Department of Informatics
Technical University of Munich

Bachelor's Thesis in Information Systems

Supporting the Legal Reasoning Process by Classification of Judgments Applying Active Machine Learning

Linus Boehm

Department of Informatics

Technical University of Munich

Bachelor's Thesis in Information Systems

Supporting the Legal Reasoning Process by Classification of Judgments Applying Active Machine Learning

Unterstützung des Legal Reasoning Prozesses durch Urteilsklassifikation mittels Active Machine Learning

Author:	Linus Boehm
Supervisor:	Prof. Dr. rer. nat. Florian Matthes
Advisor:	M.Sc Ingo Glaser
Submission Date:	16.04.2018

Declaration

Ich versichere, dass ich diese Bachelor's Thesis selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

München, den 16. April 2018

Linus Boehm

Abstract

The Digitization of information is transforming the way we live and creating many new business models. Digitization is also taking place in the legal domain. Legal documents, such as contracts and general terms and conditions, are produced thousands of times a day thanks to numerous online contract generators, e-commerce platforms, banks and insurance companies. Due to this increase of available unstructured data and the enhanced capabilities of algorithms and computing power, the demand for automated data processing, e.g. text classification is increasing. The purpose of this research is to get a better insight into active machine learning and binary legal text classification to see if this approach can support the legal reasoning process.

Contents

Abbreviations

AML	Active Machine Learning
API	Application Programming Interface
BGH	Bundesgerichtshof
CL	civil law
FE	feature extraction
FN	false negative
FP	false positive
FS	feature selection
ML	machine learning
NER	named entity recognition
NLP	natural language processing
POS	part of speech
TF-IDF	Application Programming Interface
TN	true negative
TP	true positive
ZB	Zettabyte 1ZB $= 10^{21}$ bytes
ZPO	Zivilprozessordnung

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The Digitization of information is transforming the way we live and creating many new business models. Autonomous cars, Internet of Things, Social Media and Artificial Intelligence are just examples for a few trend technology's that make heavily use of digital available data. In 2016, 16.1 ZB of data were generated worldwide. According to estimates, 80% of the new generated data is unstructured [Raghavan et al., 2004].[1] By the year 2025, the amount of data generated is expected to rise up to 163 ZB [David Reinsel, 2017].

Due to this increase of available unstructured data and the enhanced capabilities of algorithms and computing power, the demand for automated data processing, e.g. text classification, pattern finding and knowledge extraction, is increasing and is an important area for research [Khan et al., 2010]. One measure of progress in Machine Learning, is the significant amount of existing real-world applications, like Speech recognition, Computer vision, Robot control and Accelerating empirical sciences [Mitchell, 2006]. Past research has shown the successful application of various Machine Learning classification algorithms on text-based data.

Digitization is also taking place in the legal domain. During the last legislative period (2013-2017) of the German parliament more than 550 laws were updated or created.[2] Most of the laws are available online.[3] Every year, more than 6000 judgments are adjudicated at the German Federal Supreme Court

[1] https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know

[2] https://www.bundestag.de/blob/194870/7c8a01e16c98fc9c32ddb203d7bd88e0/gesetzgebung_wp18-data.pdf

[3] https://www.gesetze-im-internet.de/

(BGH).[4] Over 40,000 of these decisions can be accessed through an official database, that exists since 2016.[5] Other legal documents, such as contracts and general terms and conditions, are produced thousands of times a day thanks to numerous online contract generators, e-commerce platforms, banks and insurance companies. This information inflation the legal domain arises new challenges, especially for judges and lawyers [Paul and Baron, 2006].

This information inflation makes automatic text classification of legal texts through machine learning an attractive and promising research topic.

1.2 Research Questions

The purpose of this research is to get a better insight into active machine learning and binary legal text classification to see if this approach can support the legal reasoning process. After having dealt more deeply with the topic, the theoretical findings will flow into the development of a prototype for the binary classification of sentences. Subsequently, the performance of the prototype is evaluated by a use case for classification of civil judgments.

[4] http://www.bundesgerichtshof.de/SharedDocs/Downloads/DE/Service/StatistikZivil/jahresstatistikZivilsenate2017.pdf?__blob=publicationFile

[5] https://www.bmjv.de/SharedDocs/Pressemitteilungen/DE/2016/01272016_Webservice_www_rechtsprechung_im_Internet_de_geht_online.html

2 Legal Knowledge Base

2.1 Legal Systems

Based on the definition of [Tetley, 1999], the term "legal system" refers to the general nature and content of the legislation and to the constructions and procedures in which they are legislated upon, adjudicated upon and administered upon, in a particular jurisdiction. The legal systems of the contemporary western world are divided into two groups: common law and civil law. The long-standing legal tradition characterizes both legal families. A legal tradition is a set of deeply rooted, historically conditioned positions about how the legislation is passed, applied, studied, perfected, and taught [Tetley, 1999].
A comparison of the two major legal families of civil law and common law succeeds only from the distance of a historical perspective. Many because the closer one looks, the more the differences disappear. The common law, which has its origin in England, shaped the law in the USA, Canada, New Zealand and from other former colonies. The counterpart to common law is civil law, which has influenced the legal system in South America from its origins in Western European countries, such as Germany, France, Italy and the Netherlands[Röhl and Röhl, 2008].

2.1.1 Civil Law

The civil law, also called Romano-Germanic law, is originated in continental Europe. The jurisprudence of the civil law has developed from the Roman law, which was codified in the Corpus Iuris Civilis. The civil law heavily on abstract rules and definitions, often ignoring the details. These rules of civil law are conceptualized as behavior rules closely linked to ideas of justice and morality. The codified corpus of a civil law-based legal system is profoundly organized and structured. The codified core principles and regulations serve as the primary source of law. Another characteristic of the civil law family is the partly evolvement of the law as private law, that means that it encompasses the regulation of private relationships between individual citizens [Tetley, 1999, David and Brierley, 1978, Röhl and Röhl, 2008].

2.1.2 Common Law

The common law evolved from the law of England. During the colonial era, the legal system spread to North America, Australia, and other former British Commonwealth states. While countries with civil law systems have comprehensive, continuously updated legal codes, which are adopted through the legislative, the common law was formed mainly by judges who had to adjudicate about specific legal cases. Therefore, the rules of common law are usually less abstract than the rules in civil law. The prescriptions of a common law system are largely based on precedents, which are continuously evolved through new judicial decisions [law and civil law traditions, 2006, Tetley, 1999, David and Brierley, 1978, Röhl and Röhl, 2008, Levi, 1948].

2.2 Legal Reasoning

The legal reasoning process of a lawyer and a judge is slightly different. A broadly worded explanation would be that a Civil Law attorney reviews the table of contents of a comprehensive legal book, which is based on a systematic structure, to resolve a specific legal issue. In contrast, the common law lawyer would start in the alphabetical index [Röhl and Röhl, 2008]. When lawyers get approached by clients with their issues and often a feeling of injustice, it is the lawyer's job to determine relevant laws, precedent cases, and facts and integrate them into his legal reasoning to solve the issue in favor of the client. Based on the legal reasonings and the facts presented to the court by

the lawyers, the judge may agree one of the legal reasoning or may construct a own legal reasoning with possible additional or new legal interpretations not mentioned before by the parties [Ellsworth, 2005, Fellmann et al., 1968].
Although both legal systems are increasingly converging in some areas, the legal reasoning process and the way lawyers and judges apply jurisprudence still differ. The justification of a civil law jurisprudent arises from deductive reasoning, while a common law lawyer uses analogical reasoning. The deductive legal reasoning of a jurisprudent emerges mostly within a framework established by a comprehensive, codified set of rules [law and civil law traditions, 2006, Ellsworth, 2005, Fellmann et al., 1968]. The analogical legal reasoning process is a three-step approach described by the doctrine of precedent. The steps are these: (1) finding similarity in previously decided cases with comparable fact situation; (2) extraction of the rule of law from the previously decided case; and (3) application of the extracted rule of law to the case at hand [Herman, 2008, Levi, 1948].

2.3 Proceedings in Civil Cases at the Federal Court of Justice of Germany

The Federal Court of Justice (BGH; Bundesgerichtshof) is a court of appeal, which means that judgments are exclusively handed to it by inferior courts for reviewing for errors of law. The remedy of appeal on points of law is only available against final judgments adopted by regional and higher regional courts acting as appellate courts. Consequently, the BGH does not perform an own fact-finding or evidence-taking. After an appeal was considered as admissible by the panel, an oral-hearing is held resulting in a written judgment. If an appeal is seen as inadmissible, it will be dismissed by way of a court order [Bundesgerichtshof, 2014].
The BGH has twelve civil panels that are traditionally high specialized for specific areas of law. In the context of this thesis, we only consider judgments of the eighth civil panel, who is specialized in law on the sale of goods, landlord and tenancy law.

Table 2.1: General structure of a German civil law judgment

(1) Recital of parties; Introduction (Rubrum)
(2) Tenor
(3) Summary of circumstances (Tatbestand)
(4) Opinion of the court (Entscheidungsgründe)
(5) Instruction on the right of appeal (Rechtsmittelbelehrung)
(6) Signatures of the judges

Source: Own illustration based on [Hofmann, 2018]

2.3.1 Structure of a Civil Law Judgment

The court procedure in civil proceedings is mostly regulated by the civil procedure code (Zivilprozessordnung; ZPO), as is the general structure of a court decision in civil matters. A civil judgment is regularly divided into six parts, which are shown in table 2.1. The most important parts are listed below:

(1) Recital of parties (Rubrum)

The so-called recital of parties names in addition to the parties and their address, the type of judgment, the address of the court and the case reference. The case reference consists of the initials of the court, the elaborating panel of the court, a register reference and an ongoing case number succeeded by the year of receipt [Hofmann, 2018].

(2) Tenor

The tenor forms the essence of every judgment and states the legal consequence ordered by the court, e.g., to pay the amount claimed [Hofmann, 2018].

(3) Summary of circumstances (Tatbestand)

The summary of circumstances reflects the essential facts that are related to the decision.

(4) Opinion of the court (Entscheidungsgründe)

In addition to the opinion of the BGH, the reasoning of the lower court is also included. The argumentation of the lower court is written in indirect speech to distinguish it from the opinion of the BGH. The reasoning is written in the so-called judgment style, which begins with the result, followed by a gradual justification [Hofmann, 2018].

3 Machine Learning

Machine Learning (ML) is a combination of data analysis techniques from statistics and computer science [Witten et al., 2016, p. 30]. One measure of progress in Machine Learning, is the significant amount of existing real-world applications, like Speech recognition, Computer vision, Robot control and Accelerating empirical sciences [Mitchell, 2006]. Tom Michell defines the task of learning, by a computer program that improves its performance with experience, as follows [Mitchell, 1997, p. 2]:

> A computer program is said to learn from experience E with respect to some class of Task T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E

There are several forms of Learning, the two classic forms are supervised and unsupervised learning. In both cases, we want to match a set of samples $X = (x_i, \ldots, x_n)$ to a state of nature l_n (often called label or class in context of ML) with probability $P(l_j)$. The right label is the one with the highest probability $P(x_i \mid l_j)$ [Duda et al., 2002, p. 85][Alpaydin, 2014, p. 9].

Supervised Learning

In supervised learning the aim is to learn the mapping between the samples x and the labels y which are predefined by a "supervisor". The classifier gets a Training Set made of pairs (x_i, y_j) of samples with their associated label. Because the label mappings are predetermined, the performance of the algorithm can be easily evaluated on his predictive performance (see **??**) [Chapelle et al., 2010, p. 1].

In general, there exist three different label assignment settings for supervised learning classification tasks [Chapelle et al., 2010]:
Binary classification:
Binary classification (or filtering) is the task of classifying the instances x with a single label y from a set of labels $|Y| = 2$.
Multiclass classification:
If the label set consists of more than two labels $|Y| > 2$, then the classification task is called multiclass classification. The multiclass classification has the same restrictions on the label association as binary classification. Therefore, every instance y is associated with a single label y.
Multi-label classification:
Multi-label classification tasks associate every sample x with a subset L from the label set $L \subseteq Y$.

Unsupervised Learning

In unsupervised learning there is no such "supervisor" and the only input are the samples. The aim is to find regularities, like patterns or clusters, in the input data. The assumption is, that the input feature vectors are from a underlying common distribution of X. The aim of unsupervised learning is to find interesting structure, like patterns or cluster, in the input data X [Chapelle et al., 2010]. The kind of structure which is found, is determined by the algorithm and the data preprocessing. Unsupervised learning is often used in image processing or image compression applications
[Duda et al., 2002, p.16,85] [Alpaydin, 2014, p. 9-12].

Semi-supervised Learning

Semi-supervised learning (SSL) combienes the two approaches of supervised and unsupervised learning. The algorithm processes not only unlabeled data but also labeled instances. Therefore the training set can be divided into labeled instances and unlabeled instances. However, given labels do not necessarily have to cover all possible occurring labels. [Gabrys and Petrakieva, 2004, Albalate and Minker, 2013]. The Question arises when Semi-supervised learning produce a more accurate prediction than supervised learning. This is the case when unlabeled samples will help to illuminate the underlying distribution of the feature vectors. In other words, the knowledge on $P(x)$ which is gained through the unlabeled instances has to help with the determination of $P(y \mid x)$. Otherwise, semi-supervised learning will not yield to a better prediction than supervised learning. It might even happen that the use of unlabeled data reduces the accuracy of the prediction[Chapelle et al., 2010].

3.1 Text Classification in Context of Legal Texts

To make a classifier understand an unstructured text, the input text has to be transformed into a feature vector representation [Khan et al., 2010]. There are several techniques for generating features from the input text and represent them as a vector, which is discussed further in section 3.1.1.2. For example, if you use the bag of words representation, every word in a text could be a potential feature for the classifier. Consequently, the number of features can exceed the amount of training data multiple times. This extremely high dimensionality of the text representation is one of the significant challenges of text classification tasks [Joachims, 1998, Khan et al., 2010, PAK and GUNAL, 2017].

In the literature document categorization, text categorization or document classification are often used as synonyms. For more or less the same thing Therefore, this term needs a clear definition for the purpose of this thesis. The process of classifying is usually composed of various tasks: (1) Feature Generation, (2) Vector Representation and the actual (3) learning process with the classifier. [Khan et al., 2010]

3.1.1 Text Pre-Processing

There are different opinions and definitions of text pre-processing and what tasks belong to the concept of text pre-processing. In context of this thesis, we define text pre-processing as a general term for all techniques that aim to ensure the quality of the vector representation of the text input to improve the accuracy of predictions made by the classifier [van den Bosch, 2017, Khan et al., 2010]. To accomplish this, the task of text pre-processing is to generate features from the text input, transform them into a feature vector representation suitable for the selected classification algorithm and then perform a dimensionality reduction on the feature vectors without loosing much information [Khan et al., 2010, Khalid et al., 2014, Joachims, 1998]. When it comes to dimensionality reduction, the literature mentions two common techniques: Feature Extraction (FE) and Feature Selection (FS) [Alpaydin, 2014, Khan et al., 2010]. A feature extraction or feature selection with the goal of dimensionality reduction is not used in this thesis and therefore needs no further explanation. Accordingly, the text pre-processing task is divided into the Feature Generation phase (3.1.1.1) and the Vector Representation (3.1.1.2) of the features.

3.1.1.1 Feature Generation

The term feature generation has many synonyms, such as feature construction, feature engineering, feature extraction or feature reduction
[Scott and Matwin, 1999, Gabrilovich and Markovitch, 2005, Motoda and Liu, 2002]. Two possible objectives of this method are improving the accuracy or reducing dimensionality. If the main focus lies on the dimensionality reduction of the feature set, then the resulting feature space contains less features than the original set. In Contrast, the resulting feature space of a method that aims to improve the accuracy will most likely consist of more features than the orgininal feature set [van den Bosch, 2017, Motoda and Liu, 2002].
In the context of this thesis, we use different well known feature generation methods to improve accuracy without focusing on a dimensionality reduction.

Therefore, we define the term feature generation as the process that extracts a set of new features from one or multiple existing features with the aim to improve accuracy [Motoda and Liu, 2002, Gabrilovich and Markovitch, 2005, Cohen et al., 2004, van den Bosch, 2017].

(1) POS Tagging Filter

Part of speech (POS) taggers are used in various nartual language processing (NLP) and text processing tasks [Dale et al., 2000]. The added information about the part of speech can be filterd to use only certain tags to be included in the feature vector. By using only lemmatised words tagged as nouns, adjectives or proper nouns and applying a normalised term frequency, study [Gonçalves and Quaresma, 2005] seen an improvement in the F1 score.

(2) Named Entity and Reference Tagging

A problem that occurs particularly in German legal texts is the massive use of abbreviations, dates in different formats and references to entities like contracts, laws, judgments or institutions. One way to address this problem would be to perform a named entity recognition (NER) to replace all found references with their associated named entity. For example, references to the German Civil Code (BGB), such as "§307 Abs. 1 Satz 1, Abs. 2 Nr. 1 BGB" (Abs. stands for paragraph), could be replaced and with a more general token, e.g., $legislativReference$. As a result, the feature set is reduced, and better generalizability of the classifier can be achieved [Schölkopf and Smola, 2002, Biagioli et al., 2005]. For more information about NER in German legal documents see [Glaser et al., 2018, Glaser, 2017].

(3) Tokenization

The task of tokenization is to break the raw input text into words, phrases or other significant pieces called tokens. Depending on the classification task, punctuation marks, HTML/XML tags and special characters (e.g., brackets) can be removed by the tokenizer [Kannan and Gurusamy, 2014, Allahyari et al., 2017].

(4) N-Grams

Following the process of tokenization, the text is present as a sequence of single words, which can be considered as N-grams with size one (also called unigrams). When building an n-gram model, each n-gram is getting composed of n words. The basic approach is to combine each n successive words to an n-gram, where the following n-gram starts one word after the previous n-gram so that there is an overlapping with the last n-gram by (n-1) words. The intention behind using n-grams is that single words are not as meaningful as a combination of n-words. Walter combines the n-gram approach with POS-filtering by building bigrams (n-grams with size two) consisting of a noun and an adjective [Walter and Pinkal, 2006].

(5) Stopwords Removing

Frequently occurring words such as prepositions or conjunctions that provide little information about the content of the text are called stopwords. To prevent their frequent occurrence from affecting the result of classification algorithm, they are commonly removed [Allahyari et al., 2017, Kannan and Gurusamy, 2014]. Removing the stopwords has allowed [de Maat et al., 2010, Lewis, 1992] to achieve better classification accuracy, while [Pomikálek and Rehurek, 2007] has not observed any significant improvement in accuracy and [Méndez et al., 2005] has observed a decrease in accuracy. Removing stopwords in legal texts can lead to sentences that have a different meaning, e.g. when words such as is and not are removed.

(6) Lemmatization and Stemming

Both stemming and lemmatization aim to transform words into their basic form. The stemming process transforms the words into a common form by an algorithm, where the resulting basic form of the words does not necessarily represent the correct dictionary form [Allahyari et al., 2017, Kannan and Gurusamy, 2014]. In contrast, lemmatization performs a morphological and vocabulary analysis and trys to remove inflectional endings from the word, allowing words to be transformed back into their dictionary form. [Balakrishnan and Lloyd-Yemoh, 2014]. The influence of stemming and lemmatization on the results of information retrieval, especially in the legal domain, has been discussed in many papers, such as [Biagioli et al., 2005, de Maat et al., 2010, Gonçalves and Quaresma, 2005, Turtle, 1995, Walter, 2008]. Some early studies on stemming have shown a negative impact on precision and recall, partly due to the poor performance of the stemming algorithm [Frakes, 1992]. Balakrishnan [Balakrishnan and Lloyd-Yemoh, 2014] showed that both, stemming and lemmatization, have a positive impact on revival performance, while [de Maat et al., 2010] observed a negative impact on accuracy by applying stemming on dutch laws.

3.1.1.2 Vector Representation

The vector representation process transforms the resulting text features after the feature generation phase into a vector representation suitable for the learning algorithm. The vector representation of a text classification problem has a substantial impact on the generalization accuracy of the classifier [Joachims, 1996]. There exist different methods on how to represent the sequence of text features as a vector, most of them neglecting the order of the words and make use of a weighted vector of terms. After the definition of essential terms by the feature generation process, a vocabulary V of unique terms (e.g., words) can be created from the set of all training instances. By building a vector of weights $w_1, \ldots, w_{|V|}$, every w_i represents the amount of information of the ith element of the vocabulary which was assigned by the text representation method.

Bag of Words

The bag of words representation is one of the most popular representation methods for text classification. The bag of words model ignores the exact ordering of the terms in a document but assumes that the frequency of a word is significant [Christopher et al., 2008, Francesconi and Passerini, 2007]. The dimension of the bag for an individual query is the number of unique words in the vocabulary where each unique word operates as a key for a bag and the term frequency (defined as $tf_{t,d}$) stored as the value (weight). If a specific word is not included in the selected instance, then the corresponding value is zero. The assumption behind taking the term frequency into account is that the more often a word occurs in a corpus, the more relevant is the word for the meaning of the document. Less meaningful words, such as stop words, which occur too frequently can impact the generalizability of the bag of words model.

Binary Representation

Past research has shown, that the bag of words model does not always represent legal texts well. Other corpus types, like a news article, repeat relevant keywords quite often. Legal documents, such as court decisions or laws, the proper term possible appears only once besides lengthly argumentations and definitions. Especially when classifying sentences, counting term frequencies do not always perform well. A binary representation does only measure the presence or absence of a term within the training instance [Schweighofer et al., 2001].

TF-IDF

Another approach to represent text as a vector is TF-IDF, short for term frequency-inverse document frequency. Because raw term frequency suffers from the assumption, that all terms are equally important, the TF-IDF approach is trying to take the uniqueness of a term into account by counting the occurrences of the term in other documents. As stated above, term frequency considers the number of occurrences of each term in an instance (e.g. a document or a sentence). The inversed document frequency is defined as $idf_t = log(\frac{|D|}{df_t})$, where $|D|$ stands for the amount of instances in the training set. The function df_t symbolizes the number of documents in which term t occurs. The combination of these formulas yields in the tf-idf measure:

$$\text{tf-idf}_{t,d} = tf_{t,d} \times idf_t$$

The $tf - idf_{t,d}$ weight of a term t is increasing when t frequently occurs within very few instances and thus decreasing when t occurs fewer times in the document or occurs in many documents. The $tf - idf_{t,d}$ weight is lowest when t occurs many times in almost every document [Schütze et al., 2008].

3.1.2 Classifiers

3.1.2.1 Naïve Bayes

Naïve Bayes is a very popular and simple probabilistic classifier, which is based on Bayes‘ Theorem. This classifier "naïvely" assumes that all feature values are conditionally independent with each other given the target class. In other words, the assumption is that given the class c, the probability of observing the conjunction of different features $f_1 \dots f_n$ from a document d is simply the product of the probabilities for every observed feature: $P(f_1 \dots f_n \mid c) = \prod_i P(f_i \mid c)$

The assumption of independence has the consequence that the order and present of feature does not affect the appearance of any other feature [Witten et al., 2016, Mitchell, 1997, Khan et al., 2010].

By applying Bayes Theorem to the task of text classification, the probability that a document d belongs to class c can be expressed mathematically as:

$$P(c \mid d) = \frac{P(d \mid c)\, P(c)}{P(d)}$$

$P(d)$ can be ignored, since it is constant for all classes. The probability $P(c)$ can be easiely calculated by counting the occurence of class c in the training set. Because the possible number of occurring features is very high, calculating $P(d \mid c)$ might be very difficult [Domingos and Pazzani, 1997]. But if the features are independent given the class, which was the assumption, we can split the feature conjunction $P(df_1 \dots f_n \mid c)$ into the product of the single feature probabilities $P(f_1 \mid c) \dots P(f_n \mid c) = \prod_i P(f_i \mid c)$. The result is the following equation [Aghila et al., 2010, Mitchell, 1997, Friedman et al., 1997, Alpaydin, 2014]:

$$P(c \mid d) = P(c) \prod_i P(f_i \mid c)$$

Although the feature independence is not given in many real world scenarios, the Naïve Bayes Classifier can compete with many other algorithms, such as Linear Regression. Its simplicity and fast computability make it an often used algorithm for text classification [Muhr, 2017].

3.2 Active Machine Learning

Active Machine Learning (AML) is a subfield of machine learning. Classic machine learning algorithms need hundreds (or even thousands) of labeled instances in the training set to perform well. In applications where labels can be generated for free or for very low cost, the need for a large amount of training data is not a problem. In some use cases, such as speech recognition, information extraction and text classification, the costs to generate a label are quite high. This is especially the case in the legal domain, where labels often have to be generated or approved by a legal expert, like a lawyer or a judge. To counter the problem of the high costs for the generation of a label, AML tries to reduce the amount of required training instances by letting the algorithm decide which instances it wants to learn next [Settles, 2012].

Past research showed that active learning can achieve higher accuracy with fewer training data than classic machine learning [Settles, 2012, Muhr, 2017].

3.2.1 Active Machine Learning Scenarios

The literature states three different scenarios how an active learner access the training data to query instances. All three scenarios assume that a human oracle labels unlabeled instances from the generated queries [Settles, 2012].

3.2.1.1 Membership Query Synthesis

Membership Query Synthesis is an active learning scenario where the learner can ask for any unlabeled instance in the training set and creates new queries de novo. Therefore the scenario is not suitable for a legal text classification task, because the newly generated queries would often be nonsense or not even readable [Settles, 2012].

3.2.1.2 Stream-Based Selective Sampling

This scenario is called stream-based or sequential active learning, as each unlabeled instance is typically drawn one at a time from the data source, and the learner must decide whether to query or discard it. The resulting key advantage on the stream-based selective sampling approach is, that it consumes little memory and computing power and that's why it is mostly used with mobile or embedded devices. This scenario is only practical when unlabeled data can be gathered for free (or at low cost) [Settles, 2012]. This assumption might not applicable in the context of legal text classification.

3.2.1.3 Pool-Based Sampling

The pool-based sampling scenario consists of a small set of labeled training instances L and a large pool of unlabeled training data U. The labeled training set initially consists only of the seed set (see section 3.2.2) for the initial training round. In a turn-based process, the active learner uses the labeled training data to apply an informativeness measure on the unlabeled pool. Based on the measurement and the query strategy framework the active learner forms a query. Afterward, a human oracle annotates the instances in the query and adds the queried samples to the labeled training data [Settles, 2012].

3.2.2 Seed Set

The seed set is the initial training set, which is required for the first training round of the classifier. The success of the AML algorithm depends heavily on the quality of the seed set. Therefore, the selection of the initial training instances is crucial. The general approach is to generate the seed set by randomly selecting the desired amount of seeds. Random sampling is based on the assumption that the resulting seed set has the same or a similar distribution as the whole data set. Seed sets are chosen relatively small when compared to the entire data set; typical sizes are 10 or 20 instances. Due to this difference in size, it cannot be ensured through random sampling, that the produced seed set is representative. This can result in a mainly for AML susceptible phenomenon which is called missed cluster effect or missed class effect. The cause of the occurrence of this impact is the fact that the chosen seed examples influence the subsequent queries for the learning process. If the seed set is missing a sample which represents a specific cluster of the data, the classifier might become overconfident about the class of this region. Especially when the class label distribution is skewed, random sampling tends to miss a class or cluster [Settles, 2012, Dligach and Palmer, 2011].

When thinking about a binary classification task, the circumstance that only 3% or less of the data consists of "True" labeled examples is a frequent scenario. By random sampling 20 instances, the probability of having no "True"

labeled in the seed set is over 54%. For a heavily skewed label distribution in a multi-class classification problem the likelihood of missing a class is even higher. Therefore, there is a high risk that AML selects only those examples of the predominant class over the course of many iterations.
Tomanek et al. [Tomanek et al., 2009] analyzed the impact of the missed class effect, which is a special form of the missed cluster effect where complete label classes are missed by the AML classifier. The missed class effect is caused by an insufficient exploration phase during the seed set generation or in the course of the query generation in the learning phase [Tomanek et al., 2009, Schütze et al., 2006].

3.2.3 Query Strategies

3.2.3.1 Uncertainty Sampling

Lewis and Gale [Lewis and Gale, 1994] introduced an uncertainty sampling algorithm for text classifiers. The algorithm chooses only those instances whose label class is uncertain to the classifier. Therefore the classifiers estimates the label off all unlabeled instances based on the previously labeled instances. Uncertainty Sampling can be used straightforwad with any classifier that provides a measurement of how certain predictions for different labels are. That is the case for many classifiers, such as probabilistic, nearest neighbor and neural classifiers [Lewis and Gale, 1994]. When using a probabilistic classifiers for a binary classification problem the most uncertain instances are simply those whose posterior probability is closest to 0.5. Classification problems with more then two class labels need a more general approach. The Shannon entropy [Shannon, 1948] is a information-theoretic measurement method that measures the average amount of information of an instance based on all possible label classes.

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i$$

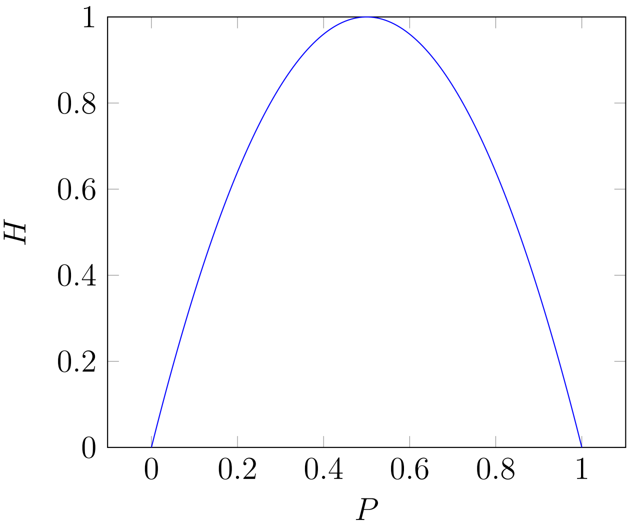

Figure 3.1: Plot of a entropy function for binary classification problem

Applying Shannon's entropy definition to the context of machine learning the entropy H of an unlabeled instance d is defined as:

$$H_d = -\sum_{i=1}^{n} P(c_i \mid d) \log_2 P(c_i \mid d)$$

Where $P(c_i \mid d)$ is probability that an instance d belongs to class c. The instance with the highest entropy represents the most uncertain. For a binary classification problem (Figure 3.1) the entropy function has its maximum for $p = 0.5$ and for $p = 0$ or $p = 1$ the entropy is zero. 6

3.2.4 Batch Size

The batch size defines the number of instances that are queried each learning round. The standard procedure is to query one instance at a time. For knowledge-intensive classification tasks which occur for example in the legal domain, the time required to generate a model using a serial query approach is expensive. Sometimes various human annotators want to train the model at the same time. In both cases a serial query approach is unpractical. Addressing this problem, querying multiple instances at once is known as the batch mode. The primary challenge in using batch mode is finding the best

Q instances. Probability-based query strategies, like uncertainty sampling, do not work as well with batch mode queries as they do with serial mode queries. The reason for this weakness is that two instances which are mutually similar or even identical, often have the same entropy values, and thus would be in the same query without providing any real information gain. This overlap of information makes the performance of a classifier that uses randomized queries better than those that only query the q-best instances [Settles, 2011].

3.2.5 Performance Measurement

Recall, precision and accuracy are well-known information retrieval standard measures to evaluate the performance of supervised text classification system. For a binary classification task, the prediction results can be illustrated through a confusion matrix. The matrix consists of four fields: true positive (TP), false positive (FP), false negative (FN) and true negative (TN). These four possible evaluation groups are assigned accordingly to the prediction result of the classifier and the desired output.

The four group names are a bit misleading because a prediction instance classified as "True" is called positive. When the predicted class matches the desired outcome, the result is assigned to an evaluation group containing true in its name. Hence, a sample which was classified as "True" by the binary classifier and the label was given correctly, is grouped as a true positive sample. Otherwise, if the label "True" was falsely assigned, then it is assigned to the false positive group. The true negative group is assigned, when the predicted and desired class are both "False". Consequently, the group false negative

Table 3.1: A confusion Matrix

	True	**False**
True	TP	FP
False	FN	TN

Source: own illustration

is assigned whenever the predicted class is "False" but the desired output is "True".

Table 3.2: Explanation of the confusion matrix evaluation groups for a binary classifier

Group	Definition
TP	Instances that are correctly classified with class "True"
FP	Instances that are falsely detected as "True"
FP	Instances that are correctly classified with class "False"
TN	Instances that are falsely detected as "True"

Source: Own illustration

Based on these four evaluation groups the following performance measurements can be defined:

$$\text{Recall} = \frac{TP}{TP + FP}$$

$$\text{Precision} = \frac{TP}{TP + FN}$$

$$\text{F}_1 \text{ (F-score)} = \frac{2 * \mathit{Precision} + \mathit{Recall}}{\mathit{Precision} + \mathit{Recall}}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

The recall indicates the proportion of samples correctly classified as positive (TP) of the entire amount of positive instances in the set (TP+FN; first column in the matrix). The precision indicates the proportion of correctly classified results of the total amount results classified as positive (first row of the confusion matrix).

4 Concept and Design

4.1 Involved Systems

Following the theory of AML and legal text classification, the characteristics of a German civil judgment and the discussion how AML can support the legal reasoning process, the findings are applied to a prototypical implementation. The implementation of the prototype builds on two existing web-based frameworks. Both systems were developed as part of the interdisciplinary research program Lexalyze [6] and the chair of "Software Engineering for Business Information Systems" at the Technical University of Munich (TUM). The initiative has set itself the task of developing interdisciplinary synergies between law and computer science.

4.1.1 Lexia Framework

Lexia is a ”data science environment for semantic analysis of German legal texts” [Waltl et al., 2016]. The collaborative web-based application allows the user, among other things, the analysis of laws, judgments and contracts [Waltl et al., 2016, Lexalyze, nd]. Apache UIMA [7] was used as baseline for the architecture for Text Mining Engine. Lexia was mainly used as a user interface for this work. Except for the importer and the database nothing was needed.

[6] Further information about Lexia and other research regarding Lexalyze can be found athttps://wwwmatthes.in.tum.de/pages/1rvivk51a20k4/Lexalyze-Interdisciplinary-Research-Program

[7] Apache UIMA, https://uima.apache.org/

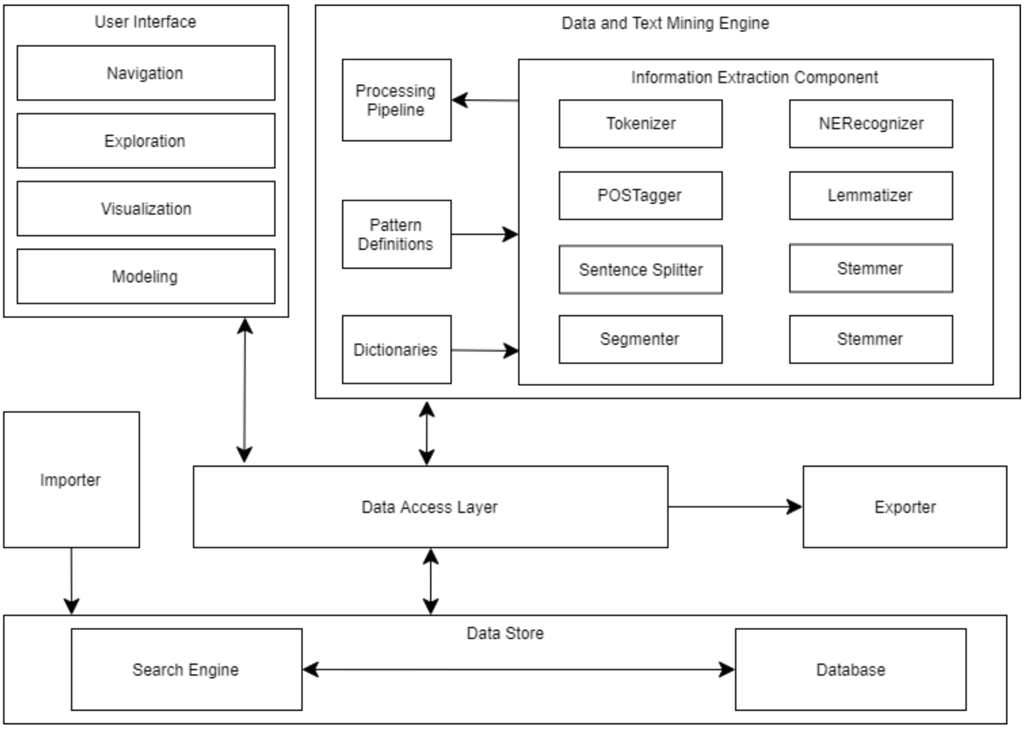

Figure 4.1: Architecture of the main components of Lexia
Source: Own illustration based on [Waltl et al., 2016, Glaser, 2017]

4.1.2 LexML Framework

LexML is a AL-microservice which extends the existing Lexia framework by a AML service. The ML functionalities are based largely on the Spark Ml implementation [Muhr, 2017]. For this thesis, LexML has been supplemented with a binary classifier that supports both Naive Bayes and logistics regression.

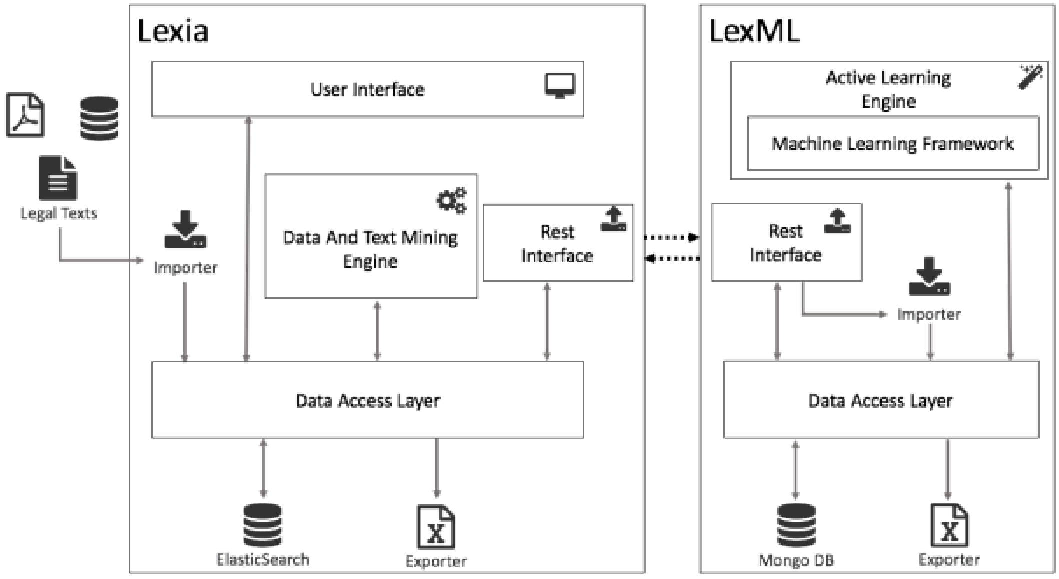

Figure 4.2: Architecture of LexML
Source: [Muhr, 2017]

5 Evaluation

5.1 Experimental Design

This section describes the experimental setup used for the binary text classification of judgments. The aim is to get an insight how different AML configurations influence the classification performance of a binary Naïve Bayes classifier. Therefore, different seed set and batch sizes are tested on the same dataset.The different test configurations are listed in table 5.1.

5.2 Data Collection and Preparation

The judgments used were imported via Lexia from an online database (Rechtssprechung im Internet [8]) of the German Federal Ministry of Justice and Consumer Protection. The judgments resulted from negotiations of the eighth Civil Panel of the Federal Court of Justice, who is specialized in law on the sale of goods, landlord and tenancy law. The imported judgments were preprocessed in Lexia by the Data and Text Mining Engine to perform a classification on sentences.

[8] https://www.rechtsprechung-im-internet.de

Table 5.1: Combination of all Evaluation Settings Used

name	query size	seed set size	learning rounds
SS_120_QS_20	20	120	120
SS_80_QS_20	20	80	120
SS_40_QS_20	20	40	120
SS_20_QS_20	20	20	120

The sentences resulting from this process were manually annotated to serve as the training set for the AML Classifier. Only sentences that are located in the tenor or the reasoning are considered, as these are the only parts of a judgment where a statement about the ineffectiveness of contractual clauses is made. Sentences have been annotated ”True” whenever the sentence establishes the connection between the contract clause and the legal reason of ineffectiveness. The evaluation was carried out on the basis of 3135 sentences, of which 71 (2.26%) sentences are annotated as "True". To counter this mismatch, the instances were weighted during the learning process. The instances of the disadvantaged class were weighted by the classifier in the learning process with a factor of 600. The division into test and training set was made in a ratio of 1/5.

5.3 Evaluation

5.3.1 Comparison of Seed Set Sizes

Figure 5.1 compares different seed set sizes based on their F1-Score for label "True". For smaller seed set sizes, the F1 score begins worsening at about 30% progress of labeling. As mentioned in section 3.2.2, random sampling of small seed sets can not guarantee that the seed set is representative. Due to the great imbalance of the classes, this effect is reinforced.

A common way to illustrate the performance of a binary classifier is the Receiver Operating Characteristics Curve (ROC Curve). Figure 5.2 shows such a ROC curve of the experiments SS80QS20 and SS20QS20. The ROC curve relates the recall with the false positive rate to confront the correctly classified positive examples with the falsely classified negative instances. A good classifier aims for the upper right corner of the ROC chart. A big advantage of the ROC graph is its resistance against an unbalanced class distribution [Davis and Goadrich, 2006, Fawcett, 2006]. Figure 5.2 also shows the superiority of the larger seed set.

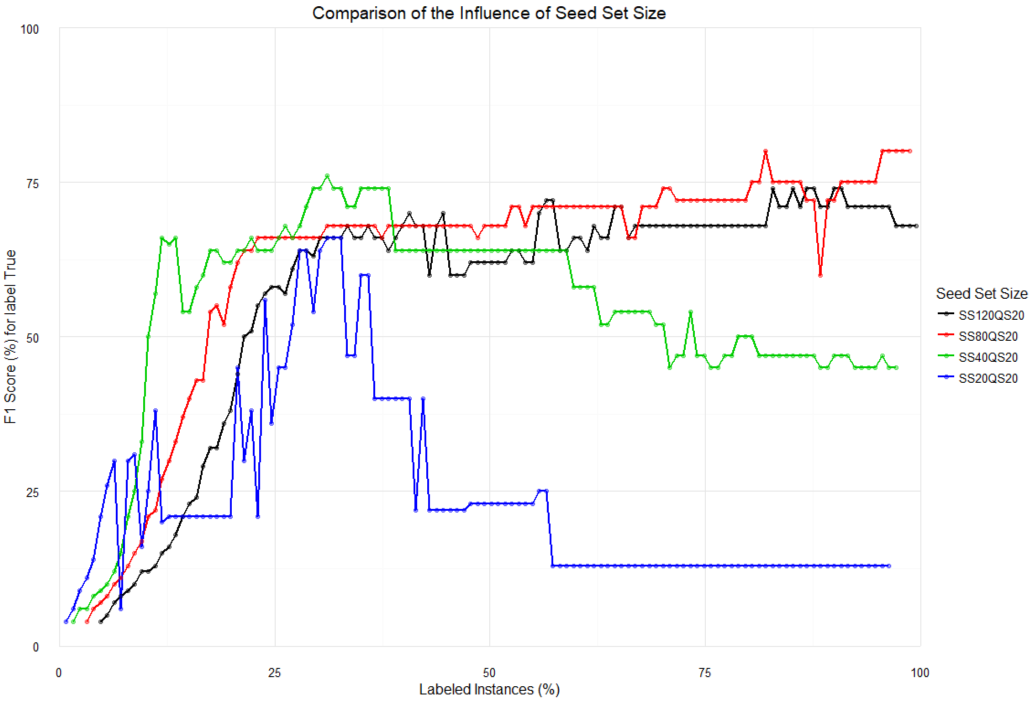

Figure 5.1: Comparison of the Influence of Seed Set Size
Source: Own illustration

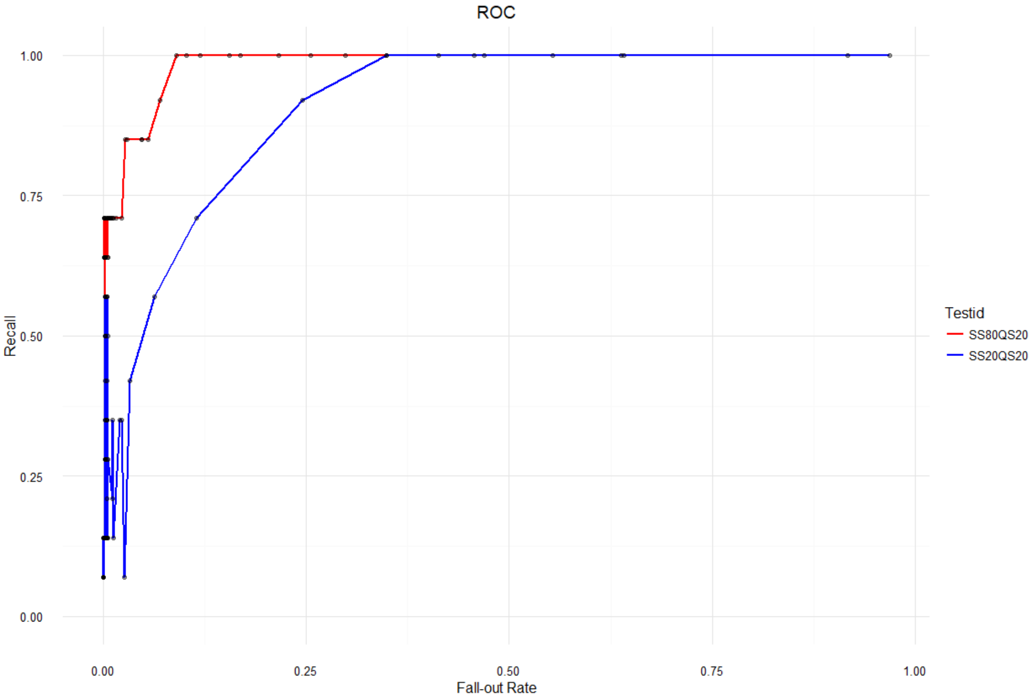

Figure 5.2: ROC-Curve of two different Seed Set Configurations
Source: Own illustration

5.3.2 Supporting the Legal Reasoning Process

Although the empirical results collected are not sufficient to make a statement based on them, literature review has revealed some possibilities in my opinion. Since the common law system mainly uses precedents for the legal reasoning process, lawyers often have to carry out extensive research. Lawyers today often use online databases equipped with simple text retrieval techniques for this research. By classifying the legal reason of the ineffectiveness of contractual clause in a judgment, more far-reaching methods can be used to recognize semantic similarities. The way in which common law lawyers work is becoming more and more relevant in the European legal area as well. Parts of German law today are already heavily influenced by case law, such as tenancy law, where many rules were created by the BGH.

6 Discussion and Reflection

In this work, only one possible use case was described, on how Legal Reasoning can be supported by binary text classification. For this purpose, various approaches to the extraction of features in the context of the legal domain were described in the literature review. The conducted classification experiment showed that binary text classification on unbalanced classes is vulnerable for a low quality seed set. This was largely caused due to the low quality of the data set. On the one hand, the data set was unbalanced on the other hand, the annotations made were possibly contradictory for the classifier. In addition to contracts, the eighth Civil Senate of the BGH also decides on the invalidity of other declarations of intent, such as Rental contract terminations and sales contract withdrawals and revocations. One possible way to improve the use case shown could be the separation of the classification into two parts. A first classifier based on a ML or a Rule-based approach would decide if the sentence has anything to do with the effectiveness of contract clauses. A second ML-based classifier would then perform the final classification task on the resulting record.

Although the literature review provides a starting point for an experimental evaluation, the available possibilities have not been fully utilized in this work. Therefore, there are many possible ways to further develop this idea.

Bibliography

[Aghila et al., 2010] Aghila, G. et al. (2010). A survey of naïve bayes machine learning approach in text document classification. *arXiv preprint arXiv:1003.1795*. 3.1.2.1

[Albalate and Minker, 2013] Albalate, A. and Minker, W. (2013). *Semi-Supervised and Unervised Machine Learning: Novel Strategies*. John Wiley & Sons. 3

[Allahyari et al., 2017] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*. 3.1.1.1, 3.1.1.1, 3.1.1.1

[Alpaydin, 2014] Alpaydin, E. (2014). *Introduction to machine learning*. MIT press. 3, 3, 3.1.1, 3.1.2.1

[Balakrishnan and Lloyd-Yemoh, 2014] Balakrishnan, V. and Lloyd-Yemoh, E. (2014). Stemming and lemmatization: a comparison of retrieval performances. *Lecture Notes on Software Engineering*, 2(3):262. 3.1.1.1

[Biagioli et al., 2005] Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., and Soria, C. (2005). Automatic semantics extraction in law documents. In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 133–140. ACM. 3.1.1.1, 3.1.1.1

[Bundesgerichtshof, 2014] Bundesgerichtshof (2014). Der bundesgerichtshof; the federal court of justice. http://www.bundesgerichtshof.de/SharedDocs/Downloads/EN/BGH/brochure.pdf?__blob=publicationFile. 2.3

[Chapelle et al., 2010] Chapelle, O., Schlkopf, B., and Zien, A. (2010). Semi-supervised learning. 3, 3, 3

[Christopher et al., 2008] Christopher, D. M., Prabhakar, R., and Hinrich, S. (2008). Introduction to information retrieval. *An Introduction To Information Retrieval*, 151(177):5. 3.1.1.2

[Cohen et al., 2004] Cohen, A. M., Bhupatiraju, R. T., and Hersh, W. R. (2004). Feature generation, feature selection, classifiers, and conceptual drift for biomedical document triage. 3.1.1.1

[Dale et al., 2000] Dale, R., Moisl, H., and Somers, H. (2000). *Handbook of natural language processing.* CRC Press. 3.1.1.1

[David and Brierley, 1978] David, R. and Brierley, J. E. (1978). *Major legal systems in the world today: an introduction to the comparative study of law.* Simon and Schuster. 2.1.1, 2.1.2

[David Reinsel, 2017] David Reinsel, John Gantz, J. R. (2017). Data age 2025: The evolution of data to life-critical. 1.1

[Davis and Goadrich, 2006] Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM. 5.3.1

[de Maat et al., 2010] de Maat, E., Krabben, K., Winkels, R., et al. (2010). Machine learning versus knowledge based classification of legal texts. In *JURIX*, pages 87–96. 3.1.1.1, 3.1.1.1

[Dligach and Palmer, 2011] Dligach, D. and Palmer, M. (2011). Good seed makes a good crop: accelerating active learning using language modeling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 6–10. Association for Computational Linguistics. 3.2.2

[Domingos and Pazzani, 1997] Domingos, P. and Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130. 3.1.2.1

[Duda et al., 2002] Duda, R. O., Hart, P. E., and Stork, D. G. (2002). *Pattern classification*. John Wiley & Sons. 3, 3

[Ellsworth, 2005] Ellsworth, P. C. (2005). Legal reasoning. 2.2

[Fawcett, 2006] Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874. 5.3.1

[Fellmann et al., 1968] Fellmann, D., Jenks, C. W., and Sills, D. L. (1968). Adjudication. *International encyclopedia of the social sciences*. 2.2

[Frakes, 1992] Frakes, W. B. (1992). Stemming algorithms. 3.1.1.1

[Francesconi and Passerini, 2007] Francesconi, E. and Passerini, A. (2007). Automatic classification of provisions in legislative texts. *Artificial Intelligence and Law*, 15(1):1–17. 3.1.1.2

[Friedman et al., 1997] Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, 29(2-3):131–163. 3.1.2.1

[Gabrilovich and Markovitch, 2005] Gabrilovich, E. and Markovitch, S. (2005). Feature generation for text categorization using world knowledge. In *IJCAI*, volume 5, pages 1048–1053. 3.1.1.1

[Gabrys and Petrakieva, 2004] Gabrys, B. and Petrakieva, L. (2004). Combining labelled and unlabelled data in the design of pattern classification systems. *International journal of approximate reasoning*, 35(3):251–273. 3

[Glaser, 2017] Glaser, I. (2017). Semantic analysis and structuring of german legal documents using named entity recognition and disambiguation. Master's thesis, Department of Informatics, Technical University of Munich. 3.1.1.1, 4.1

[Glaser et al., 2018] Glaser, I., Waltl, B., and Matthes, F. (2018). Named entity recognition, extraction, and linking in german legal contracts. 3.1.1.1

[Gonçalves and Quaresma, 2005] Gonçalves, T. and Quaresma, P. (2005). Is linguistic information relevant for the classification of legal texts? In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 168–176. ACM. 3.1.1.1, 3.1.1.1

[Herman, 2008] Herman, H. J. (2008). Legal reasoning. 2.2

[Hofmann, 2018] Hofmann, R. (2018). Aufbau des urteils in zivilsachen. 2.1, 2.3.1, 2.3.1, 2.3.1

[Joachims, 1996] Joachims, T. (1996). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science. 3.1.1.2

[Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer. 3.1, 3.1.1

[Kannan and Gurusamy, 2014] Kannan, S. and Gurusamy, V. (2014). Preprocessing techniques for text mining. 3.1.1.1, 3.1.1.1, 3.1.1.1

[Khalid et al., 2014] Khalid, S., Khalil, T., and Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. In *Science and Information Conference (SAI), 2014*, pages 372–378. IEEE. 3.1.1

[Khan et al., 2010] Khan, A., Baharudin, B., Lee, L. H., and Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20. 1.1, 3.1, 3.1.1, 3.1.2.1

[law and civil law traditions, 2006] law, C. and civil law traditions, u. (2006). The common law and civil law traditions. https://www.law.berkeley.edu/library/robbins/CommonLawCivilLawTraditions.html. 2.1.2, 2.2

[Levi, 1948] Levi, E. H. (1948). An introduction to legal reasoning. *The University of Chicago Law Review*, 15(3):501–574. 2.1.2, 2.2

[Lewis, 1992] Lewis, D. D. (1992). Feature selection and feature extraction for text categorization. In *Proceedings of the workshop on Speech and Natural Language*, pages 212–217. Association for Computational Linguistics. 3.1.1.1

[Lewis and Gale, 1994] Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc. 3.2.3.1

[Lexalyze, nd] Lexalyze ([n.d.]). Whitepaper: Lexia - legal information analysis, exploration, and reasoning platform. 4.1.1

[Méndez et al., 2005] Méndez, J. R., Iglesias, E. L., Fdez-Riverola, F., Díaz, F., and Corchado, J. M. (2005). Tokenising, stemming and stopword removal on anti-spam filtering domain. In *Conference of the Spanish Association for Artificial Intelligence*, pages 449–458. Springer. 3.1.1.1

[Mitchell, 1997] Mitchell, T. (1997). Machine learning. wcb. 3, 3.1.2.1

[Mitchell, 2006] Mitchell, T. M. (2006). *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning Department. 1.1, 3

[Motoda and Liu, 2002] Motoda, H. and Liu, H. (2002). Feature selection, extraction and construction. *Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol*, 5:67–72. 3.1.1.1

[Muhr, 2017] Muhr, J. (2017). Design, prototypical implementation, and evaluation of an active machine learning service in the context of legal text classification. Master's thesis, Department of Informatics, Technical University of Munich. 3.1.2.1, 3.2, 4.1.2, 4.2

[PAK and GUNAL, 2017] PAK, M. Y. and GUNAL, S. (2017). The impact of text representation and preprocessing on author identification. *Anadolu Üniversitesi Bilim Ve Teknoloji Dergisi A-Uygulamalı Bilimler ve Mühendislik*, 18(1):218–224. 3.1

[Paul and Baron, 2006] Paul, G. L. and Baron, J. R. (2006). Information inflation: Can the legal system adapt. *Rich. JL & Tech.*, 13:1. 1.1

[Pomikálek and Rehurek, 2007] Pomikálek, J. and Rehurek, R. (2007). The influence of preprocessing parameters on text categorization. *International Journal of Applied Science, Engineering and Technology*, 1:430–434. 3.1.1.1

[Raghavan et al., 2004] Raghavan, P., Amer-Yahia, S., and Gravano, L. (2004). Structure in text: Extraction and exploitation. In *Proceeding of the 7th international Workshop on the Web and Databases (WebDB), ACM SIGMOD/PODS*. 1.1

[Röhl and Röhl, 2008] Röhl, K. F. and Röhl, H. C. (2008). Allgemeine rechtslehre: ein lehrbuch. 2.1, 2.1.1, 2.1.2, 2.2

[Schölkopf and Smola, 2002] Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press. 3.1.1.1

[Schütze et al., 2008] Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press. 3.1.1.2

[Schütze et al., 2006] Schütze, H., Velipasaoglu, E., and Pedersen, J. O. (2006). Performance thresholding in practical text classification. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 662–671. ACM. 3.2.2

[Schweighofer et al., 2001] Schweighofer, E., Rauber, A., and Dittenbach, M. (2001). Automatic text representation, classification and labeling in european law. In *Proceedings of the 8th international conference on Artificial intelligence and law*, pages 78–87. ACM. 3.1.1.2

[Scott and Matwin, 1999] Scott, S. and Matwin, S. (1999). Feature engineering for text classification. In *ICML*, volume 99, pages 379–388. 3.1.1.1

[Settles, 2011] Settles, B. (2011). From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, pages 1–18. 3.2.4

[Settles, 2012] Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114. 3.2, 3.2.1, 3.2.1.1, 3.2.1.2, 3.2.1.3, 3.2.2

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423. 3.2.3.1

[Tetley, 1999] Tetley, W. (1999). Mixed jurisdictions: Common law v. civil law (codified and uncodified). *La. L. Rev.*, 60:677. 2.1, 2.1.1, 2.1.2

[Tomanek et al., 2009] Tomanek, K., Laws, F., Hahn, U., and Schütze, H. (2009). On proper unit selection in active learning: co-selection effects for named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 9–17. Association for Computational Linguistics. 3.2.2

[Turtle, 1995] Turtle, H. (1995). Text retrieval in the legal world. *Artificial Intelligence and Law*, 3(1-2):5–54. 3.1.1.1

[van den Bosch, 2017] van den Bosch, S. (2017). Automatic feature generation and selection in predictive analytics solutions. Master's thesis, Faculty of Science, Radboud University. 3.1.1, 3.1.1.1

[Walter, 2008] Walter, S. (2008). Linguistic description and automatic extraction of definitions from german court decisions. In *LREC*. 3.1.1.1

[Walter and Pinkal, 2006] Walter, S. and Pinkal, M. (2006). Automatic extraction of definitions from german court decisions. In *Proceedings of the workshop on information extraction beyond the document*, pages 20–28. Association for Computational Linguistics. 3.1.1.1

[Waltl et al., 2016] Waltl, B., Matthes, F., Waltl, T., and Grass, T. (2016). Lexia: A data science environment for semantic analysis of german legal texts. *Jusletter IT*. 4.1.1, 4.1

[Witten et al., 2016] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann. 3, 3.1.2.1