Classifying Semantic Types of Legal Sentences: Portability of Machine Learning Models

Ingo GLASER [a], Elena SCEPANKOVA [a], and Florian MATTHES [a]
[a] *Software Engineering for Business Information Systems, Department of Informatics, Technical University of Munich, Germany*

Abstract. Legal contract analysis is an important research area. The classification of clauses or sentences enables valuable insights such as the extraction of rights and obligations. However, datasets consisting of contracts are quite rare, particularly regarding German language.

Therefore this paper experiments the portability of machine learning (ML) models with regard to different document types. We trained different ML classifiers on the tenancy law of the German Civil Code (BGB) to apply the resulting models on a set of rental agreements afterwards. The performance of our models varies on the contract set. Some models perform significantly worse, while certain settings reveal a portability. Additionally, we trained and evaluated the same classifiers on a dataset consisting solely of contracts, to be able to observe a reference performance. We could show that the performance of ML models may depend on the document type used for training, while certain setups result in portable models.

Keywords. legal sentence classification, portability of machine learning models, natural language processing, text mining

1. Introduction

Nowadays, many sectors face the obstacle called digitalization. So, does the legal domain as well. The rising of legal technology is highlighted by the increasing number of digitized legal documents, in particular legal contracts [1]. Due to the vast progress in research with regard to natural language processing (NLP), text mining is becoming more powerful in terms of its accuracy and performance. The tools and use cases for text mining in the legal field that are relevant for legal experts or practitioners, e.g., scientists, lawyers, judges, courts, etc., are diverse [2].

The computer-aided analysis of legal contracts is an important research are. Companies, law firms, government agencies, but also private individuals need to monitor contracts for a wide range of tasks [3]. For example, law firms and legal departments need to process large numbers of contracts to monitor the compliance. In terms of B2C, individuals are always involved as a contractual party. Therefore they need to understand their rights and obligations within that business relationship. However, the complex legal language hampers this understanding [4]. Thus we consider the task of extracting different legal concepts out of contracts.

In the legal domain however, the issue of data scarcity exists. This applies particularly to the German legal domain with regard to contracts, due to the nature of such documents, as they carry big privacy concerns. In this paper we want to investigate on the portability of machine learning (ML) models. Usually it is differentiated between two types of portability. Domain portability describes the capability of a ML model to perform its designated task on a different domain than it was originally trained on. In contrast, it can be also distinguished between different kind of documents. This paper focuses on the latter type of portability in order to overcome the lack of legal training data.

The reminder of the paper is structured as follows: Section 2 provides a short overview of the related work, Section 3 describes the semantic types leveraged, the experimental setup along with the used datasets are discussed in Section 4, finally the approaches and its performance is evaluated in Section 5, before Section 6 closes with a conclusion and outlook.

2. Related Work

The computer-assisted analysis of text phrases in legal documents with regard to their semantic type is highly relevant and has attracted researchers for quite some time. However, hardly any attempt has been made in the German contract domain. Waltl et al. [5] introduced a semantic type taxonomy for the German civil law, which was used to classify norms of German statutory texts. They used rule-based approaches as well as ML for the classification. In a previous work, Waltl et al. [6] incorporated active learning (AL) into that process in order to overcome the problem of data scarcity.

Approaches to classify legal norms with ML however exist in different jurisdictions. An early contribution to the classification of norms in legislative texts using ML approaches was made by Biagioli et al. [7] in 2005. The authors distinguish between 11 different semantic types, which are assigned on a norm level, achieving an average F_1 measure of 0.80. The same 11 functional classes were used by Francesconi and Passerin [8] to evaluate a multinomial naive bayes (MNB) classifier as well as a support vector machine (SVM). Maat et al. [9] classified legal norms within the Dutch legislation, using a taxonomy of 13 different classes. Utilizing SVMs, they could achieve an accuracy of more than 90%. O'Neill et al. [4] achieved an accuracy of 82% on the task of classifying sentences within different financial regulations leveraging deep learning (DL).

Research on classifying text phrases within contracts has been made in different domains. Indukuri and Krishna [10] applied a SVM on contract clauses to detect whether a clause is concerned with payment terms or not. Chalkidis et al. [11] tried to extract obligations and prohibitions out of Englisch contracts using hierarchical recurrent neural networks (RNN). While there is other work existing on the application of NLP for the information extraction (IE) of legal contracts [12,13,14,15], hardly any other work on classifying contract clauses with regard to their semantic types exists.

To the best of our knowledge no attempt to classify sentences for German legal contracts using ML approaches has ever been made before. Furthermore, we are not aware of any work related to the portability of ML models between statutory texts and legal contracts, even though portability of ML models as such, often referred to as transfer learning, is an examined research area.

3. Semantic Types of German Legal Sentences

The classification of legal sentences can be addressed from different perspectives, such as legal theoretical, philosophical, or a constructive one. In order to capture the semantics of sentences in legal documents, a functional classification approach seems to be most suitable. For the legal domain, different functional type systems were already introduced in other works as well as by us, e.g. [5,6,9]. We looked at existing classifications but also tried to leverage them, to come up with other possible taxonomies.

Waltl et al. [6] came up with a legal theoretically funded taxonomy of legal norms for German statutes. This classification system distinguishes at the first granularity between normative, auxiliary, legal-technical, and legal-mechanism statements (a deeper description of the taxonomy can be found in [6]). Due to the fact that this work investigates in the document type portability of ML models, a classification system appropriate to the legal domain, rather than special to a document type such as statutes, is required. Furthermore, this taxonomy constitutes 21 types at the finest fidelity. Section 4.2 introduces the datasets used during this research and reveals limited datasets in terms of size. Hence, a classification into 21 different types would cause a very low support for various classes, which is not a good setting for a supervised ML approach. Therefore we came up with three different taxonomies. The first system distinguishes between rights and obligations, including an additional fall-back class. Secondly, a taxonomy consisting of obligations, rights, references, definitions, legal consequences, and objections is proposed. A third taxonomy is shown in Table 2. We evaluated the different taxonomies from a legally theoretical perspective as well as from a technical one. For the technical analysis we annotated a dataset constituting the tenancy law of the German Civil Code (BGB) sentence-by-sentence with each of the taxonomies. We then trained two linear classifiers (SVM and logistic regression (LR)), as well as a decision tree (extra tree classifier (ETC)) using term-frequency (TF) on each dataset. Table 1 shows the results.

Classifier	F_1		
	3 Classes	6 Classes	9 Classes
SVM + TF	0.796	0.875	0.828
LR + TF	0.787	0.848	0.807
ETC + TF	0.817	0.874	0.831

Table 1. Technical analysis of the three taxonomies

The taxonomy consisting of three classes performed the worst. Moreover, the differentiation assumed by it is not sufficient for our purposes. While the taxonomy by Waltl et al. [5] seems to be more robust from a legal perspective, the six classes are from a technical point of view superior to the former. However, when applying NLP to the legal domain, we already learned, that domain knowledge is crucial and should not be neglected by technical assumptions. Hence the taxonomy described in Table 2 is used for this work. A detailed description of the thoughts behind this taxonomy can be found in [5].

4. Experimental Setup

4.1. Objective

Classifying sentences in legal contracts is, due to several reasons, attractive for the field of legal informatics. In the first place, it allows a more elaborate differentiation of a sen-

Semantic Type	Description
Duty	The primary function of a duty is to stipulate actions, inactions or states
Indemnity	The primary function of an indemnity is to clarify that, resp. under which conditions a duty does not exist
Permission	The primary function of a permission is to authorize actions, inactions or states
Prohibition	The primary function of a prohibition is to forbid or disallow actions, inactions or states
Objection	The primary function of an objection is to define that, resp. under which circumstances an existent claim may not be asserted
Continuation	The primary function of a continuation is to extend or limit the scope of application of a precedent legal statement
Consequence	The primary function of a consequence is to stipulate legal effects, without ordering or allowing character as far as the legal consequence part is concerned
Definition	The primary function of a definition is to describe and clarify the meaning of a term within the law
Reference	The primary function of a reference is to cite another norm with the aim of total or partial application transfer or non-application

Table 2. The used taxonomy created by us in an earlier work[5]

tence's meaning and thus enables subsequent contract analysis. Secondly, it is beneficial for information retrieval (IR) tasks in legal information databases and consequently supports the efficiency of e-discovery concerning legal documents. Last but not least, it helps determining dependencies and references between contracts and statutory documents.

Due to the lack of (annotated) data in this field, we are investigating on the portability of ML models. More precisely, the goal is to leverage the existing amount of statutory texts in order to create ML models which can be used to classify contract clauses or sentences.

4.2. Data

In order to prepare a proper setup for the legal sentence classification experiment, three different datasets were used. The first dataset compromises 601 sentences which constitute the tenancy law of the German Civil Code (§535-§597) in its consolidated version, effective from 21st of February 2017. Secondly, a dataset consisting of 169 sentences from rental agreements was required in order to test the trained ML models. Furthermore, this dataset was extended to 312 sentences, so that it could be used for training as well.

In a first preprocessing step, the raw text of these articles and clauses was segmented into sentences. Sentence segmentation in the legal domain can be a challenging task and results in a performance, yet inferior to other more common domains [16]. For this work a straight-forward rule-based approach by Waltl et al. [5] was chosen.

NLP typically involves various further pre-processing steps. Due to the fact that this work is based on a sentence classification problem and subject to be solved via supervised ML, the sentence segmentation was performed even before the actual NLP pipelines take place. Different normalization steps were incorporated into the varying pipelines and are discussed in Section 4.3.

Finally, the 913 sentences were manually classified by two human legal experts, according to the taxonomy described in Section 3. The annotations were performed using Gloss, the web-based annotation environment developed by Jaromir Savelka at the University of Pittsburgh. In this process, a third legal expert acted as the editor in order to decide on an annotation in the case of disagreement between the first two experts. The distribution of the different semantic types is revealed in Table 3. While some types have

a very low occurrence, e.g. *Indemnity*, *Prohibition*, or *Definition*, some occur regularly, e.g. *Duty*, *Permission*, or *Consequence*.

Semantic type	BGB (n=601)		Rental agreements (n=169)		Rental agreements (n=312)	
	#	Relative (%)	#	Relative (%)	#	Relative (%)
Duty	117	19.5	52	30.8	105	33.7
Indemnity	8	1.3	0	0.0	1	0.3
Permission	148	24.6	35	20.7	75	24.0
Prohibition	18	3.0	2	1.2	3	1.0
Objection	98	16.3	8	4.7	4	4.5
Continuation	21	3.5	7	4.1	12	3.8
Consequence	117	19.5	64	37.9	101	32.4
Definition	18	3.0	1	0.6	1	0.3
Reference	56	9.3	0	0	0	0
	$\sum$ 601	100	$\sum$ 169	100	$\sum$ 312	100

Table 3. Occurrence of the different semantic types in each of the datasets

4.3. Experiment

In order to measure the portability of ML models between document types, our experimental setting constituted four steps:

1. *Original Training:* We trained various classifiers on the BGB dataset and evaluated them using 10-fold cross-validation with 20% for testing.
2. *Portability Testing:* The resulting models were applied on the small set of rental agreements (n=169).
3. *Contract Training:* We extended the small set of rental agreements to 312 sentences and used it to train again various classifiers. The resulting models were evaluated on the same dataset using 10-fold cross-validation on 20% of the data.
4. *Portability Evaluation:* We applied the models from the original training on the new contract dataset (n=312). The results were compared with the performance from the contract training in order to assess the true portability.

Hereby, the classification of legal sentences using supervised ML was implemented following a basic workflow consisting of the following steps:

Data Acquisition: The data described in Section 4.2 was used[1].

Pre-Processing: We used three different pre-processing procedures: (1) The normalization (PRE) consisted of the removal of line breaks as well as duplicated whitespaces, replacing German umlauts, spelling numbers, and removing punctuation. (2) Stop word removal (SWR) was performed according to the spaCy[2] stop word list. (3) A lemmatization (Lemma) was conducted leveraging spaCy[2]. These three procedures were incorporated into pipelines in different combinations. Section 5.1 discusses the different variations.

[1] Available at: https://github.com/sebischair/Legal-Sentence-Classification-Datasets-and-Models
[2] https://spacy.io/usage/linguistic-features

Feature Extraction: Two different feature representations were used: (1) A bag-of-words approach was used to represent features. We used simple word count vectors and term frequency-inverse document frequency (TFIDF) transformer on these vectors. Where indicated, part-of-speech (POS) tags have been created and used as well. In order to keep the bag-of-words approach in this case as well, each token was combined with the respective POS tag using a dash. (2) The second feature representation leveraged word embeddings. We trained word2vec models on different legal corpora as well as used pre-trained models. These models were used to calculate the mean embedding of a sentence.

Training of Machine Learning Model: Six different classifiers were applied on the task of predicting the semantic types of legal sentences. We used MNB, LR, SVMs, multilayer perceptrons (P), random forests (RF), and a ETC. The models were trained using a 10-fold cross-validation on 80% of the dataset each iteration.

Evaluation and Error Analysis: Weighted variants of precision, recall, and F_1 were used to evaluate the performance of the trained models.

5. Evaluation and Error Analysis

5.1. Evaluating the portability

The objective of this experiment was to evaluate the portability of ML models with regard to different document types.

To achieve this, different classifiers were incorporated into various pipeline settings to train models on the BGB dataset. Waltl et al. [5] showed already that for simple pipelines, SVMs perform best on this dataset. For that reason, we initially trained the six classifiers by just relying on a simple count vectorizer (CV) as well as on TFIDF. Table 4 shows the performance of the models.

Classifier	Features	Precision	Recall	F_1
ETC	CV	0.814	0.836	**0.815**
	TFIDF	0.788	0.803	0.783
LR	CV	0.810	0.823	0.808
	TFIDF	0.724	0.749	0.719
MNB	CV	0.688	0.710	0.680
	TFIDF	0.699	0.646	0.616
P	CV	0.777	0.762	0.757
	TFIDF	0.798	0.780	0.777
RF	CV	0.728	0.718	0.705
	TFIDF	0.710	0.733	0.710
SVM	CV	0.838	0.839	**0.828**
	TFIDF	0.829	0.825	0.815

Table 4. Performance of the six classifiers on the BGB dataset

Two major observations can be made from this: (1) ETC and SVM perform the best, while (2) TFIDF creates worse results than simple TF. As a result, we tried several variations of combining different pipeline stages with these two classifiers based on TF. Furthermore we used sentence mean vectors by leveraging two pre-trained general word2vec

models[3,4] as well as two manually trained word2vec models[5]. The results of the training are shown in Table 5.

ML classifier	Pipeline	Precision	Recall	F_1
ETC	CV	0.814	0.836	0.815
	PRE + CV	0.816	0.838	0.818
	PRE + SWR + CV	0.731	0.735	0.720
	PRE + SWR + Lemma + CV	0.701	0.698	0.688
	PRE + Lemma + CV	0.815	0.841	0.818
	Lemma + CV	0.810	0.831	0.809
	POS + CV	0.824	0.846	**0.827**
	POS + Lemma + CV	0.826	0.843	**0.825**
	SWR + POS + Lemma + CV	0.728	0.736	0.721
	word2vec JRCAquis	0.642	0.641	0.612
	word2vec Datev	0.661	0.652	0.623
	word2vec Google news	0.588	0.584	0.557
	word2vec Wikipedia	0.645	0.649	0.625
SVM	CV	0.838	0.839	**0.828**
	PRE + CV	0.838	0.834	**0.826**
	PRE + SWR + CV	0.748	0.743	0.735
	PRE + SWR + Lemma + CV	0.722	0.713	0.707
	PRE + Lemma + CV	0.830	0.823	0.816
	Lemma + CV	0.850	0.850	**0.839**
	POS + CV	0.831	0.831	0.823
	POS + Lemma + CV	0.839	0.839	**0.830**
	SWR + POS + Lemma + CV	0.703	0.702	0.694
	word2vec JRCAquis	0.687	0.716	0.691
	word2vec Datev	0.696	0.725	0.701
	word2vec Google news	0.622	0.636	0.614
	word2vec Wikipedia	0.680	0.666	0.658

Table 5. Performance of the best two classifier on the BGB dataset

At a first glance it is already obvious, that the bag-of-words approach outperforms the word embeddings by far. However, the word2vec models based on German legal corpora result in a greater F_1 than the pre-trained models. Our word2vec models were trained on two different corpora with default configuration, except dimensions is set to 300, window size to five and iterations to 10: (1) The JRCAcquis [17] with 33.686.085 token, and (2) a corpus consisting of judgments from the fiscal law constituting 114.091.840 token. The sizes are still pretty small for that matter. Therefore, further research is necessary to investigate in the suitability of word embeddings for such a classification task.

The highest F_1 was achieved by the following pipelines using a count vectorizer for feature extraction: (1) SVM using Lemma for pre-processing, (2) SVM using Lemma for pre-processing and the combination of the original POS tag along with the lemmatized token as features, (3) SVM without any pre-processing, (4) ETC without any pre-processing, but the combination of the token along with its POS tag as features, (5) SVM with PRE as pre-processing, and (6) ETC using Lemma as pre-processing and the combination of the lemmatized token along with its original POS tag as features. The F_1 varied from 0.839 to 0.825.

In the next step, the best resulting models were applied on the small contract dataset. Furthermore, the extended rental agreement dataset was used to train the six pipelines again, using a 10-fold cross-validation with 80% of the data. Afterwards, we took the

[3] https://devmount.github.io/GermanWordEmbeddings/

[4] https://code.google.com/archive/p/word2vec/

[5] Available at: https://github.com/sebischair/Legal-Sentence-Classification-Datasets-and-Models

six models from the original training and applied them on the bigger contract dataset as well. Table 6 shows the resulting F_1 measures.

Classifier	Pipeline	F_1			
		Train BGB	Test Rental (n=169)	Test Rental (n=312)	Train Rental (n=312)
ETC	POS + CV	**0.827**	**0.825**	0.720	0.713
	POS + Lemma + CV	0.825	0.789	0.709	0.718
SVM	CV	0.828	0.728	0.670	0.707
	PRE+ CV	0.826	0.705	0.667	0.682
	Lemma + CV	0.839	0.727	0.680	0.685
	POS + Lemma + CV	0.830	0.694	0.667	0.707

Table 6. Comparison of the performance on the BGB and contract dataset from the best classifiers

The column *Train BGB* captures the existing results from the first phase (see Table 5). The next column includes the results when applying the models from the BGB to the smaller contract dataset. A varying delta in F_1 from almost equal to zero (ETC with POS and CV) to 0.136 (SVM with POS + Lemma + CV) could be observed. This is already a first clue for a limited portability of ML models between document types in the legal domain. However, it is also obvious, that certain feature representations may be feasible for a portable model. Nonetheless, the actual performance of a classifier trained on contracts needs to be heeded yet. The last column (*Train Rental (n=312))* serves for this purpose. The models resulting in the training on the bigger contract dataset cease in an even worse performance between a F_1 of 0.718 and 0.682. These results were quite surprising. As a consequence, we conducted the original experiment phase (using all possible combinations of pre-processing, features and the six classifiers) again on the contract dataset. A F_1 measure of 0.734 however was the maximum, which is still significantly below the performance of the BGB models. Hence, we also applied the BGB models on the bigger contract dataset. The column *Test Rental (n=312))* reveals the respective results. The delta between the two models varies. The BGB model created with ETC and POS + CV performs better than the contract model. This model has already revealed a well suited portability in the previous step. For the remaining models however, the contract models outperformed the BGB model.

Our results reveal an evidence, that certain settings allow a portability between document types, even though portable models across-the-board are not given.

5.2. Error Analysis

To be able to better understand the classification process, but in particular the portability between document types, the worst portable configuration (SVM with POS + Lemma + CV) is examined in greater detail. Table 7 shows the performance of each class for the evaluation on the statutory dataset and the small contract dataset.

The resulting performance measures differ from the results shown in Table 6. This is because of different train test splits applied. While Table 6 is generated from a 10-fold cross-validation method, Table 7 is based on a static test split. The weighted mean mitigates the positive impact of small classes such as *Definition*, where only one instance is present in the test set, or even no instance as for *Reference*.

The results indicate that the portability issue of the models may be caused by the imbalance between the datasets. Particularly the types with a very low occurrence in

Type	BGB				Rental agreements (n=169)			
	Precision	Recall	F_1	#	Precision	Recall	F_1	#
Duty	0.783	0.857	0.818	21	0.693	0.736	0.684	53
Indemnity	1.000	1.000	1.000	1	0.250	1.000	0.400	1
Permission	0.919	0.810	0.861	42	0.944	0.829	0.883	41
Prohibition	0.333	0.500	0.400	2	0.000	0.000	0.000	1
Objection	0.800	0.923	0.857	13	0.454	0.714	0.556	7
Continuation	1.000	0.833	0.909	6	0.667	0.667	0.667	6
Consequence	0.731	0.950	0.826	20	0.660	0.525	0.585	59
Definition	1.000	0.769	0.870	3	0.000	0.000	0.000	1
Reference	1.000	0.769	0.870	13	0.000	0.000	0.000	0
Arith. mean (weigh.)	0.857	0.835	0.835	$\sum$ 120	0.704	0.675	0.682	$\sum$ 169

Table 7. Comparison of the performance by each type for the worst portable model

the contract dataset can have a huge impact on the results. Even though the language may vary between statutory texts and contracts, the characteristics of the different semantic types remain the same. In order to provide evidence for this hypotheses, we looked into the existing model (SVM with POS + Lemma + CV) and inspected the coefficients of each feature. The most important features for the class *Permission* are: (1) *können_verb*, (2) *berechtigen_verb*, and (3) *dürfen_verb*. For *Duties* the features with the highest weights are: (1) *muss_adj*, (2) *zu_part*, and (3) *verpflichten_verb*. Now it becomes also clear why SWR worsens the performance. Typical stop words such as *zu* depict important features for our models. This actually makes sense, since *zu* indicates the infinitive form and thus is crucial for determining the type of a sentence. Table 8 provides two examples per class from each corpora.

Type	Corpus	Sentence
Duty	BGB	Er hat die auf der Mietsache ruhenden Lasten **zu tragen**.
Duty	Contract	Insgesamt **zu zahlen** sind 2600 Euro.
Permission	BGB	Setzt der Mieter einen vertragswidrigen Gebrauch der Mietsache trotz einer Abmahnung des Vermieters fort, so **kann** dieser auf Unterlassung klagen.
Permission	Contract	Gegen Erstattung angemessener Kopier- und Portokosten **kann** der Mieter verlangen, dass ihm Kopien der Berechnungsunterlagen zugesandt werden

Table 8. Examples of Duties and Permission from both corpora

As one can see in Table 8, the most important features of the models are present in the examples. As a consequence, the models can properly represent these instances and thus classify them correctly. Looking at the most important features of the models, it seems obvious that using symbolic classification methods utilizing grammars or regular expressions may be more promising. However, Waltl et al. [5] examined such an approach already and could show the superiority of ML-based approaches.

6. Conclusion & Outlook

This work examined the portability of ML models with regard to different document types for the legal domain. Various classifiers were trained on the tenancy law of the German Civil Code and applied on a rental agreement dataset afterwards. Furthermore, the same settings were used to train models directly on the contractual dataset. We could show that ML models can be portable up to a certain degree in terms of document types.

Nonetheless, this research includes some limitations. The rental agreement dataset was pretty small for a supervised ML approach and differed in size in comparison to the

statutory dataset. Furthermore, the class distribution varied between the two datasets. As a consequence, future research needs to define an even more suitable setting in terms of data distribution and size in order to provide more evidence on the portability of ML models. Yet, this work builds a solid base for future research in this area.

Another promising approach may be the incorporation of word2vec. Even though we have used word2vec features, it was not our focus and thus we have not investigated in greater detail into the vast amount of options concerning the training of word2vec models as well as the feature representations utilizing word2vec. Due to the nature of word2vec, capturing the semantics of words, it may be feasible for such a semantic classification task.

Lust but not least, this work did not look into domain portability of ML models, which is indeed another interesting and potentially helpful research field.

References

[1] M. Saravanan, B. Ravindran, and S. Raman, "Improving legal information retrieval using an ontological framework," *Artificial Intelligence and Law*, vol. 17, no. 2, pp. 101–124, Jun. 2009.

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

[3] Z. Milosevic, S. Gibson, P. F. Linington, J. Cole, and S. Kulkarni, "On design and implementation of a contract monitoring facility," in *First IEEE International Workshop on on Electronic Contracts*. IEEE, 2004, pp. 62–70.

[4] J. O. Neill, P. Buitelaar, C. Robin, and L. O. Brien, "Classifying sentential modality in legal language: a use case in financial regulations, acts and directives," in *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*. ACM, 2017, pp. 159–168.

[5] B. Waltl, G. Bonczek, E. Scepankova, and F. Matthes, "Semantic types of legal norms in german laws: classification and analysis using local linear explanations," *Artificial Intelligence and Law*, Jul 2018.

[6] B. Waltl, J. Muhr, I. Glaser, G. Bonczek, E. Scepankova, and F. Matthes, "Classifying legal norms with active machine learning," *Legal Knowledge and Information Systems*, p. 11, 2017.

[7] C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni, and C. Soria, "Automatic semantics extraction in law documents," in *Proceedings of the 10th international conference on Artificial intelligence and law*. ACM, 2005, pp. 133–140.

[8] E. Francesconi and A. Passerini, "Automatic classification of provisions in legislative texts," *Artificial Intelligence and Law*, vol. 15, no. 1, pp. 1–17, 2007.

[9] E. de Maat, K. Krabben, and R. Winkels, "Machine learning versus knowledge based classification of legal texts." in *JURIX*, 2010, pp. 87–96.

[10] K. V. Indukuri and P. R. Krishna, "Mining e-contract documents to classify clauses," in *Proceedings of the Third Annual ACM Bangalore Conference*. ACM, 2010, p. 7.

[11] I. Chalkidis, I. Androutsopoulos, and A. Michos, "Obligation and prohibition extraction using hierarchical rnns," *arXiv preprint arXiv:1805.03871*, 2018.

[12] I. Chalkidis, l. Androutsopoulos, and A. Michos, "Extracting contract elements," in *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*. ACM, 2017, pp. 19–28.

[13] I. Chalkidis and I. Androutsopoulos, "A deep learning approach to contract element extraction." in *JURIX*, 2017, pp. 155–164.

[14] M. Curtotti and E. Mccreath, "Corpus based classification of text in australian contracts," in *Proceedings of the Australasian Language Technology Association Workshop*, 2010.

[15] I. Glaser, B. Waltl, and F. Matthes, "Named entity recognition, extraction, and linking in german legal contracts," in *Internationales Rechtsinformatik Symposium*, 2018.

[16] J. Savelka, V. R. Walker, M. Grabmair, and K. D. Ashley, "Sentence boundary detection in adjudicatory decisions in the united states," *Traitement automatique des langues*, vol. 58, no. 2, pp. 21–45, 2017.

[17] R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufis, and D. Varga, "The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages," *arXiv preprint cs/0609058*, 2006.