NAMED ENTITY RECOGNITION, EXTRACTION, AND LINKING IN GERMAN LEGAL CONTRACTS

Ingo Glaser / Bernhard Waltl / Florian Matthes

Research Associate, Technical University of Munich, Department of Informatics, Software Engineering for Business Information Systems
Boltzmannstraße 3, 85748 Garching bei München, DE
ingo.glaser@tum.de; http://wwwmatthes.in.tum.de

Research Associate, Technical University of Munich, Department of Informatics, Software Engineering for Business Information Systems
Boltzmannstraße 3, 85748 Garching bei München, DE
b.waltl@tum.de; http://wwwmatthes.in.tum.de

Professor, Technical University of Munich, Department of Informatics, Software Engineering for Business Information Systems
Boltzmannstraße 3, 85748 Garching bei München, DE
matthes@tum.de; https://wwwmatthes.in.tum.de

Keywords: ***Named Entity Recognition and Disambiguation, Legal Text Analysis, UIMA***

Abstract: *The semantic knowledge revealed by the continuously increasing amount of digitized legal documents is highly relevant to the reader. Since documents are mostly available as unstructured data, they are not processable by computer systems. We provide support for this business need by implementing a software component, enabling semantic analysis and structuring of legal contracts. Hence, different approaches to Named Entity Recognition are incorporated into an Apache UIMA pipeline. The evaluation of the developed system, using German legal data, demonstrates the applicability of such approaches.*

1. Introduction

Nowadays, many sectors face the obstacle called digitalization, and so does the legal domain. When talking about digitalization, one must distinguish between unstructured, semi-structured and structured data. In terms of digitizing texts these distinctions must be considered as well [Hashimi 2015]. Structured data can be processed easily and is the simplest way to manage information. However, semi-structured and in particular unstructured data is hard to process. Hence, transforming unstructured or semi-structured data into structured data is an important task in order to manage and process information [Svyatkovskiy et al. 2016].

The rise of legal technology is highlighted by the increasing number of digitized legal documents, in particular legal contracts [Saravanan et al. 2009]. After capturing these, in many cases they are only available as unstructured or semi-structured data and thus barely processable by computer systems. However, the semantic knowledge within such a document is highly relevant to the reader. Considerable added value can be created when modelling and structuring these digitized legal documents properly [Walter 2009]. This is in particular also true due to the fact that lawyers and legal experts use frequently different ways of expression. Having two lawyers creating two contracts with the same intent, the result is most likely two different contracts. Furthermore, legal contracts include a lot of information which is not highly relevant to the reader [Hashimi 2015]. When there is a structured way of revealing the crucial information while neglecting superfluous passages of text of such a document, the resulting view would be the same. This is the main motivation behind this work.

The technical capabilities to accomplish such a task have arisen only very recently. Intensive digital work is becoming more and more attractive, due to the increasing possibilities of text mining, support for data, time, and knowledge [Waltl et al. 2017]. Having the legal technology on a growing branch, along with all the new technical capabilities, as well as the fact that the structuring of text through computer-supported-analysis

is very attractive for the legal domain, further research in terms of semantically analyzing and structuring legal contracts is an interesting and promising task.

This work provides support for the business needs by implementing a software component, enabling semantic analysis and structuring of legal contracts. In order to implement this process, common natural language processing (NLP) tasks like named entity recognition (NER) and named entity disambiguation (NED) are incorporated into an Apache unstructured information management architecture (UIMA) pipeline.

2. Related Work

There has been considerable work on NER, in particular for the English language [Sang/de Meulder 2003]. Jurafsky summarizes much of this research in his book Speech & Language processing [Jurafsky 2007, p. 349 ff.]. Borthwick [Borthwick/Grisham 1999] gives a good overview about the research in this field. When recognizing named entities (NEs), we distinguish between distinct categories. The literature as well as various shared tasks suggest different categorizations. Borthwick [Borthwick/Grisham 1999] for instance, uses the following categories in his work: person, location, organization, date, time, percentage, monetary value, and «non-of-the-above». The CoNLL-2003 shared task suggests to use just three, respectively four, categories: person, location, organization, and «other» [Sang/de Meulder 2003]. Such a categorization of NE types often depends on the domain. For this work, the suggestion from CoNLL-2003 was adopted, enhanced by some of the categories from the literature. This led to the following set of categories: person, organization, location, date, money value, reference, and «other».

Developing a NER system for German is a difficult, but well researched task. German is a wide-spread and comparatively well-resourced language [Benikova et al. 2015]. However, only three notable datasets exist, namely CoNNL-data [Sang/de Meulder 2003], an extension to user-generated content by Faruqui and Padó [Faruqui/Padó 2010] and the NoSta-D NE dataset [Benikova et al. 2014]. Even though there have been a lot of German NE taggers, only one is freely available, developed by Benikova et al. [Benikova et al. 2015]. Faruqui and Padó [Faruqui/Padó 2010] created a German NER model for the Stanford NER, which is licensed under the GNU General Public License. Stanford NER is also known as a conditional random field (CRF) classifier [Finkel et al. 2005]. A NER system based on the maximum entropy model (MEM) for German was developed by Bender et al. [Bender et al. 2003]. Chieu and Ng [Chieu/Ng 2002] as well as Curran and Clark [Curran/Clark 2003] who created similar systems for the German language in the course of the CoNNL-2003 shared task. Florian et al. [Florian et al. 2003] and Klein et al. [Klein et al. 2003] came up with an approach to German NER, using a combination of MEM and other techniques. The CoNNL-2013 [Benikova et al. 2014] shared task caused further research in German NER.

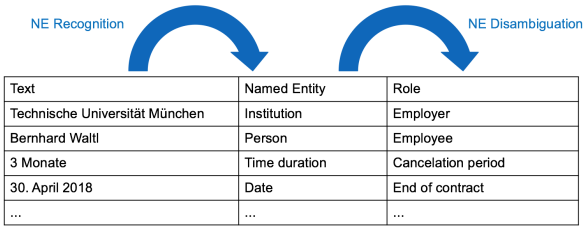
Even though legal informatics is growing [Waltl et al. 2017], not much research has been conducted concerning NER in the legal domain. Dozier et al. [Dozier et al. 2010] discusses NER in legal documents such as US case law, depositions, pleadings, and other trial documents. Hereby they differentiate between judges, attorneys, companies, jurisdictions, and courts as NE types. They outline three methods in their discussion: lookup, context rules, and statistical models. A nested NER system with neural networks was defined and implemented by Reimers et al. [Reimers et al. 2014]. The system was developed during the GermEval-2014 shared task and got inspired by the findings of Collobert et al. [Collobert et al. 2011].

NED is defined as the process of linking a NE to an entry in some resource, which is the correct one for the context of occurrence [Bunesca/Pasca 2006]. When we talk about linking or disambiguating NEs, the literature often uses the term named entity linking (NEL) or NED for that task. In this work, the term NED is used in order to describe the task of linking a NE to a semantic function or role. Manning dedicates a whole chapter in his book Statistical NLP [Manning 1999, p. 229 ff.] to the linking of words to senses. He suggests different techniques for word sense disambiguation (WSD): supervised disambiguation, unsupervised disambiguation as well as a dictionary-based disambiguation. The same suggestions are made by Jurafsky

in his book Speech & Language Processing [Jurafsky 2000]. By applying small changes, those approaches may be feasible to NED as well.

3. Conceptual Overview

In order to achieve the goal of semantically analyzing and structuring legal contracts, a process consisting of NER and NED was defined. This concept serves as a reference for the actual implementation. For the explanation of this concept, the example of an employment agreement is taken. The first step towards the extraction of semantic knowledge is the application of NER. The goal of this step is to extract all NEs in the agreement. The result of this task is illustrated in the left column of Figure 3.1.



Text	Named Entity	Role
Technische Universität München	Institution	Employer
Bernhard Waltl	Person	Employee
3 Monate	Time duration	Cancelation period
30. April 2018	Date	End of contract
...	...	...

Figure 3.1: Conceptual overview of the recognition and disambiguation process

Once the NEs are recognized, the actual disambiguation can take place. Each contract needs to be modeled. For that reason, the prototypical implementation allows the definition of such a semantic model. For the sake of this example, a model with roles like *Employer*, *Employee*, *Cancellation period* and *End of contract* is assumed. Figure 3.1 includes both steps of the NER and NED process. The phrase «*Technische Universität München*» is recognized and classified as an organization in the first step. During the second step, the NE is linked to the respective role *Employer*. This is the basic concept behind the software component, being implemented in the course of this paper. Figure 3.2 depicts the described concept within a conceptual software architecture. It reflects the basic architecture of the semantic analysis component. The Semantic Analysis Component consists of two sub components, that is: (1) the *Named Entity Recognition Component*, and (2) the *Named Entity Disambiguation Component*. The former gets a *Contract* as input and performs NER on it. Optionally for the templated NER, it involves a *Template*, too. This results in an *Annotated Contract*. This *Annotated Contract* is forwarded to the disambiguation component. Depending on the approach, a *Template*, external resources such as knowledge bases as well as artificial intelligence (AI) is used to create the *Structured Contract*. The three implemented approaches of this work are discussed in Section 4.

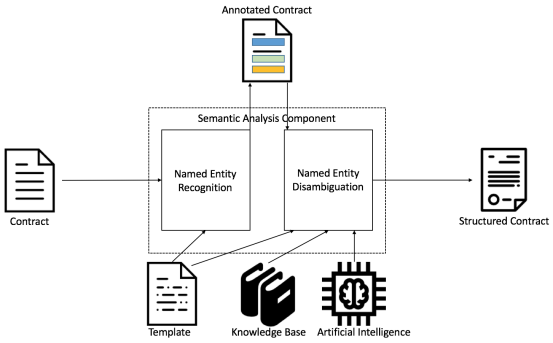


Figure 3.2: Conceptual architecture of the semantic analysis component

4. Recognizing Named Entities and Linking towards Semantic Models

The prototypical implementation has been done in an existing legal data science environment that allows the collaboration on legal documents. The environment is a web application implemented with a Java backend. Apache UIMA, developed by IBM and also used in IBM Watson, conduces as a reference architecture. The Apache UIMA utilizes the use of pipelines to process legal texts. This is achieved by a state-of-the-art pipes & filters architecture. Such a pipeline allows the incorporation of various NLP tasks, by implementing Apache UIMA components. The different steps such as tokenization, sentence splitting, or part-of-speech (POS) tagging are executed subsequently. This way, the pipeline is executed on a legal text. The processing results are stored in UIMA common analysis system (CAS) objects for further processing. More details on the data science environment, the different components and the base line architecture to perform computational intensive data analysis processes on large text corpora can be found in [Waltl et al. 2016].

4.1. Named Entity Recognition Pipelines

Three different approaches to NER are utilized in this paper. Each approach was integrated into an Apache UIMA pipeline, as described in the previous section, which is embedded in our legal data science environment. The following sections treat each of the three pipelines in greater detail.

4.1.1. GermaNER

GermaNER, a generic German NE tagger that can be readily used from command line or integrated into any NLP application to automatically tag NEs, was used. For the latter, the tagger is available as an Apache UIMA component. A crucial contribution to the NER community has been made due to the fact that this system is under a permissive license that allows academic and commercial use without licensing fees. This system integrates a CRF [Lafferty et al. 2001] for sequence tagging. CRFs are scalable, highly accurate and easy to use as the training data can be prepared without the need of ML experts [Hoefel/Elkan 2008]. The CRFsuite by Okazaki [Okazaki 2007] has been integrated into a clearTK UIMA framework [Bethard 2014]. This enables more convenient training, feature annotation, classification and entity extraction [Benikova et al. 2015]. Hereby the system is highly configurable, as it allows the user to either use the built-in model, or train it with new training data and feature sets, while the standard model is optimized with the existing feature set. A nice benefit of this system is its NER tagger pipeline. The pipeline consists of distinct components integrated into an UIMA [Ferucci/Lally 2004] pipeline written in the Java programming language. This allows us to easily integrate the tool into our system.

GermaNER only accepts the CoNLL-2013 format as input. Such a file contains one token per line, while sentence should be separated by a blank line. The output of the tagger is a tab separated file. The first column corresponds to the same as in the input file. In the second column, the predicted NE tag is stored in form of the beginning-inside-outside (BIO) scheme. The BIO-scheme suggest to learn classifiers that identify the beginning, the inside and the outside of the text segments [Ratinov/Roth 2009]. Our system however holds legal documents at html-formatted texts. For that reason, our pipeline first removes the html tags before we perform tokenization and sentence splitting. Eventually, a specific transformer creates a CoNLL representation of the actual legal document. All this information is stored in a CAS object and forwarded through the pipeline. Eventually the GermaNER component tries to identify the NEs. While GermaNER is responsible for the extraction of persons, organizations, locations and others, rule-based approaches are applied to the other types. We have used regex rules to identify dates and money values. The work from Landthaler et al. [Landthaler et al. 2016] was used to identify references. Since GermaNER uses its own type system, the recognized NEs are transformed into our own type system within Lexia. This step allows us to further process the recognized NEs. Afterwards, the pipeline finishes with a post-processing, in order to enrich the html representation with the gained information.

4.1.2. DBpedia Spotlight

DBpedia is an interlinking hub in the web of data, enabling access to many data sources in the linked open data cloud [Mendes et al. 2011]. DBpedia contains about 3.5 million resources from Wikipedia. The ontology is populated with classes such as places, persons or organizations. Furthermore, fine-grained classifications like soccer players or IT companies are existing. Resources possess attributes as well as relations to each other [Daiber et al. 2013]. Mendes et al. [Mendes et al. 2011] developed DBpedia Spotlight Annotator to enable the linkage of web documents with that hub. It is a system to perform annotation tasks on text fragments, such as documents, paragraphs or sentences, provided by a user. Hereby, the user wishes to identify URIs for resources mentioned within that text. This can be seen as a typical NER system.

From a technical point of view, the integration of the DBpedia Spotlight UIMA component has been done in the same fashion as the GermaNER component. Our pipeline performs html stripping in order to feed plain text to the DBpedia Spotlight UIMA component. After the NE extraction, the component described in the previous section for post-processing is reused, as well as the type system transformation.

4.1.3. Templated Named Entity Recognition

In the course of this study, a new approach to NER in contracts was developed. This approach is called templated NER. The creation of a contract is mainly a manual and labour intensive task. Legal practitioners need to be able to understand the requirements of a deal to define a suitable contract [Rousseau/Greller 1994]. However, existing contracts are often refined, instead of creating a new contract from scratch. Over time, this lead to the existence of contract templates. For simple circumstances such as a rental deal, contract templates exist. The legal expert only needs to fill the placeholders with the respective information. Contract creation via templates is pretty common today [Minakov et al. 2007]. Having this in mind, NER can be carried out easily on contracts, defined by a template, as long as the template is at hand as well.

The intuition behind this templated NER approach is that if we compare an actual contract with its template, only the populated information remains as differences. When thinking about relevant information in a contract, mostly NEs emerge. With other words, when a contract template is filled in the majority of information are NEs. Of course, this method basically just picks off the low hanging fruits, nonetheless it is a valid NER system for that specific kind of contracts.

> Vertragsgegenstand ist die Lieferung von insgesamt −−Verkaufsprodukt.Menge−− Stück −−Verkaufsprodukt.Name−− des Herstellers −−Verkaufsprodukt.Hersteller−−.

Figure 4.1: Example sentence from a template

A possible example of a sentence from such a template is shown in Figure 4.1. For the placeholders, a concept has been used, where two dashes followed by some word ensued by another two dashes indicate a NE. With other words, the following regex highlights a placeholder «*(–)*.(–)*». During the contract creation process, such a template may be filled as follows.

> Vertragsgegenstand ist die Lieferung von insgesamt 12 Stück MacBook Pros des Herstellers Apple.

Figure 4.2: Example sentence from an instantiated template

Figure 4.2 depicts an instantiated template. The goal of a templated NER approach is to extract the three NEs: (1) «*12*», (2) «*MacBook Pros*», and (3) «*Apple*». By comparing the template and the instance, it is

shown that those three NEs are the only difference between the two sentences. That is already the concept behind templated NER, which is implemented in the course of this study. In order to implement it, Google's *diff-match-patch*[1] (DMP) algorithm is utilized. The algorithm is based on Myers' diff algorithm [Myers 1986]. When executing the algorithm, only pairs of differences augmented by the diff-option (equal, insert, and delete) are returned. Due to this, no types can be extracted, but just the NEs. However, the next section deals with the NED, which actually goes even further as having a type system for NEs.

4.2. Named Entity Disambiguation

As already mentioned in Section 2, legal data corpora are rare, in particular when talking about annotated data. In order to perform NED to link NEs towards semantic functions such as *Employee*, huge data sets are required. We have no access to such data yet. For that reason, we have not built a classifier for NED as of now. However, the templated NER approach from the previous section can be extended to perform NED.

Having a template of a contract, we can create a semantic model with regard to the template. To be more precise, a template consists of various placeholders, where the actual content is inserted during the contract creation process. A semantic model of such a contract can be created while adding each placeholder to the model (as a type or an attribute). Going even further, and regarding the placeholder names in the model, a linking is already created. Of course, the linking is established manually and this is basically just picking up the low hanging fruits, but this enables the straight process from NER, via NED towards a populated semantic model of a contract.

5. Evaluation

5.1. Evaluation Method

In terms of evaluating the NER approaches, each of the three implemented techniques was first evaluated, before the obtained results were compared. Different evaluation metrics exist for the evaluation of NER systems [Nadeau/Sekine 2007]. The assessment is basically to check the system's ability to find the boundaries of names and their correct types. The evaluation in this work only approves a tagged span when it is equal to the span enclosing the actual NE. With other words, perfect matching is required. For the evaluation, the state of the art approach of IR and IE has been used. This means that a confusion matrix was created for every approach. Based on this confusion matrix, each NE type was evaluated first [Jurafsky 2000], by means of precision, recall, and F1 measure. Afterwards the overall measures for each method were determined. For this the accuracy was not used, because it is quite superfluous for a NER system. Just a minority of all tokens from a given text represent NEs. Thus, TN of such a system are very high relatively to the total number of tokens [Jurafsky 2000, Sang/de Meulder 2003].

5.2. Data Set

As already mentioned in Section 2, data sets for NER barely exist within the legal domain. As a consequence, the evaluation data set for this work has to be created manually. Since the focus of this work is to semantically analyze and structure legal contracts, an evaluation corpus consisting of contracts would be a great fit. However, due to a lack of contracts in this study, the evaluation of the GermaNER pipeline and DBpedia Spotlight pipeline was performed on judgments. A corpus of 500 judgments from the law of tenancy of the 8th Zivilsenat of the German BGH was downloaded from *Rechtsprechung im Internet*[2]. A random selection of 20 judgments from this corpus constitute the evaluation dataset used for this assessment. The data set consisted of 25'423 token. Since these judgments were not annotated, a gold standard was created by hand as well.

[1] http://code.google.com/archive/p/google-diff-match-patch (all websites last visited in January 2018).
[2] http://www.rechtsprechung-im-internet.de.

NE Types	PER	ORG	LOC	DA	MV	REF	OTH	O
Count	114	106	45	267	78	310	182	24'314

Table 5.1: Composition of the evaluation data set

The composition of this data set is shown in Table 5.1. This distribution of NE types is pretty common for the legal domain. The abbreviations used in the table are applied for the rest of this chapter, while *O* is referring to *not a NE*. Templates do not exist for judgments and thus, the templated NER approach had to be evaluated on legal contracts. For this reason, 5 different contracts were selected: (1) a purchase agreement, (2) a lease contract, (3) an employment agreement, (4) a lease agreement for commercial premises, and (5) a GmbH contract. This ended up in a total of 7'790 token, including the distribution of NEs as depicted in Table 5.2.

NE Types	PER	ORG	LOC	DA	MV	REF	OTH	O
Count	14	8	23	38	23	25	46	7'614

Table 5.2: Composition of the evaluation data set for templated NER

5.3. Assessment

In the course of this work, only templated NED is implemented. An evaluation based on measures such as precision, recall, and F1 measure, as it has been done for NER approaches, is not suitable at this point. The concept behind templated NED is quite simple, (cf. Section 4.2). The disambiguation is solved by means of comparing the placeholder names in the template with the type and attribute names in the semantic model. This linking works always, as long as the user chooses the names accordingly. Hence, an error only occurs if there is a mismatch between the naming in the semantic model and the contract template. It does not make sense to evaluate the person who defined the evaluation set. One could suggest to conduct the evaluation by incorporating the whole process from the textual contract representation via NER and NED to the populated semantic model. However, only NEs recognized by the templated NER can be linked and thus, the evaluation result would mirror the results from assessing templated NER. This is the reason why no evaluation was performed on templated NED, but just for NER. In order to get more accurate results, the evaluation was performed over three rounds for each method. The average values for these three rounds was then used to answer the three main questions, as discussed in the next sections.

5.3.1. Which implemented NER pipeline performs best?

It is not common to compare three systems, whereas one of the systems was evaluated on a different evaluation data set. However, for this work it was not possible to evaluate all three approaches on the same data, as already mentioned in Section 5.2. Table 5.3 summarizes the results of this evaluation.

System	Per-entity F1							Overall		
	PER	ORG	LOC	DA	MV	REF	OTH	P	R	F1
Templated	0.88	0.77	0.82	0.86	0.88	0.93	0.71	0.94	0.91	0.92
GermaNER	0.35	0.71	0.45	0.91	0.89	0.91	0.33	0.98	0.68	0.80
DBpedia	0.51	0.76	0.52	0.91	0.86	0.91	0.59	0.87	0.87	0.87

Table 5.3: NER performance of all three systems over the evaluation data set

The overall performance of templated NER (F1) clearly exceeds the results of the pipelines incorporating GermaNER and DBpedia Spotlight. Comparing just the latter two, GermaNER reveals the better overall precision (0.98) over DBpedia Spotlight (0.87). This is not unexpected due to the fact that knowledge bases consist of a huge variety of different terms, which leads to the recognition of many tokens actually not representing any

NE of interest. On the other side, the higher overall recall of DBpedia Spotlight (0.87 over 0.68), is not a surprise either, caused by the same fact. Hence, DBpedia Spotlight overall outperformed GermaNER (overall F1 of 0.87 over 0.80). Nonetheless, templated NER is a very suitable and outstanding approach for NER on legal contracts, as long as templates exist. Both systems, GermaNER and DBpedia Spotlight were incorporated into a pipeline, but the system implemented in this work offers the possibility for errors as well. This leads to the assumption that the two tools could perfrom much better, when evaluating independently from this system. When looking at different evaluations, such as the CoNNL-2013 shared task [Benikova et al. 2014, Benikova et al. 2015] or the evaluation of DBpedia Spotlight [Mendes et al. 2011], this assumption is partially confirmed.

5.3.2. Which NE type is recognized best?

The answer to this question can be given easily. The types being recognized via rule-based approaches (date, money value, and reference) obviously perform the best. This is mainly due to the structure which represents types like those. Once those types are neglected, organizations perform the best.

5.3.3. Which NE type is recognized worst?

The type *other* has in its nature that it not just comprises miscellaneous entities, it also often covers NE of other types, falling through their own classifiers. Moreover, there exist a huge variety of different NE types, excluding the set of categories used in this work. All those types shall be recognized by the *other* type. This may be feasible for a system such as DBpedia Spotlight, but statistical approaches, and even the templated NER approach, clearly fail to detect all NEs of such types.

6. Conclusion and Outlook

6.1. Conclusion

In this work, a prototypical implementation enabling the semantic analysis and structuring of legal contracts was designed and developed, utilizing Lexia. Common concepts and strategies found in the literature form the basis for the developed requirements and solutions. Three different NER methods, namely GermaNER, DBpedia Spotlight, as well as an individually developed solution called templated NER, are responsible for the extraction of NEs. The disambiguation of the recognized entities towards semantic functions, which are represented in semantic models, is done by NED. When having individual domain specific models, it is very hard to incorporate proper NED. This is mainly because of the lack of training data. In order to achieve the breakthrough from a legal contract to a populated contract model, this work implemented the templated NED approach. By means of this approach, a contract model was successfully populated with semantic information within the contract. The pipeline architecture is based on Apache UIMA and thus can be easily extended. This enables the integration of existing analysis engines, used in Lexia into the pipelines for NER and NED. Future work on the semantic analysis of legal contracts can be easily integrated into the existing pipeline architecture. The evaluation of the different approaches used in this study showed that templated NER is an appropriate approach for recognizing NEs within legal contracts that are based on templates. It also revealed the applicability of common NER tools like GermaNER or DBpedia to the legal domain, but also showed the necessity of future research in this field.

The prototypical implementation along with the outcomes of this work are an additional knowledge base and provide an appropriate starting point for future research in the fields of NER and NED on German legal contracts.

6.2. Limitations and Future Work

Even though this work provides a good starting point for further work, some limitations must be kept in mind.

Although each evaluation experiment was conducted three times in order to obtain a significant result, and even though the results looked still quite promising, the evaluation experiments require further replication to attain a statistically significant value. This is caused in particular by the manual creation of the evaluation date set, which is furthermore very small. Moreover, the evaluation of templated NER was obviously conducted on a different data set than the other two approaches (GermaNER and DBpedia Spotlight) and hence, the comparison of the three methods it not suitable.

The results of the GermaNER as well as the DBpedia Spotlight pipeline may not reflect their actual performance. The NE types, considered in this work are: person, organization, location, date, money value, reference, and *other*. Dates, money values and references were only detected using rule-based methodologies, but incorporated into both pipelines. This already refines the results. In addition, these two technologies were not used in isolation, but utilized by the prototypical implementation of this work. Hence, system errors are conveyed to the two tools.

The templated NER approach is only suitable for corpora where a small number of templates define a massive number of contracts. But if that is the case, by diligently defining the template placeholders and incorporating templated NED, spectacular results can be achieved. Due to this, the implementation of the templated approaches to NER and NED are a promising approach for the semantic analysis and structuring of legal contracts. For the future work, it could be an interesting approach to train the models of NER tools, such as GermaNER, specifically for the German legal domain. If at the same time, big evaluation data sets arise, the NER task on German legal contracts could be improved considerably. The next step then would be to build classifiers for the disambiguation of those recognized NEs, towards individual semantic models. Eventually, this may lead to digitized and properly structured legal contracts.

7. References

Bender, Oliver/Och, Franz Josef/Ney, Herman, Maximum entropy models for named entity recognition, In Proceedings of the seventh conference on Natural language learning at HLT-NAACL, 2003, pp. 148–151.

Benikova, Darina/Biemann, Chris/Kisselew, Max/Padó, Sebastian, Germeval 2014 named entity recognition shared task: companion paper, Organization, 7:281, 2014.

Benikova, Darina/Muhe, Seid/Prabhakaran, Yimam/Biemann, Chris, Germaner: Free open german named entity recognition tool, In In: Proc. GSCL-2015, 2015.

Borthwick, Andrew/Grisham, Ralph, A maximum entropy approach to named entity recognition, PhD thesis, New York University, Graduate School of Arts and Science, 1999.

Bunesco, Razvan C./Pasca, Marius, Using encyclopedic knowledge for named entity disambiguation, In Eacl, 2006, pp. 9–16.

Chieu, Hai Leong/Ng, Heww Tou, Named entity recognition: a maximum entropy approach using global information, In Proceedings of the 19th international conference on Computational linguistics, 2002, pp. 1–7.

Collobert, Ronan/Weston, Jason/Bottou, Léon/Karlen, Michael/Kayukcuoglu, Koray/Kuksa, Pavel, Natural language processing (almost) from scratch, Journal of Machine Learning Research, August 2011, pp. 2493–2537.

Curran, James R./Clark, Stephen, Language independent ner using a maximum entropy tagger, In Proceedings of the seventh conference on Natural language learning at HLT-NAACL, 2003, pp. 164–167.

Cuceran, Silviu, Large-scale named entity disambiguation based on wikipedia data, 2007.

Daiber, Jocahim/Jakob, Max/Hokamp, Chris/Mendes, Pablo N., Improving efficiency and accuracy in multilingual entity extraction, In Proceedings of the 9th International Conference on Semantic Systems, 2013, pp. 121–124.

Dozier, Christopher/Kondadadi, Ravikumar/Light, Marc/Vaccher, Arun/Veeramachaneni, Sriharsha/Wudali, Ramdev, Named entity recognition and resolution in legal text, In Semantic Processing of Legal Texts, Springer, 2010, pp. 27–43.

Faruqui, Manaal/Padó, Sebastian, Training and evaluating a german named entity recognizer with semantic generalization, In KONVENS, 2010, pp. 129–133.

Ferucci, David/Lally, Adam, Uima: an architectural approach to unstructured information processing in the corporate research environment, Natural Language Engineering, 10(3-4), 2004, pp. 327–348.

Finkel, Jenny/Grenager, Trond/Manning, Christopher D., Incorporating non-local information into information extraction systems by gibbs sampling. In Proceedings of the 43rd annual meeting on association for computational linguistics, 2005, pp. 363–370.

Florian, Radu/Ittycheriah, Abe/Jing, Honguyan/Zhang, Tong, Named entity recognition through classifier combination, In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, pp. 168–171.

Hasmi, Mustafa, A methodology for extracting legal norms from regulatory documents, In Enterprise Distributed Object Computing Workshop (EDOCW), 2015, IEEE 19th International, pp. 41–50.

Jurafsky, Dan, Speech & language processing, Pearson Education, India, 2000.

Klein, Dan/Smarr, Joseph/Nguyen, Huy/Manning, Christopher D., Named entity recognition with character-level models, In Proceedings of the seventh conference on Natural language learning at HLT-NAACL, 2003, pp. 180–183.

Landthaler, Jörg/Waltl, Bernhard/Matthes, Florian, Unveiling References in Legal Texts: Implicit versus Explicit Network Structures, In Netzwerke / Networks, Tagungsband des 19. Internationalen Rechtsinformatik Symposions IRIS 2016, Salzburg, Austria, 2016.

Manning, Christopher D./Schütze, Hinrich, Foundations of statistical natural language processing, 1999, MIT Press.

Mendes, Palo N./Jakob, Max/Garcia-Silva, Andrés/Bizer, Christian, Dbpedia spotlight: shedding light on the web of documents, In Proceedings of the 7th international conference on semantic systems, 2011, pp. 1–8.

Myers, Eugene W., An o (nd) difference algorithm and its variations, Algorithmica, 1(1), 1986, pp. 251–266.

Nadeau, David/Sekine, Satoshi, A survey of named entity recognition and classification, Lingvisticae Investigationes, 2007, 30(1), pp. 3–26.

Ratinov, Lev/Roth, Dan, Design challenges and misconceptions in named entity recognition, In Proceedings of the Thirteenth Conference on Computational Natural Language Learning, 2009, pp. 147–155.

Reimers, Nils/Eckle-Kohler, Judith/Schnober, Carsten/Kim, Jung/Gurevych, Iryna, Germeval-2014: Nested named entity recognition with neural networks, In Workshop Proceedings of the 12th Edition of the KONVENS Conference, 2014, pp. 117–120.

Sang, Erik F./Tjong, Kim/de Meulder, Fien, Introduction to the conll-2003 shared task: language-independent named entity recognition, In CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL, 2003, pp. 142–147.

Saravanan, M./Ravindran, B./Raman, S., Improving legal information retrieval using an ontological framework, Artificial Intelligence and Law, 2009, 17(2) pp. 101–124.

Svyatkovskiy, K./Imai, M./Kroeger, M./Shiraito, Y., Large-scale text processing pipeline with apache spark, In Big Data (Big Data), 2016, IEEE International Conference, pp. 3928–3935.

Walter, Stephan, Definition extraction from court decisions using computational linguistic technology, Formal Linguistics and Law, 2009, 212, pp. 183.

Waltl, Bernhard/Mathhes, Florian/Waltl, Tobias/Grass, Thomas, LEXIA: A data science environment for Semantic analysis of german legal texts, In Netzwerke / Networks, Tagungsband des 19. Internationalen Rechtsinformatik Symposions IRIS 2016, Wien/Bern, Austria, 2016.

Waltl, Bernhard/Landthaler, Jörg/Scepankova, Elena/Matthes, Florian/Geiger, Thomas/Stocker, Christoph/Schneider, Christian, Automated extraction of semantic information from german legal documents, In Trends und Communities der Rechtsinformatik / Trends and Communities of Legal Informatics, Tagungsband des 20. Internationalen Rechtsinformatik Symposions IRIS 2017, Salzburg, Austria, 2017.