Sentence Boundary Detection in German Legal Documents

Ingo Glaser[a], Sebastian Moser and Florian Matthes
Chair of Software Engineering for Business Information Systems, Technical University of Munich, Boltzmannstrasse 3, 85748 Garching bei München, Germany

Keywords: Sentence Boundary Detection, Natural Language Processing, Legal Document Analysis.

Abstract: Sentence boundary detection on German legal texts is a task which standardized NLP-systems have little or no ability to handle, since they are sometimes overburdened by more complex structures such as lists, paragraph structures and citations. In this paper we evaluate the performance of these systems and adapt methods directly to the legal domain.
We created an annotated dataset with over 50,000 sentences consisting of various German legal documents which can be utilized for further research within the community. Our neural networks and conditional random fields models show significantly higher performances on this data than the tested, already existing systems. Thus this paper contradicts the assumption that the problem of segmenting sentences is already solved.

1 INTRODUCTION

For a human being, the segmentation of a given text into sentences is a rather easy task. However, when trying to summarize this segmentation in rules, which could be utilized by a machine, one quickly realizes how complex this task actually is. On the other side, sentence boundary detection (SBD), which is the task for a machine to detect sentence boundaries, is an important processing step in almost any natural language processing (NLP) task. As a result, SBD has a long history of research (Read et al., 2012) within the community of NLP.

Many SBD papers exist and reveal high performances. (Read et al., 2012) provide an excellent and comprehensive overview of such systems. Thus, one could see the problem of SBD as solved. On the other side, domains such as the legal domain significantly differ from normal texts encountered daily. Content, structure, and language are vastly different and more difficult to comprehend. This is not only a problem for every layman who tries to understand those laws but also for a software system processing these texts. There is no universal structure inherent to all texts, which leads to a performance decrease when processing structurally different documents without a ground truth.

As already stated, the structure and contents of a legal text differ from a normal text and even within the legal domain there is a vast variation of documents. The basic assumptions previously made when dealing with SBD are not always applicable in the legal domain. For example, it is often defined as an abbreviation disambiguation task, which is only a partial solution in the legal domain. Here a sentence is not always linked to dots and other characters denoting an abbreviation.

That is why many researchers reported difficulties with standard NLP tools in the legal domain. (Wyner, 2010) achieved promising results when extracting rules from regulatory texts but also encountered problems with the *Stanford Parser* while processing different phrase structures. ”Linguistic indicators, structure, and semantic interpretations which interact with domain knowledge” (Wyner and Peters, 2011) are utilized by legal experts, but a computer is not entirely capable of modelling or utilizing those pieces of information. The legal domain poses issues for automation of legal work because a ”large amount of linguistic, domain, and communicative knowledge” (Moens, 2001) is needed for tasks such as information retrieval. Furthermore, various research in the field of AI&Law, revealed issues with SBD for the legal domain. (Ashley, 2017) provides a comprehensive overview of the most recent research in this area.

For this research, we formulated the research hypothesis, that SBD is not perfectly solved and requires adaption of existing methods, particularly for the Ger-

[a] https://orcid.org/0000-0002-5280-6431

Glaser, I., Moser, S. and Matthes, F.
Sentence Boundary Detection in German Legal Documents.
DOI: 10.5220/0010246308120821
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 812-821
ISBN: 978-989-758-484-8

man legal domain. One could even generalize this hypothesis across all domains and claim that with rising complexity there might be a need to re-evaluate previous sentence segmentation systems on complex texts. Therefore, we structured our paper as follows. After this introduction, Section 2 describes the taxonomy of sentences used in this paper. In Section 3 we describe and publish a dataset from the German legal domain to train and evaluate SBD systems. This dataset is not just used for this purpose, but shall be also used within the community for further research. Our experiments are described in Section 4, while the evaluation, including an error analysis, is discussed in Section 5. Finally, this paper concludes with Section 6.

2 LEGAL SENTENCE BOUNDARY DETECTION

Most of the time, legal documents are structured in smaller parts such as paragraphs, clauses, and so on. Their content is often only loosely linked to the other parts of the text. Sentences are long and might incorporate complex structures such as lists. Even within one legal document type (such as judgments or laws) there does not exist a uniform formatting style or structure. These are all problems a sentence segmentation system has to deal with in order to produce pure sentences from the mixture of such distinct textual units. As a result, a corpus constituting German legal documents requires different approaches, particularly a different notion of sentences, in contrast to other domains. That includes not only diverse domains, but also different languages as the legal system varies heavily between countries. Therefore, in the following the taxonomy utilized in this paper is discussed.

A **sentence** can be seen as "a sequence of tokens [...] that together form a full stand-alone grammatical sentence" (Savelka and Ashley, 2017b). Translating this to the legal domain is not trivial. There are a lot of different textual structures which do not fit this definition, but still need to be treated as stand-alone or independent segments of text. Thereby, distinguishing structural information from normal text is a significant challenge.

As a result, we came up with an own taxonomy of sentences for German legal documents. That taxonomy is described in the remainder of this section. A textual part of a legal document is seen as a **sentence** if its combination with the next or previous sentence destroys that sentence's meaning or adds unrelated pieces of information. This is the basis of all the sentence boundary decisions made in the following section. This does also align with the critique by Read et al. (Read et al., 2012), who saw the focus of sentence boundary detection on sentence terminating characters as an oversimplification.

To define a rule set for sentence boundaries, the collected documents were analyzed and the different types of sentences extracted. The following sentence categories are heavily influenced by Savelka and Ashley (Savelka and Ashley, 2017a) but adapted to the German legal domain. They are motivated by the definition found above and define the basis for the all the annotations.

Most of the sentences found in legal documents are **standard sentences**, i.e. they have a subject, object and verb in the right grammatical order. Additionally, they are concluded by a sentence-terminating character such as a period, semi-colon, question or an exclamation mark.

Another structure very closely related to standard sentences are **linguistically transformed sentences**, which are in some way altered, but still have the basic building blocks of a standard sentence. By changing the word order or making other small adjustments, these sentences can be transformed into a standard sentence.

There are more complex structures in legal texts, all collected in the category of **special sentences**. Those are linguistic units with stand-alone content that cannot be incorporated into other sentences. The following paragraphs describe those different textual units which are considered sentences in legal documents if not stated otherwise.

Headlines are one of the most common textual units found in legal documents. They are used to determine the structure of the text and to show which parts of the documents are related to another, thus they convey crucial information to understand the overall structure of the text.

Citations are often part of judgments and thus need to be addressed in this paper. They are used to link different documents and passages. Citations in the middle of a sentence are not annotated (if they are not a sentence on their own), otherwise they are treated normal sentences with additional tokens at the beginning and the end. **Ellipses/Parentheses** are annotated in the same way.

One of the most obvious structures in legal documents are **lists**, which have a wide variety of different types. Lists of **stand-alone sentences** or **alternative sentence ends** convey no message beyond their own list entry. Thus, they are annotated as individual sentences. Another variation of lists are **enumerations**. They consist of short entries, which lack a verb or a concluded operation. These lists can be found in sentences with a variable middle part, e.g. there are different possible subjects or objects for a clause. The list items are not individual sentences but form one whole sentence because they do not conclude any action. All of those list variations may start with a sentence concluded by a colon which is then individually annotated. **Lists of lists** are treated in a recursive way and the decision on the sentence boundary is based on the sub-entries. If one entries consist of multiple sentences, each is individually annotated. These rules follow de Maat (de Maat, 2012) who states that lists can form “single sentences [...] [though] the list items are often referenced separately”. This more fine grained segmentation is motivated despite the fact that thus incomplete sentences can be created.

Definitions are a combination of a headline and (multiple) sentences, thus their individual parts are annotated accordingly. **Data fields** which hold the information about dates, people, locations, etc. and case names are similar. **Endnotes/Footnotes** can also be seen as a combination of headline and sentences.

Page numbers are, if not already removed from the text, only annotated if they are found between two sentences.

Mixed cases of all the previous structures are also possible. The described textual units are only the basic building blocks found in a legal document. There is a large amount of variation even within the same legal document type. In such cases, the smaller segmentation is chosen, e.g. in a list which combines an enumeration and a list, the individual parts of the enumeration would also be annotated as sentences to avoid any inconsistency within the list.

The aim of this paper is not to create the perfect definition for a sentence in the legal domain but to create a logical, consistent and practical rule set to work with. This paper sometimes produces textual parts which cannot be described as complete sentences in a grammatical way because they are lacking essential parts of speech. But those segments are needed for further processing steps, as overall they reduce the complexity of the given text part.

Table 1 provides a few examples for the different types of special sentences based on our taxonomy. As such examples can be quite complex, we provide the reference to selected norms from the German Civil Code (BGB).

3 DATASET

In order to get a broad overview about the performance of SBD in the (German) legal domain, a diverse set of different documents was collected. The main focus was on judgments and laws with around 20,000 sentences each, but privacy policies (PRVs) and terms of services (ToS) are also part of the collection. The dataset consists of approximately 52,000 sentences. The judgments and laws are used for training and the other document types mainly for validation and testing.

The dataset contains the BGB, the German Consititution (GG), Code of Social Law (SGB) 1 to 3 and the Criminal Code (StGB), with the BGB being by far the longest text. We have made this specific selection of laws on purpose to capture a broad variety of different areas of law with it’s distinct linguistic characteristics.

The judgments were collected from the website *Bayern.Recht*[1], which is a collection of many judgments from Bavarian courts. They are from different legal domains and courts (Verfassungs-, Ordentliche, Verwaltungs-, Finanz-, Arbeits- and Sozialgerichtsbarkeiten) and thus have a wide difference in structure and content. The PRVs of major tech companies were collected and the ToS of online shops gathered by Koller (Koller, 2019) are also used. A detailed statistic on the documents and the number of sentence boundaries can be seen in Table 2.

The documents are annotated with the help of an already existing online annotation tool developed by Savelka (not published yet) and later on with a newly created graphical user interface which is also used to display and analyze the classification results. The last non-punctuation token or word in every sentence is annotated as the end of the sentence. Each document is saved as a separate JSON file containing the text and the sentence boundary annotations with their position in the text. The annotations, based on the taxonomy described in Section 2, were performed by three researchers. For this purpose an annotation guideline based on the mentioned taxonom was utilized. Each researcher annotated 33.33% percent of the whole corpus, while we randomly selected 10.00% of each annotator and assigned it additionally to all the other two annotators. As a result, each annotator annotated 36.66% of the complete corpus and 9.99% were annotated three times. Table 3 reports on the numbers which are relevant for the inter-annotator agreement for this threefold annotated part of the corpus.

Annotator *A* is a data science student with three years of experience in NLP applied to the legal domain. Annotator *B* is a PhD student in NLP with research focus on the application of legal court rulings. The third annotator *C* is a legal expert working with a working experience of 10 years. For this work, we report the inter-annotator agreement by means of the

[1]See https://www.gesetze-bayern.de/

Table 1: Some special examples from the sentence taxonomy.

Type		Examples
Identical sentence start		§55a, §79, §104, §126b, §199(1), §199(3), §212, §271a(2), §286
Variable sentence middle		§207(1), §312h, §350, §438, §505a, §575, §595(1)
Enumerations	Standard	§58, §75, §81, §101, §197, §204, §207, §271a(5)
	Sentences	§98, §310, §502(3), §576, §569, §651h(4), §1059a
Definitions		§308, §309
List of lists		§204, §207, §305a, §308, §309, §312, §356
Footnotes & Official notes		§14, §241a, §247, §271a, §275, §286, §288, §304, §308, §310

Table 2: Statistics on the dataset created per document type.

Document type	# Documents	# Sentences	# Tokens
Laws	13	20,322	498,403
Judgments	131	21,484	518,052
Terms of Service	100	8,297	175,529
Privacy Policies	11	2,186	53,552
Total	255	52,289	1,245,536

Table 3: Inter-Annotator agreement for the sentence boundaries.

	Boundary	No Boundary	Total Tokens
A	5,229	119,325	124,554
B	5,104	119,450	124,554
C	4,818	119,736	124,554

Kappa coefficient (Cohen, 1960). As we had three human annotators, we utilized the Fleiss Kappa coefficient (Fleiss, 1971). Based on the annotations shown in Table 3, $p_{boundary}$ is 0.0405 and $p_{no\_boundary}$ is 0.9594. Therefore, $\overline{P}_e$ equals to 0.9222. In a next step, for each of the 124,554 tokens the respective P_{token} was calculated. As only two possibilities are possible for each token (either all three annotators agreed, or one annotator had a different perception) this value is either 1.0 or 0.33. A discrepancy was found in 411 cases and thus the sum of P_i is 124,278.63 resulting in a $\overline{P}$ of 0.9978. This numbers were used to calculate a Fleiss Kappa k of 0.9716.

Hence, the created corpus can be seen as very reliable. Additionaly, a manual inspection of the discrepancies has shown that the legal expert ocassionally did not annotate tokens such as headlines, which represent a sentence boundary according to our guideline. This fact is also suggested by the overall lower number of annotated boundaries by that annotator as shown in Table 3.

Furthermore, sometimes typographical errors were encountered and repaired to the extent possible. In order to allow better processing later on, structural information linked to a sentences is kept with the sentence, e.g. the clause number in a paragraph is always part of the first sentences of that clause. Thus, it is possible to reconstruct the original text if desired.

4 EXPERIMENTAL SETUP

SBD is a common pre-processing step which can be found in almost any NLP framework. The tested, existing methods in this paper are chosen based on the reported performance by Read et al. (Read et al., 2012). *OpenNLP*[2] had the best overall performance on the combined dataset without any preprocessing. The sentence segmentation module from *NLTK* (Loper and Bird, 2002) is the only unsupervised approach with a performance levels comparable to the supervised approaches and a high robustness. Unfortunately the best reported method on all individual datasets tokenizer could not be found and consequently not tested, but the reported performance difference is only marginal. Hence, it was not necessary to re-implement this method manually. The beforementioned methods are used as a baseline. We have to stress that most older methods focus on sentence boundaries found on sentence terminating characters, thus especially NTLK is at a disadvantage in the legal domain.

For the method chosen in this paper an aggressive tokenizer is used, also utilized in other research (Savelka and Ashley, 2017b). Words, numbers and special characters are separated from one another. For example, abbreviations are split from their terminating dot, thus making no previous, maybe misleading

[2]See https://opennlp.apache.org/

assumptions. In this processing step, all escape sequences except the newline sequence is removed because it might indicate the boundary of certain textual structures such as headlines. These decisions are motivated by the many ways legal documents can be structured or written in. In most cases it is not possible to determine every possible writing style and combination of special characters in advance.

4.1 Conditional Random Fields

As conditional random fields (CRF) (Lafferty et al., 2001) are commonly used for sequence modelling, i.e. assign labels for the items in a possibly correlated input sequence, this work investigates the applicability of such models for sentence boundary detection in legal documents. The list of tokens provided by the tokenizer is translated into a list of features for each individual token by means of feature extractors. Hereby, the surrounding tokens and their features are also taken into consideration.

The features for a token are a combination of their own properties and the specific properties of the tokens in the predefined window around the word or token. The size of the window is variable for each used feature extractor. Table 4 shows all extractors and features utilized.

Many different combinations of features were tried. Therefore, based on the average F_1 score for a single feature extractor, the respective word window and feature was selected for further testing. This iterative procedure revealed that the most important features were (1) whether the token was a special character, (2) the token in lowercase variation, (3) the length of the token, (4) the token signature, (5) whether the token starts with an uppercase letter or (6) lowercase character, and (7) whether the token is a number. These respective feature extractors were then used to train the final models.

4.2 Neural Networks

We trained different neural networks, inspired from related work, to predict the sentence boundary probability. Here we report on the best performing network. As an input it receives the word embeddings of the possible boundary and its context window, which are the 7 preceding and subsequent token. If the considered token is at the beginning or end of a text, the context window is padded with zero vectors to get a uniform size. A symmetric window of token is used as the input for the whole neural network with the possible sentence boundary in the middle. If a word is not in the embedding dictionary, its word vector is set to zero. In this case, averaging the embeddings of the context around the word to estimate the vector is also possible, but the neural network already can access the whole context around the word. Word2vec (Mikolov et al., 2013) vectors are used in the paper because of their simplicity. *gensim* (Řehůřek and Sojka, 2010) is used to train the embeddings with a window size of 5, a vector size of 100 and a minimal word occurrence of 2. The whole dataset is used to train the word embeddings.

The neural network is implemented in *PyTorch*[3] and can be seen in Figure 1. It consist of four bidirectional layers of Gated Recurrent Units (GRU, simplified in Figure 1 to one layer), which are followed by two linear layers. The GRU uses its standard nonlinear activation function Tanh and between the linear layers there is one ReLU. The sigmoid function generates the wanted probability. The hidden dimension size of the bidirectional GRU is 64 and its output for each token is concatenated (two hidden states per token), resulting in a 1920 dimensional vector. This vector is then reduced to 64 dimensions in the first linear layer.

The Adam algorithm is used for optimization. The learning rate of $2.5e^{-4}$ is gradually decreased every five epochs by half, while the network was trained for 60 iterations or epochs over the whole training dataset.

5 RESULTS

The main focus of this evaluation is put on the law and judgment collections. The other collections are used as a reference value because they do not consist of enough sentences to get reliable results for training and testing. After evaluating each method on those two collections, their performance is calculated on the other previously unseen datasets.

To evaluate the neural network and to avoid overfitting, the laws and judgments are each split into the three parts: train, validation and test dataset. The validation and test dataset each consist of 10% of the texts. Each different model is trained on the train dataset and after every epoch evaluated on the validation dataset. The model with the highest F1 score is saved and at the end of the training tested against the test dataset. For the dataset the model was trained on, the test score at the end of the training is reported.

[3]See https://pytorch.org/

Table 4: The features used as input for the CRF.

Feature Extractor	Output
Special	Categorizes the different special characters. If a token is not a special character, the output is ”No”, otherwise ”End” stands for sentence-terminating characters, ”Open” for opening parentheses, ”Close” for closing parentheses and ”Newline” for newline characters. In all other cases, such as the paragraph sign, the output is ”S”.
Lowercase	Extracts the individual tokens in lowercase. This way, it is possible for the system to learn certain trigger tokens, which denote the end of a sentence.
Length	Extracts the length of every token in the window. Also used to distinguish between possible abbreviations.
Signature	The signature of each token is extracted, which is as a feature a combination of the length and the type of the token, e.g. numbers, words and special characters. The token is then translated into the specific combination of numbers as ”N”, lowercase characters as ”c”, Uppercase characters as ”C” and special characters as ”S”.
Upper	Extractor which outputs the truth value if the token starts with an uppercase letter to distinguish abbreviations from sentence ends. If the token after the dot is uppercase, this is a strong indicator for a sentence end.
Lower	Same idea as the Upper-extractor, but in this case the output is ”True” for all tokens that start with a lowercase letter.
Number	Extracts wheter the token is a number. Also useful to determine whether the token can be found on a headline or not.

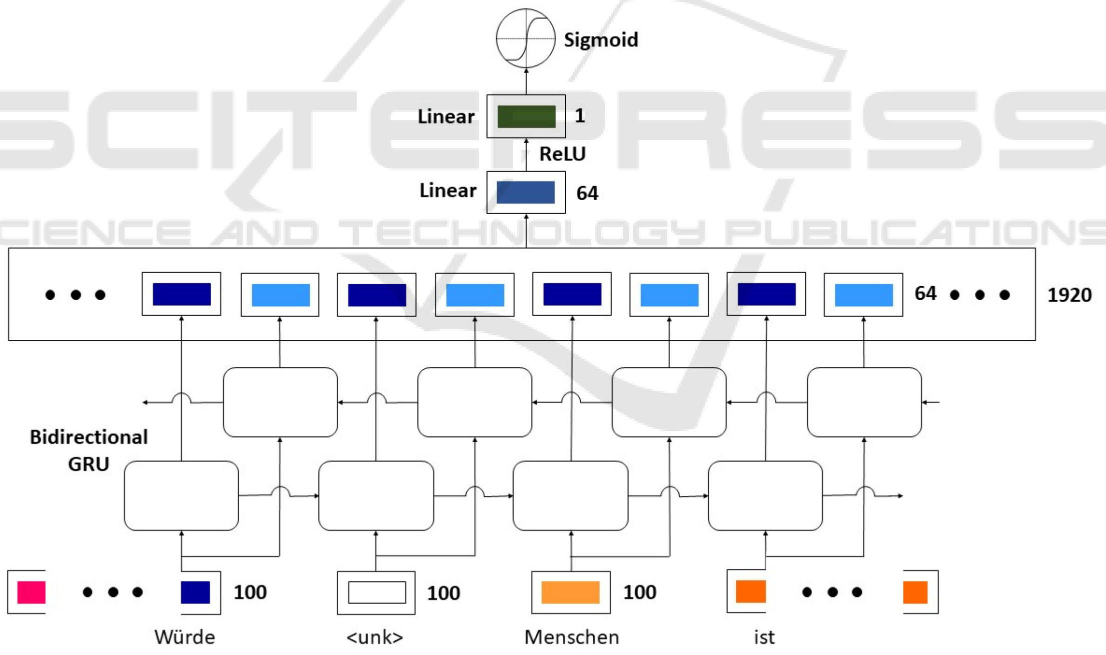

Figure 1: The neural network architecture used in this paper. A bidirectional GRU followed by a concatenation and two linear layers.

5.1 Evaluation

The already existing approaches *OpenNLP* and *NLTK* have a low performance, with the *OpenNLP* model trained on judgments being slightly better (see Table 7). Counter-intuitively, the *OpenNLP* model trained on judgments outperforms the model trained on laws when evaluated on laws. A reason for this could be the higher complexity of the law texts, indicated by the higher F1 scores on the judgments for almost all methods (comparing the results for each method specifically trained for that dataset). In this case, *OpenNLP* might not be capable of modelling the laws adequately, but can generalize the concept of

Table 7: Performance evaluation for *NLTK* and *OpenNLP* on laws and judgments.

	Laws			Judgments		
	F1	Rec	Pre	F1	Rec	Pre
NLTK	73.0	61.2	92.1	69.9	66.4	73.9
OpenNLP *Laws*	72.9	61.1	90.4	64.4	64.5	64.4
OpenNLP *Judgments*	74.8	60.3	98.4	78.1	65.6	96.3

sentence boundaries well enough from the judgments.

When visually examining the results obtained from the existing methods it is evident that most errors are linked to textual structures such as citations, headlines and lists. These methods shall be used as the baseline, although it is advised not to use them in the German legal domain.

First, we evaluated the performance of our CRFs. While we tested many different feature combinations (see Section 4.1), we encountered that widening the scope of a feature extractor leads to better results, but increasing the window over a certain threshold often degrades the performance of that extractor. Using more lowercased words as a feature increases the model performance, but the computational and memory cost is to high to further increase the viewed scope. Table 5 reports on the results of the best CRFs trained on laws and judgments.

The best CRF trained on laws uses the following features: *Special* with a window of 10, *Lowercase* and *Length* with a window of 7, *Signature* with a window of 5 and *Lower*, *Upper* and *Number* with a window of 3.

Table 5: Performance of the best CRF on laws and judgments.

	Laws			Laws		
	F1	Rec	Pre	F1	Rec	Pre
CRF *Law*	**95.4**	**94.4**	**96.4**	77.3	68.4	89.0
CRF *Jug*	81.2	76.8	86.0	**96.8**	**96.6**	**97.0**

The best CRF for the judgments has following extractors: *Length*, *Signature*, *Lowercase*, *Special* with a window of 5 and *Lower*, *Upper* and *Number* with a window of 3.

Particularly the inclusion of special tokens is important for a good performance on the law documents. The assumption here is that they are essential for a better understanding about the overall structure. For training on judgments, a smaller scope for each extractor yields the best performance.

Overall the performance of CRF increases significantly with the inclusion of the *Lowercase* extractor. This extractor was introduced by (Savelka and Ashley, 2017a), but they used a smaller scope. A similar model to the one introduced in their paper is also

Table 6: Performance of the CRFs on terms of service and privacy policies.

	ToS			Privacy Pol.		
	F1	Rec	Pre	F1	Rec	Pre
CRF *Law*	**91.8**	**86.4**	**95.7**	81.7	81.0	82.3
CRF *Jug*	81.1	82.4	79.8	**84.5**	**81.6**	**87.6**

tested but produces worse results with an F_1 score of 95.0% for laws and 96.0% for judgments.

The evaluation for the terms of service and privacy policies showed a challenge already seen in (Read et al., 2012). The performance of the SBD system significantly decreases when processing unknown documents, see Table 6 . The F_1 score for the privacy policies nearly reaches the baseline value from the OpenNLP with 84.5%.

CRF are a versatile model, but their performance levels are bound to the creativity while engineering the features. As seen during testing, changes to features often only have a marginal impact on the overall performance. Additionally, there are only a limited number of parameters possible. This is not the case with a RNN, which will be evaluated next.

As a result, we evaluated the performance of the created neural network. For this section we trained models on three different parts of the dataset, one only on the judgments (Table 8), one for the laws (Table 9), and one using both as training input (Table 10). For each such part five randomly-initialized models were trained and their performance averaged. The maximum deviation from the average is also reported.

The best performance of each document type is marked. As expected, the model trained on a document type also yielded the best performance on that specific dataset part. There is a performance decline for the PRVs noticeable, which can at least be partly explained by their more casual writing style. With an increase of up to 22 points in the F1 score, a significant difference can be seen to the existing methods. We can report performance levels comparable to the state-of-the-art scores reported by Read et al. (Read et al., 2012) on the WSJ, the Brown corpus, etc., although it can be argued that the task of finding a sentence boundary potentially anywhere in a text is significantly harder.

In order to evaluate how our neural network

Table 8: Performance of the neural network only trained on judgments. Averaged over five models with maximum deviation.

	F1	Recall	Precision
Law	87.80 ± 0.60	84.38 ± 1.90	91.59 ± 0.77
Jug	**98.51** ± 0.19	**98.80** ± 0.33	**98.23** ± 0.42
ToS	92.88 ± 0.90	88.25 ± 1.61	98.02 ± 0.49
Prv	81.32 ± 0.90	77.02 ± 1.43	86.13 ± 0.34

Table 9: Performance of the neural network only trained on laws. Averaged over five models with maximum deviation.

	F1	Recall	Precision
Law	**97.46** ± 0.18	**98.04** ± 0.77	**96.89** ± 0.47
Jug	83.77 ± 2.04	77.59 ± 3.56	91.11 ± 1.64
ToS	91.92 ± 0.32	90.56 ± 1.13	93.35 ± 1.22
Prv	82.55 ± 0.53	**83.49** ± 1.12	81.63 ± 0.68

performs on normal text the TIGER corpus was used (Brants et al., 2004) as a source for normal German sentences. This corpus is already segmented into sentences and in order to create a real-life setting we introduce a simple formatting rule of only allowing a maximum of eight alphanumeric tokens or words per line.

In Table 11, we tested each of the models from above which were trained on the legal documents on the TIGER corpus. Only focusing on legal documents introduces performance issues with normal texts. In order to overcome this problem, we also trained our neural network on the whole dataset additionally with the around 50,000 sentences from the TIGER corpus. To test the effect of these added sentences, one half of the test and validation data was taken form the TIGER corpus and the other from our legal dataset. As seen in Table 12, we can achieve a high performance for both corpora with a non-legal F1 score of almost 99%.

5.2 Error Analysis

A manual analysis of the errors from the legal documents was performed by Annotator *B*. The analysis showed that 22.6% are wrongly annotated tokens, which nonetheless can be interpreted as correctly classified by the neural network. These are either tokens, which were obviously overlooked by the annotator, or unclear examples, where a decision could have been made in both directions. As a result, the performance of our models can be even seen as slightly better than actually reported. The remaining errors are distributed as follows: 37.7% are related to lists or enumerations, 13.1% are numbers or names and only 6.5% are abbreviations. The remaining 21.1% are other errors which do not have a direct link.

A large portion of the errors are found within or around lists and enumerations. When looking at these examples, it is sometimes even for a human not possible to determine whether there is a sentence boundary or not based on the given context. Classifying sentence boundaries in such textual structures is sometimes only possible in a more global way, looking at the list as a whole. Mixed cases and the lists spanning over many lines further complicates this. In order to at least partly solve this problem, we propose to increase the size of the window.

The problem with the numbers and names, is that they most of the time do not have an embedding. By using character embeddings we could give the neural network information, even about words that it has never seen before. In the following paragraphs we want to evaluate the influence of zero embeddings on the performance of the system.

In this paper, the strategy to deal with unknown words or tokens was to insert a zero vector as a placeholder. The neural network should then be able to infer some meaning about that particular token. We tested this assumption by extracting all the errors the neural network produced when evaluated on the legal sentences and Figure 2 a contradiction with this assumption. A far larger portion of error is encountered, if at least one token is unknown. And if we encounter a context window with many missing tokens, the probability of making an error is much higher. When we have six or seven missing tokens this does not hold true. This is simply the case, because we are padding the context window at the beginning and end of the text. 31 out of these 33 cases are found at the beginning or the end of a text. With 255 documents, we have over 500 such context windows which heavily influence the probability.

To summarize this error analysis: We looked at two error factors for sentence boundaries. On the one hand the writing style of legal documents with their long, complex and nested textual structures introduces errors. On the other hand missing statistical information for unknown tokens at least influences a large part of the errors produced by the neural network.

6 CONCLUSION & OUTLOOK

In this paper, we firstly introduced the problem of SBD in the German legal domain. Furthermore, we created and published an annotated dataset consist-

Table 10: Performance of the neural network trained on the judgments and laws. Averaged over five models with maximum deviation.

	F1	Recall	Precision
Law&Jug	97.82 ± 0.14	98.33 ± 0.63	97.31 ± 0.40
ToS	**95.00** ± 0.98	**91.78** ± 1.98	**98.46** ± 0.33
Prv	**83.35** ± 0.30	81.90 ± 1.06	**84.84** ± 0.49

Table 11: Performance of the models tested on the TIGER corpus. Averaged over five models.

	F1	Recall	Precision
Jug	88.14 ± 0.41	84.80 ± 1.18	**91.75** ± 0.51
Law	87.97 ± 0.23	**88.25** ± 0.68	87.71 ± 0.87
Law&Jug	**88.81** ± 0.29	86.93 ± 0.70	90.79 ± 0.74

Table 12: Performance of the models tested on the TIGER and Legal Corpus. Averaged over 5 models.

	F1	Recall	Precision
Total	98.21 ± 0.11	98.36 ± 0.11	98.05 ± 0.33
Legal	97.48 ± 0.14	97.95 ± 0.28	97.01 ± 0.52
Non-Legal	98.93 ± 0.09	98.77 ± 0.12	99.09 ± 0.15

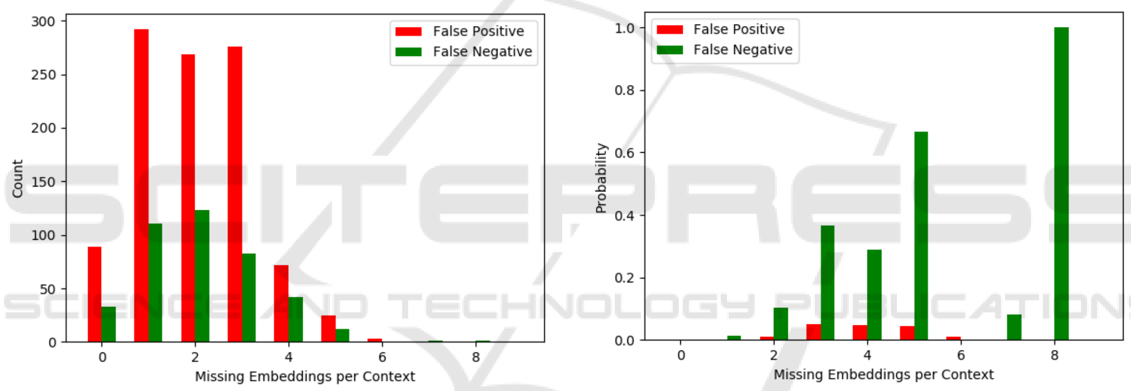

Figure 2: The influence of missing/zero embeddings on the errors.

ing of various types of German legal documents[4]. This dataset was then used to train and evaluate newly developed methods of SBD and to compare these against existing methods. However, this dataset is now publicly available and can be used within the community to perform further research on SBD within the legal domain.

Our proposed models achieved state-of-the art performance with regard to SBD. Hence, this research can be utilized by other researchers to perform more complex NLP tasks within the legal domain.

Even though we could show that our models perform on a very high level and that they therefore could be utilized in NLP pipelines for various legal tasks, some limitations exist. We only used simple word embeddings (Word2Vec (Mikolov et al., 2013)) and trained them on our corpus. For future work, it could be even more beneficial for the models, to utilize state-of-the art sequence-to-sequence language models such as BERT (Devlin et al., 2018).

[4]The corpus and codebase is available at the following repository: https://github.com/sebischair/Legal-Sentence-Boundary-Detection

REFERENCES

Ashley, K. D. (2017). *Artificial intelligence and legal analytics: new tools for law practice in the digital age*. Cambridge University Press.

Brants, S., Dipper, S., Eisenberg, P., Hansen, S., König, E., Lezius, W., Rohrer, C., Smith, G., and Uszkoreit, H. (2004). TIGER: Linguistic linguistic interpretation of a german corpus. In *Journal of Language and Computation*, 2, pages 597–620.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

de Maat, E. (2012). *Making sense of legal texts*. PhD thesis, University of Amsterdam.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Koller, D. (2019). Interactive analysis of a corpus of general terms and conditions for variability modeling. Master's thesis, Technical University of Munich, Munich, Germany.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Loper, E. and Bird, S. (2002). NLTK: The natural language toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, Philadelphia. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Moens, M.-F. (2001). Innovative techniques for legal text retrieval. *Artificial Intelligence and Law*, 9(1):29–57.

Read, J., Dridan, R., Oepen, S., and Solberg, L. J. (2012). Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, Mumbai, India. The COLING 2012 Organizing Committee.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Savelka, J. and Ashley, K. D. (2017a). Sentence boundary detection in adjudicatory decisions in the united states. *Traitement automatique des langues*, 58(February):21–45.

Savelka, J. and Ashley, K. D. (2017b). Using conditional random fields to detect different functional types of content in decisions of united states courts with example application to sentence boundary detection. In *Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts*, London, Great Britain. ASAIL.

Wyner, A. Z. (2010). Towards annotating and extracting textual legal case elements. *Informatica e Diritto: special issue on legal ontologies and artificial intelligent techniques*, 19(1-2):9–18.

Wyner, A. Z. and Peters, W. (2011). On rule extraction from regulations. In *JURIX*, volume 11, pages 113–122.