set of classes and relations between them, from natural language or structured data [3]. Common approaches are based on the discovery of linguistic patterns to detect domain-specific terms and relations. The classification of the identified terms into classes is often left to human experts, who do this based on pre-defined classification schemes. This process is sometimes supported by clustering algorithms, but to the best of our knowledge, the discovery of classes from scratch is

not reported in the literature, though Cimiano et al. [6] target a related task; the discovery of a class hierarchy from text data. In their work, they parse sentences to collect (verb, subject), (verb, object) and (verb, prepositional phrase) tuples. These tuples are later transformed into a formal context, the concept lattice of which is computed and pruned. The outcome of the procedure is a compacted partial order of classes. In our work we do not parse the sentences contained in a text. We focus on the top levels of the class hierarchy and, therefore, do not compute complete concept lattices[4].

3 Methods

The pre-trained NER tools that are available online like the NER specific models in OpenNLP[5] or in BERT[6] typically have a few models for recognizing some general purpose classes like "Person", "Location" and "Organization". We rely on the annotations by these pre-trained models and aim at inducing new finer-grained (sub-)classes.

To process annotations from an NER tool, we use the capability of language models to predict substitutes for named entities. A language model is a probability distribution over a sequence of words[7]. Therefore, language models can be used to predict the most suitable *substitutes* for any word in a given sequence. Later on these substitutes will serve as class labels for newly induced classes. In our experiments we used the pre-trained BERT language model [8] as implemented in [23], and obtain the top $2 * m$ words that could potentially substitute each occurrence of a named entity, see (1) in Figure 1. These are then lemmatized in order to avoid counting duplicate words, for example to avoid counting separately single and plural forms of a word.

The obtained substitutes are then collected into a binary matrix with contexts as rows and substitutes as columns, see (2) in Figure 1. Let N_{ctx_i} be the total number of contexts for named entity NE_i and N_{s_i} be the total number of substitutes. We can represent the contexts and their substitutes as a binary matrix $\mathcal{I} \in \{0,1\}^{N_{ctx_i} \times N_{s_i}}$, where a 1 (or `True`) in entry $\mathcal{I}_{a,b}$ means that the named entity NE_i in the a-th context could be substituted by b-th substitute as predicted by our language model, see (2) in Figure 1. For named entities that have at least five contexts, we perform a matrix factorisation on top of this binary matrix, in order to find the most representative clusters of substitutes. The used clustering procedure resembles approaches for word sense induction, (e. g., [1]), more specifically, we adapted algorithm 2 for matrix factorisation from [4]. These representative clusters can be interpreted as *senses* (sns_{ij}) of the corresponding named entity NE_i while the set of substitutes serve as sense *descriptors* $[descr]_{ij}$ (see output of (2) in Fig. 1).

[4] The size of the resulting formal contexts is rather large and the computation of the concept lattice would need significant computational resources.

[5] `https://opennlp.apache.org` accessed 02 May 2020

[6] `https://github.com/google-research/bert` accessed 02 May 2020

[7] `https://en.wikipedia.org/wiki/Language_model` accessed 06 May 2020

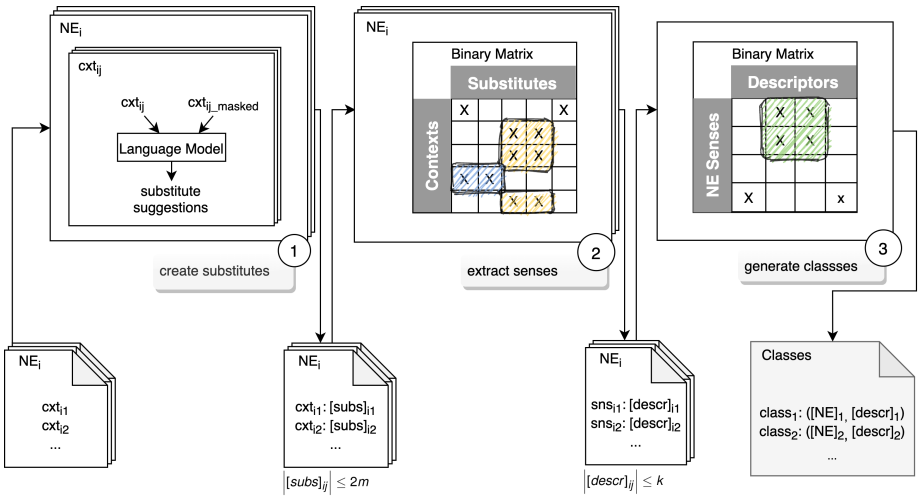

Fig. 1. Class induction diagram

Finally, we represent all senses of all named entities along with their descriptors in another binary matrix and create representative clusters in the same manner as for the previous matrix. We take only those clusters that have at least $th_C = 6$ descriptors. The resulting clusters correspond to the predicted candidate classes $class_i$, which each consist of a set of descriptors $[descr]_i$ and a set of named entities $[NE]_i$ (see (3) in Figure 1). We call $[NE]_i$ the *core entities* of the respective class.

Note that the maximum possible value of N_s is $N_{NE} * k$. However, in the results we expect and observe smaller values of N_s, as representative substitutes for different named entities actually overlap; and the smaller value of N_s indicates that we could expect better results of the whole procedure as many different named entities share substitutes and could be efficiently grouped.

4 Evaluation

We consider a legal dataset with manually annotated NEs [11, 12], which contains almost 54,000 manually annotated entities, mapped to 19 fine-grained semantic subclasses (NER types), belonging to four coarse classes: PER, LOC, DOC, ORG. Neither of these types are used during training. For evaluation, we use three criteria to compare candidate classes against, which are explained below.

C1: The first criterion implies comparing to a classification system corresponding to the original fine-grained classes used to annotate the dataset. Thus, the more each of our candidate classes is fully contained in one of the original NER types, the more consistent it is according to this criteria. We say that an NER type d is the best match for a candidate class c if d is the NER type that has more

entities in common with c. Because we are not evaluating individual *instances* of Named Entities (typically evaluated using precision, recall and F_1-score), but the (distribution of) entity types as groups, to quantify consistency, we compute the log-odds ratio of the fraction of the entities of c that belong to d, over the fraction of the entities in the whole corpus that belong to d.

C2: For each candidate subclass, we attempt to relate it to some Wikidata category. For this, we use Entity Fishing [14] to find German language Wikidata entities that have a surface form (label) similar to the named entity. Then we investigate the Wikidata category that said entity belongs to, as well as all of its transitive broader categories, all of which are deemed comparable classes to the candidate classes. We compute the consistency of the candidate class, this time with respect to Wikidata categories in the same fashion as above.

C3: Word embeddings have the possibility to capture the semantics of a given word in the notion of distance within a vector space. This means that for trained word embeddings, similar words tend to be closer together than unrelated words, meaning that the clusters of NE embeddings in the vector space can interpreted as NE classes. To obtain the corresponding NE word embeddings, we first resolve abbreviations in the entities. Herefore, we identify abbreviations by using a simple regex pattern[8], annotate them using the DBpedia spotlight API (similarity threshold set to 0.85), and replace the abbreviation with the surfaceform of the interlinked DBpedia resource. Then, we use the German DistilBERT model provided by huggingface[9] to receive the token embeddings. Finally, for each entity, we calculate the embedding vector by computing the mean of all token embeddings. In order to quantify the consistency of the clusters, we compute the mean cosine similarity between the entity embeddings of each candidate class and their embedding centroid.

For all three criteria, we evaluate the quality of the resulting consistencies by comparing them to what would be expected by a random partition of entities into candidate classes. To do this, we produce 100 such random partitions respecting the number and sizes of the candidate classes, compute criteria $C1$-$C3$ for each of their candidate classes, and compare their distribution with that of the candidate classes discovered by our method.

5 Results

In total, our method produced 21 candidate classes, see Table 1. In general, we were able to find candidate classes that resembled well the coarse *DOC* and *LOC* classes generated by the domain experts. However, comparing to the more fine-grained NER types, we see that the coverage of the best-matching ones to our candidate classes varies (Figure 3 (a)). 16/21 of the candidate classes intersect

[8] `[A-ZÄÖÜ][a-zöüä]*[A-ZÄÖÜ][a-zA-ZäöüÄÖÜß\s]*`

[9] `https://huggingface.co/transformers/pretrained_models.html`, 13 Aug. 2020

	Candidate Class Descriptors	Fine-grained NER Types
1	Ziel, Problem, Konzept, System, Gesetz, Recht	RS: 8, GS: 5
2	Teil, Band, Vers, Satz, Gesetz, Haupt	GS: 72, VO: 1
3	Aufgabe, Christus, Deutschland, Anwendung, Bedeutung, Recht, Punkten	VT: 4, GS: 7, RS: 1, VO: 1
4	Definition, Deutschland, Ordnung, Anwendung, Schutz, Recht	VT: 8, GS: 130, EUN: 11, RS: 15, VS: 2, VO: 4, LIT: 4
5	Praxis, Weber, Gesetz, Regel, Zusammenhang	RS: 26, LIT: 5
6	Russland, Schweden, Deutschland, Norwegen, Liechtenstein, Frankreich	RS: 19, EUN: 3, GS: 11, LD: 5, VS: 2, LIT: 4, ORG: 2
7	Schmidt, Neumann, Huber, Schulz, Weber, Muller, Schneider	LIT: 44, RR: 7, RS: 9, PER: 1
8	Sicherheit, Folge, Ausnahme, Hilfe, Erfolg, Wirkung	RS: 14, EUN: 1, GS: 1
9	Buch, Ober, Weber, Bauer, Muller, Fischer	LIT: 12
10	Verband, Verein, Bund, Deutschland, Deutsche, Deutschen	ORG: 24, INN: 4
11	Berlin, Deutschland, Bayern, Sachsen, Bonn, Karlsruhe, Hessen	LIT: 1, GS: 14, EUN: 1, RS: 20, VS: 1, GRT: 1, VO: 1
12	Verfassung, Koenig, GmbH, Senat, Polizei, Gesetz	RS: 18, GS: 3, VS: 1
13	Bericht, Grundlage, Entwurf, Revision, Auflage, Gesetz	RS: 14, GS: 1, LIT: 1
14	Beispiel, Bild, Artikel, Quellen, Abschnitt, Tabelle, Angaben	VT: 1, LIT: 9, GS: 2, RS: 34
15	Geld, Bonus, Kosten, Deutschland, Leistung, Gesetz, Wert	RS: 11, GS: 4, VS: 1
16	Satz, Anlage, Grund, Gesetz, Form, Artikel	GS: 1492, VT: 72, VO: 32, LIT: 2, VS: 8, EUN: 10, RR: 1
17	Richter, Verfahren, Gericht, Urteil, Landgericht, Hamburg	GRT: 144, GS: 7, RS: 178, INN: 2
18	Stadt, Kreis, Bezirk, Gemeinde, Landkreis, Land	INN: 7, ST: 1, VS: 1, GRT: 3, GS: 2, VO: 1, ORG: 1
19	Lebens, Ortes, Ersten, Menschen, Gesetz, Patienten	VS: 1, GS: 7, EUN: 2, RS: 2, LIT: 1
20	Ausnahme, Schweiz, Auswahl, Stand, Hinweis, Gesetz	EUN: 11, LIT: 70, RS: 329, GS: 70, VO: 5, VT: 9, VS: 2
21	Holding, Unternehmen, Verwaltung, GmbH, Gesellschaft, Firma	UN: 37

Table 1. 21 classes produced by the conducted analysis.

with one of the original NER types in more than 50% of their entities, and one third do in what would be considered very unlikely by a random partition (more than 2 standard deviation away from the mean). For at least four of them we are confident that they represent refinements to the original NER types while the other could still be valid refinements for the coarse classes. In general, the distribution of log-odds ratios is shifted to the right with respect to that of the random classifications (Figure 2 (a)) indicating that our candidate classes are of good quality.

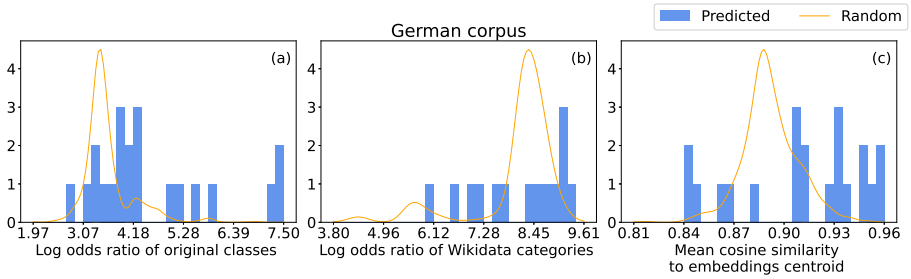

Fig. 2. Every candidate class was evaluated according three metrics: (a) how over-represented are the original NER types, (b) how over-represented are Wikidata categories and (c) how similar to the centroid of the point cloud in embedding space are its members. Shown as a line are also the distributions expected from a randomly generated categorization into candidate classes of the same sizes.

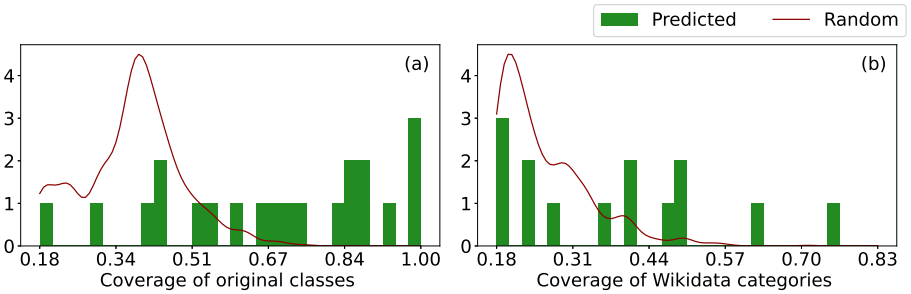

Fig. 3. For each candidate class, the fraction of its entities that correspond to the best matching original fine-grained NER type (a), and Wikidata category (b) is shown, along with the corresponding distribution for randomly generated partitions.

With respect to Wikidata categories, coverage is rather small (Figure 3 (b)), with only two candidate classes having more than half of their entities belonging

to a single Wikidata category. This is in part due to candidate classes related to People (PER) are hard to link to Wikidata, as the entities of the dataset rarely have a Wikipedia entry, and many of them have been anonymized. Furthermore, the categorization system of Wikidata leads to many entities being covered under very broad categories, leading to very similar log-odds ratios (Figure 2 (b)).

Analyzing the distribution in embedding space of the different candidate classes, we find a relatively high similarity of entities in word-embedding space, even for randomly generated classes (Figure 2 (c)). The contrast of this high similarity with the modest matching of the NER types, suggest that word embeddings trained on general domain corpora are unable to distinguish the details of the specialized legal corpus analyzed here. That being said, we do note that five of the candidate classes produced by our method are represented in embedding space by very compact point clouds, when compared against the backdrop of the randomly generated classes. However, only one of them is also corresponding strongly with one of the original fine-grained NER types, only corroborating our hypothesis that expert-derived classes are not properly captured by general-domain word embeddings, but suggesting that these also contain clusters of words with similar contexts.

6 Conclusions and Future Work

The experiments covered in this paper solely rely on inducing the candidate classes by a pre-trained BERT Language Model. Further fine tuning is necessary to overcome the limitations of this general domain Language Model (trained on Wikipedia articles) when applied to the legal domain. While not all obtained candidate classes match the expert-derived NER annotations, those which do can be considered refinements of the NER classes. In this experiment we ignored these NER annotations because we wanted to test the ability of this method to reproduce them. However, taking them into account in order to produce only refinements will no doubt lead to better results, for which we, unfortunately, lack any dataset to compare against.

The generality of the approach is another important point for us, so future work will be the application of this technique to another domain and other types of entities, such as geographic entities. Finally, the results of this work as well as the code to repeat the experiments is available for reuse and improvement by the community at `https://github.com/semantic-web-company/ptlm_wsid`.

In terms of future work, it will be interesting to investigate the case when several named entities share a label, for example, appearing as an organization and as a location in the same corpus. Already now we perform sense induction for named entities and, therefore, can potentially capture the different senses, however, a deeper investigation of such ambiguous named entities and strategies for best disambiguation are still to be developed.

References

1. Amrami, A., Goldberg, Y.: Word Sense Induction with Neural biLM and Symmetric Patterns. In: Proceedings of EMNLP 2018. pp. 4860–4867. Brussels (2018). https://doi.org/10.18653/v1/d18-1523
2. Angelidis, I., Chalkidis, I., Koubarakis, M.: Named entity recognition, linking and generation for greek legislation. In: JURIX (2018)
3. Asim, M.N., Wasim, M., Khan, M.U.G., Mahmood, W., Abbasi, H.M.: A survey of ontology learning techniques and applications. Database **2018** (2018)
4. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. Journal of Computer and System Sciences **76**(1), 3 – 20 (2010). https://doi.org/10.1016/j.jcss.2009.05.002
5. Bourgonje, P., Moreno-Schneider, J., Nehring, J., Rehm, G., Sasaki, F., Srivastava, A.: Towards a Platform for Curation Technologies: Enriching Text Collections with a Semantic-Web Layer. In: Sack, H., Rizzo, G., Steinmetz, N., Mladenia, D., Auer, S., Lange, C. (eds.) The Semantic Web. pp. 65–68. No. 9989 in Lecture Notes in Computer Science, Springer (2016)
6. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. J. Artif. Intell. Res. **24**, 305–339 (2005)
7. Del Corro, L., Abujabal, A., Gemulla, R., Weikum, G.: FINET: Context-aware fine-grained named entity typing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 868–878. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015). https://doi.org/10.18653/v1/D15-1103, `https://www.aclweb.org/anthology/D15-1103`
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
9. Gillick, D., Lazic, N., Ganchev, K., Kirchner, J., Huynh, D.: Context-dependent fine-grained entity type tagging. CoRR **abs/1412.1820** (2014), `http://arxiv.org/abs/1412.1820`
10. Giorgi, J., Wang, X., Sahar, N., Shin, W.Y., Bader, G.D., Wang, B.: End-to-end named entity recognition and relation extraction using pre-trained language models. arXiv preprint arXiv:1912.13415 (2019)
11. Leitner, E., Rehm, G., Moreno-Schneider, J.: Fine-grained Named Entity Recognition in Legal Documents. In: Acosta, M., Cudré-Mauroux, P., Maleshkova, M., Pellegrini, T., Sack, H., Sure-Vetter, Y. (eds.) Proceedings of SEMANTiCS 2019. pp. 272–287. No. 11702 in Lecture Notes in Computer Science, Springer, Karlsruhe, Germany (9 2019)
12. Leitner, E., Rehm, G., Moreno-Schneider, J.: A Dataset of German Legal Documents for Named Entity Recognition. In: Calzolari, N., Béchet, F., Blache, P., Cieri, C., Choukri, K., Declerck, T., Isahara, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020). pp. 4480–4487. European Language Resources Association (ELRA), Marseille, France (2020)
13. Ling, X., Weld, D.S.: Fine-grained entity recognition. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence. p. 94–100. AAAI'12, AAAI Press (2012)
14. Lopez, P.: entity-fishing. `https://github.com/kermitt2/entity-fishing` (2016–2020)
15. Machado, I.M., de Alencar, R.O., Junior, R.d.O.C., Davis Jr, C.A.: An ontological gazetter for geographic information retrieval. In: GeoInfo. pp. 21–32 (2010)

16. Moreno-Schneider, J., Rehm, G., Montiel-Ponsoda, E., Rodriguez-Doncel, V., Revenko, A., Karampatakis, S., Khvalchik, M., Sageder, C., Gracia, J., Maganza, F.: Orchestrating NLP services for the legal domain. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 2332–2340. European Language Resources Association, Marseille, France (May 2020), https://www.aclweb.org/anthology/2020.lrec-1.284
17. Murty, S., Verga, P., Vilnis, L., Radovanovic, I., McCallum, A.: Hierarchical losses and new resources for fine-grained entity typing and linking. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 97–109. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-1010, https://www.aclweb.org/anthology/P18-1010
18. Nothman, J., Ringland, N., Radford, W., Murphy, T., Curran, J.R.: Learning multilingual named entity recognition from wikipedia. Artif. Intell. **194**, 151–175 (Jan 2013). https://doi.org/10.1016/j.artint.2012.03.006, https://doi.org/10.1016/j.artint.2012.03.006
19. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter (2019)
20. Shimaoka, S., Stenetorp, P., Inui, K., Riedel, S.: Neural architectures for fine-grained entity type classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. pp. 1271–1280. Association for Computational Linguistics, Valencia, Spain (Apr 2017), https://www.aclweb.org/anthology/E17-1119
21. Tjong Kim Sang, E.F.: Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In: COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002) (2002), https://www.aclweb.org/anthology/W02-2024
22. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. pp. 142–147 (2003), https://www.aclweb.org/anthology/W03-0419
23. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: Huggingface's transformers: State-of-the-art natural language processing. ArXiv **abs/1910.03771** (2019)