Legal Knowledge and Information Systems
A. Wyner and G. Casini (Eds.)

doi:10.3233/978-1-61499-838-9-11

Classifying Legal Norms with Active Machine Learning

Bernhard WALTL [a], Johannes MUHR [a], Ingo GLASER [a], Georg BONCZEK [a], Elena SCEPANKOVA [a], and Florian MATTHES [a]

[a] *Software Engineering for Business Information Systems, Department of Informatics, Technical University of Munich, Germany*

Abstract. This paper describes an extended machine learning approach to classify legal norms in German statutory texts. We implemented an active machine learning (AML) framework based on open-source software. Within the paper we discuss different query strategies to optimize the selection of instances during the learning phase to decrease the required training data.

The approach was evaluated within the domain of tenancy law. Thereby, we manually labeled the 532 sentences into eight different functional types and achieved an average F1 score of 0.74. Comparing three different classifiers and four query strategies the classification performance F1 varies from 0.60 to 0.93. We could show that in norm classification tasks AML is more efficient than conventional supervised machine learning approaches.

Keywords. norm classification, active machine learning, text mining

1. Introduction

More and more textual data that is relevant for the legal domain is digitally available. Algorithms and technological infrastructure for text mining and natural language processing are becoming more powerful in terms of their accuracy and performance. The use cases and tools for text mining in the legal field that are relevant for legal experts or practitioners, e.g., scientists, lawyers, judges, courts, etc., are manifold. A recent overview was published by Ashley in 2017 [1].

From an algorithmical point of view two major approaches exist to structure textual data: rule-based (knowledge-based) approaches and machine learning (ML) (statistical). Both approaches are attractive and have their specific advantages and disadvantages. Nowadays, rule-based approaches are still more common in practice, although science focuses much more on ML (see [2]). Many different notions of ML exist that can be applied to classify, categorize, predict, or cluster textual data. Thereby, active machine learning (AML) seems to be highly attractive, since it decreases the effort of training by providing mechanisms to train ML classifiers more efficiently [3].

This paper describes the combination of rule-based text mining with AML, a specific form of semi-supervised ML, for the classification of legal norms. The

reminder of the paper is structured as follows: Section 2 provides a short overview of the related work, Section 3 describes the architecture of the AML approach, the dataset and used labels are discussed in Section 4, finally the approach and its performance is evaluated in Section 5.

2. Related Approaches in Norm and Sentence Classification

Maat and Winkels performed this task for Dutch legislative text [4,5]. Thereby, they achieved a remarkable accuracy of more than 90% by classifying 13 different classes using a Support Vector Machine (SVM). They also performed the classification using a context free grammar, i.e., rule-based approach, for the classification (see [5]).

Wyner et al. extracted rules from regulations using JAPE grammar and the GATE framework [6]. They have developed a methodology for the extraction of deontic rules using linguistic rules. The quality of the results is varying, but promising: several categories have been extracted with high precision and recall.

The research group of Ashley, Grabmair and Savelka [7,8] extracted of semantic information from legal documents, e.g., statutory texts and cases. Thereby, they used an Apache UIMA type system to extract legal concepts from vaccine injury decisions (see [7]). Beside these rule-based approaches they investigated the potentials of interactive ML in classifying relevancy during an analysis task of statutory texts [8]. They were able to show that this can lead to major improvements during classification tasks.

To the best of our knowledge no attempt to classify norms and sentences for statutory texts in Germany using an active or supervised machine learning approach has ever been made before.

3. Active Machine Learning to Classify Legal Norms

3.1. Knowledge Engineering with Rule-based Approaches

Especially for rule-based approaches, linguistic variation as well as vocabulary variety constitute challenges. This holds within a professional language as well as in technical languages. Variations of pronunciation, vocabulary, and inflections steadily occur. Current research is still facing the so-called paraphrasing issue. Two different people phrase the same message by different wording [9]. A knowledge engineer must pay attention to these facts in order to define proper rules. Although, rule-based approaches are not very popular at today's scientific conferences, they are still pre-dominant in practice [2].

3.2. Active Machine Learning

AML is an adapted form of semi-supervised machine learning, in which the training is done in so-called rounds. Within each round a pre-defined amount of instances are manually labelled. The instances are not randomly selected but determined by a mathematical founded query strategy. The process starts by utilizing

random queried instances (seed set) to initially train a classifier model (1). This trained model is used to predict the labels of the unlabeled instances (2). Based on a query strategy, the unlabeled norms are selected by the classifier to distinguish more efficiently between the types (3). Thereby, query strategies are algorithms using the output probabilities/scores of the classifier to calculate an informativeness measure such as the entropy. These instances (e.g. instances having the highest entropy) are presented to a person to be labeled and added to the training set consisting of the random queried instances. The other (not labeled) instances remain in the unlabeled dataset (4). This process is repeated until some kind of stopping criterion (e.g., confidence threshold, maximum number of rounds) is met [3].

We implement our approach with Apache Spark, which is a fast, fault-tolerant and general-purpose open-source cluster computing framework for large-scale data processing. Spark provides an ecosystem consisting of several components managing the basic functionality (e.g., memory management, task scheduling). Apache Spark offers a ML library called MLlib[1] consisting of a variety of efficient and scalable implementations of common ML settings to conduct (semi-)supervised and unsupervised ML. Additionally, MLlib provides so-called ML Pipelines that facilitate the execution of typical ML classification tasks, i.e., preprocessing, feature extraction, and classification.

AML is an iterative and interactive extension of conventional semi-supervised ML. The key hypothesis of AML is that if the learning algorithm can select the data from which it learns, it will perform better with a smaller training set resulting in a more efficient learning.

3.3. Best-of-breed: Combining Rule-based and Active Machine Learning

The discussed approaches can be combined to tackle two challenges. Firstly the generation of labeled datasets that can be used for supervised machine learning techniques in text analysis and secondly the classification of textual data.

As described in Section 3.1, information extraction based on explicitly formulated rules is an effective way of directly integrating the expertise of domain professionals into the process of knowledge engineering. However, generally rules fail to fully capture the broad linguistic variety encountered in natural language.

The combination of (active) ML and rule-based approaches seems suitable to address the aforementioned challenges assuming that rule-based information extraction suffers from low recall but high precision (assuming the rules are written correctly) and (supervised) ML needs large amount of training data for correct inference. Figure 1 shows the structure of the integration of these two approaches, implemented in different software components. The entities extracted with rules bootstrap the active machine learning part, where the domain expert monitors and supports the learning process by providing input for the ML component (see Section 4.4).

[1] https://spark.apache.org/mllib/, last access on 08/24/2017

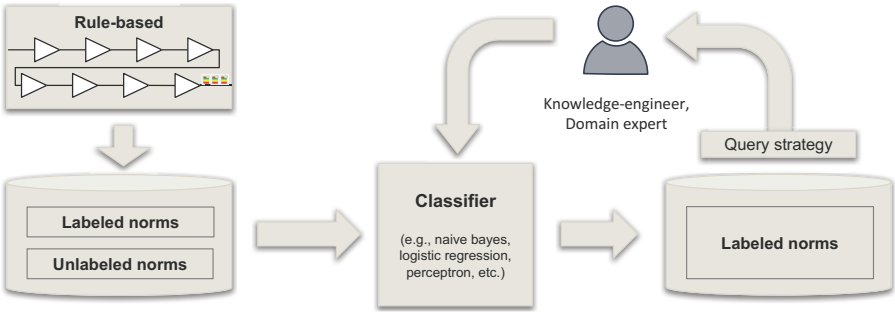

Figure 1. Combining rule-based and AML based approaches for classification of legal norms.

4. Norm Classification with Active Machine Learning

4.1. Objective

The classification of norms is, due to several reasons, attractive for the field of legal informatics. First of all, it allows a more elaborate differentiation of a norm's meaning and thus supports subsequent norm interpretation and formalization. Secondly, it is beneficial for the search and exploration tasks in legal information databases and consequently supports the efficiency of searching of and within legal documents. And finally, it helps determining references and dependencies between and within legal norms.

4.2. Types and Classes

Classification of legal norms can be addressed from different perspectives, e.g., from a philosophical, a legal theoretical or, a constructive one. To achieve the aforementioned tasks—a deeper understanding of interactions between legal norms—, we chose a classification regarding functional types. The taxonomy as well as the gold standards was developed on German statutory legal norms by two legal experts.

In a functional norm classification system, legal norms can be divided into 4 types of statements: normative, auxiliary, legal-technical, legal-mechanism. Our taxonomy comprises normative statements into the following categories: statutory duties, statutory rights, shall-to-do rules and (positive/negative) statutory consequence rules. The taxonomy is shown in Table 1.

The category of statutory duties further comprises the subcategories of order and prohibition, the category of statutory rights is composed of the subcategories of permission and release. The type of auxiliary statement norms can be divided into statements about terms and statements about norms. The first category can be subdivided into explanatory, extending and limiting statements, in which the explanatory statements include the subcategories of definition and precision statements. The category "statement about norms" is subdivided into modifications, legal validity, scope and area of application categories. Where the norms are dominated by their legal-technical or legal-mechanism nature, we identified the categories of reference and continuation in the first section and the categories

Normative statements	Statutory duties		Order
			Prohibition
	Statutory rights		Permission
			Release
	Shall-to-do rules		Shall-to-do rules
	Legal consequences		Legal consequences pos.
			Legal consequences neg.
Auxiliary statements	Statements about terms	Explanatory statements	Definition
			Precision
		Extension and limitations	
	Statement about norms	Legal validity	Legal validity and non-validity
		Scope of application	Temporal
			Personal
			Factual
		Area of application	Extension
			Limitation
			Definition
		Modifications	
Legal-technical statements			Reference
			Continuation
Legal-mechanism statements			Procedure
			Objection

Table 1. Functional type classification of statutory legal norms for Germany's legislative texts.

of procedure and objection in the second section. Table 1 shows 22 types are identified, with considerable differences in their support within the tenancy law.

4.3. Data

In order to prepare a suitable dataset for the norm classification experiment, a legal expert assigned a type to every sentence of the tenancy law section in the German civil code (§535 - §595) published on March 1st, 2017. The result was 532 labeled sentences using 16 different labels. As 16 of the 22 labels had a support less than 1,2%, they were removed from the dataset used. The 504 remaining sentences used for this classification task were composed of the eight classes illustrated in Table 2.

From this dataset, 126 sentences (25%) were randomly added to the test set. The remaining 378 sentences (75%) were used for iterative training. It was ensured that enough instances of each class were in both datasets. We used tokens and their POS tags as features to represent norm instances.

4.4. Experiment and Query Strategies

In this experiment, nine combinations using AML query strategies (see Tables 3 and 4) as well as three combinations using conventional supervised learning (CSL)

Type (German)	Type (English)	Occurrences	Support
Recht	statutory rights	126	25,00%
Pflicht	statutory duties	109	21,63%
Einwendung	objection	92	18,25%
Rechtsfolge	legal consequence	50	9,92%
Verfahren	procedure	49	9,72%
Verweisung	reference	46	9,13%
Fortführungsnorm	continuation	19	3,77%
Definition	definition	13	2,58%

Table 2. Types and statistics of used and manually labeled dataset.

were conducted for each classifier. In CSL, instances are queried randomly without applying any query strategy. These query strategies refer either to uncertainty sampling (US) or to the more elaborated query by committee (QBC) methods. While the former uses only one classifier model, the latter creates a committee of classifiers with the intention to cover a larger area of the version space. To create the classifier committee, the composition of the training data was adapted for each committee. Except for the QBC Vote Entropy strategy, all strategies take advantage of the output probabilities.

Query Strategy	Method	Description
Uncertainty Sampling (US)	Entropy	Selection based on the avg. information content (Shannon entropy) of an instance.
	Margin Sampling (MS)	Selection based on the output margin of the predicted outcomes with the highest prob.
Query by Comittee (QBC)	QBC Vote Entropy (VE)	Selection based on a committee of different QS methods (ensemble with majority vote).
	QBC Soft VE	Selection based on a committee of different QS methods (ensemble with majority vote, including probabilities).

Table 3. Query strategies for active machine learning.

As the MLP does not produce any output score, only the QBC vote entropy approach could be used with this classifier. Each of these twelve combinations was executed five times and averaged to obtain a significant and comparable result.

In the first round, instances (seed set) were randomly queried from the unlabeled training set, labeled and used for learning in the first round. In the subsequent rounds, again either the five most informative instances in the case of a

Abbr.	Classifier	Query Strategy
NB	Multinomial Naive Bayes	Entropy, MS, QBC VE, QBC Soft VE
LR	Logistic Regression	Entropy, MS, QBC VE, QBC Soft VE
MLP	Multi-layer Perceptron	QBC VE

Table 4. Combination of the applied evaluation settings.

query strategy were used; or five random instances were removed from the unlabeled training set, labeled and added to the labeled training set. For both classifiers used (NB and LR), five-fold cross validation was applied to ensure that these predictions were made with the best model found.

After each round, the resulting pipeline model was applied to the test data to evaluate the performance of the current model. This process was repeated until all instances of the training set were labeled (72 learning rounds in total).

4.5. Parametrization of Classifiers

The classifiers NB has been used with standard parametrization of MLLib. Due to performance reasons, the number of iterations for LR was decreased from 100 (default) to 10. The MLP had four layers, whereas the number of nodes of the two intermediate layers was 20 and 10, respectively. The size of the input layer was 2^{13} and the size of the output layer eight (i.e., number of types). The size of the seed set was 18 instances for each of iteration of norm classification.

5. Evaluation

The objective of this experiment was to evaluate (1) the potential of AML compared to CSL and (2) the quality of legal norm classification using ML/AML.

To achieve this, the model was evaluated with an independent test set after each round. To compare the performance of the AML approach, we used standard evaluation metrics[2]: *precision*, *recall*, F_1 and *accuracy*. Additionally, learning curves were utilized to monitor and visualize the learning progress.

None of the four used query strategies had shown to be significantly predominant compared to the others. Thus, the *average accuracy* combines the result of all query strategies used for the classifiers NB and LR, respectively, is visualized in Figure 2. It shows the performance of classifiers applying AML techniques opposed to CSL methods querying random instances.

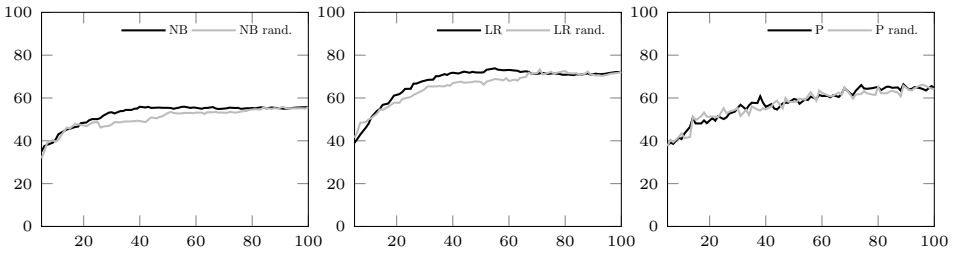

Figure 2. Average accuracy of classifiers vs. random learning (NB=Naive Bayes, LR=Logistic Regression, P=Perceptron.). Y-axis is accuracy in %, X-axis is labeled instances in %.

It becomes evident that AML is clearly superior to CSL when using NB and LR. The use of AML increased not only the speed of learning, but also resulted

[2]Note: no binary classification

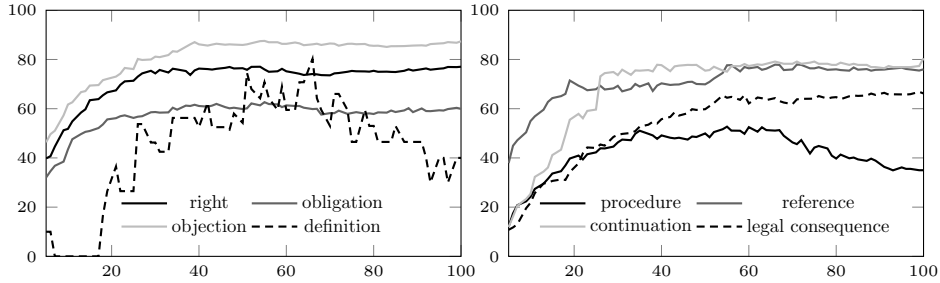

Figure 3. Average F_1 per class (active LR). Y-Axis is F_1 in %, X-axis is labeled instances in %.

in a higher maximum accuracy obtained during the classification process. In both experiments, the average accuracy was after a short "discovery phase" up to 5%-10% higher when having labeled 20%-70% of the instances compared to the random approach. Additionally, the accuracy obtained was higher all instances. Increasing the number of AML rounds, the chance of overfitting is increasing as well, so that after a certain number of labeled instances (70%-95%) both approaches align to the same final accuracy.

When analyzing the results of the individual learning rounds of a specific combination, the importance of having a "high quality seed" set becomes clear. As

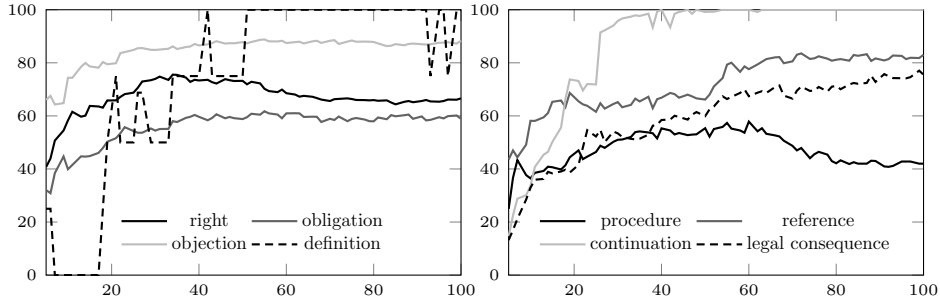

Figure 4. Average precision per type using logistic regression classification. Y-Axis is precision in %, X-axis is labeled instances in %.

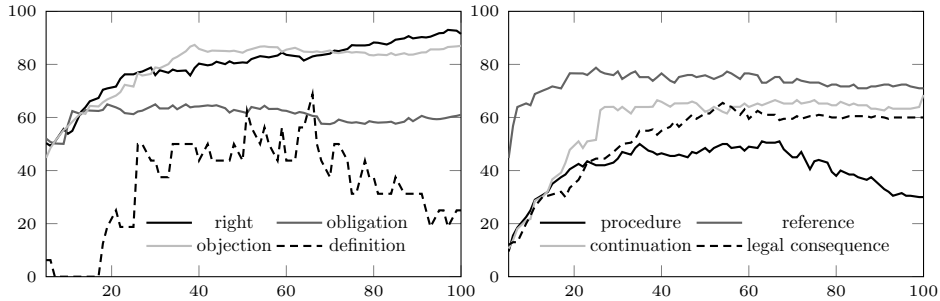

Figure 5. Average recall per type using logistic regression classification. Y-Axis is recall in %, X-axis is labeled instances in %.

the seed set in this study was created randomly for each experiment, the learning differs especially in the initial phases. Only after a discovery of the version space (discovery phase), AML was significantly superior to CSL. An improved coverage of the version space resulted in an almost 20% higher accuracy having labeled only 17% of the instances. Further, a maximum accuracy of almost 80% could be achieved having labeled only 35% of the instances (see Figure 3). An increase of more than 6% compared to CSL using 65% less instances.

To analyze the recognition of individual classes, *consolidated evaluation measures* (averaging the results of all four query strategies) obtained by the LR, the best classifier, are used. Figure 3, 4 and 5 show the consolidated curves.

Thereby, the different (final) results of the individual labels are very noticeable. While norms belonging to the type *objection* are well recognized, soon having an F_1 of almost 90%, towards the end, norms referring to the type *definition* or *procedure* cannot be classified easily by a classifier. The reason for the low end-value for the type *definition* might be their low support - with less than 3% - resulting in an only very small training set. Despite the fact that the training set for the type *continuation* contains only two more instances this type has an F_1 value of more than 80%. Thus, the classifier might also have problems to distinguish a *definition* from other types or the kind of *definitions* in the training set is linguistically varying from the one of the test set (different sub-type).

However, considering the intermediate results, the types *continuation* and *definition* that have only a very low support in the dataset, have both a very high precision and also a good recall temporarily. Hence, the reason for the worsening results is more likely caused by the overfitting of the classifier. This can be confirmed by the results attained by the type *procedure* that achieves much better results during the classification process. Nevertheless, this type shows the worst results having both a low precision and recall. Although the number of training instances is high for the type *obligations*, the classifier has problems recognizing them in the test set. The norm type *right* had the highest recall towards the end, but a rather low precision (see Figures 4 and 5).

6. Outlook and Future Work

This work is an additional step towards supervised machine learning with the objective to decrease the effort of labeling. Based on the results of this study, we see several next steps that can be addressed: i) Deep investigation of the reasons why the F1 measure for different norm types differ so heavily? ii) How do comparably low support of norm types (e.g., definitions) effect the classifier and how can negative impacts be avoided? iii) Does the full-stack integration of AML and rule-based approaches lead to even better performance and faster learning?

Beside these technical questions it would be interesting to adapt and apply this method to statutory (or judicial) texts of foreign languages, e.g. english. This could support current ongoing research projects, e.g. [8,10].

7. Summary

This paper describes active machine learning to classify legal norms in German statutory texts. Thereby, the classifier is trained in multiple rounds using a mathematical function, i.e. query strategy, which selects the most informative instances. This leads to an efficient learning for the classifier an minimizes the required training data.

Based on a functional type classification of legal norms we evaluated the approach in the field of German tenancy law. We compared three classifiers and four different query strategies in 72 learning rounds. For certain norm types, e.g., objections, rights, and obligations, a high detection accuracy of about 0.90 was achieved.

We consider this as a fruitful research direction to decrease the efforts required in supervised machine learning approaches for legal text classification.

Acknowledgment

This research was partially funded by an R&D grant on Contract Analysis by SINC GmbH, Wiesbaden, Germany.

References

[1] K. D. Ashley, *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge: Cambridge University Press, 2017.

[2] L. Chiticariu, Y. Li, and F. R. Reiss, "Rule-based information extraction is dead! long live rule-based information extraction systems!" in *EMNLP*, no. October, 2013, pp. 827–832.

[3] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[4] E. de Maat, K. Krabben, and R. Winkels, "Machine Learning versus Knowledge Based Classification of Legal Texts," in *JURIX*, 2010, pp. 87–96.

[5] E. de Maat and R. Winkels, "Automated Classification of Norms in Sources of Law," in *Semantic processing of legal texts*, E. Francesconi, Ed. Springer, 2010, pp. 170–191.

[6] A. Z. Wyner and W. Peters, "On rule extraction from regulations." in *JURIX*, vol. 11, 2011, pp. 113–122.

[7] M. Grabmair, K. D. Ashley, R. Chen, P. Sureshkumar, C. Wang, E. Nyberg, and V. R. Walker, "Introducing LUIMA: An Experiment in Legal Conceptual Retrieval of Vaccine Injury Decisions Using a UIMA Type System and Tools," in *ICAIL '15: Proceedings of the 15th International Conference on Artificial Intelligence and Law*. New York, NY, USA: ACM, 2015, pp. 69–78.

[8] J. Šavelka, G. Trivedi, and K. D. Ashley, "Applying an Interactive Machine Learning Approach to Statutory Analysis," in *JURIX 2015*.

[9] L. Romano, M. Kouylekov, I. Szpektor, I. Dagan, and A. Lavelli, "Investigating a generic paraphrase-based approach for relation extraction," in *11th Conference of the European Chapter of the Association for Computational Linguistics*.

[10] V. Walker, J. Hae Han, X. Ni, and K. Yoseda, "Semantic Types for Computational Legal Reasoning: Propositional Connectives and Sentence Roles in the Veterans' Claims Dataset," in *ICAIL '17: Proceedings 2017*.