Classifying law area of Dutch legal documents

Guido C. A. Zuidhof [a,1]

[a]Radboud University Nijmegen, The Netherlands

Legal documents are part of one or more areas of law (also called practice areas). Often this law area meta-data is not present in a structured manner or not present at all. In this paper we describe a method for automatic classification of the law areas of a legal document. We apply a supervised learning approach in which two types of binary classifiers, a naïve bayes model and a logistic regression model, are trained in a one-versus-rest setting. A dataset of 357,412 labeled legal documents was used for evaluating this method. A recall, precision and F-score of greater than 0.96 is achieved, showing that this automatic classification is indeed possible.

multilabel text classification | law topology | legal documents

Law can be divided into a topology of law areas (also referred to as *practice areas*). Examples of law areas are *corporate law*, *criminal law* and *tax law*. Legal documents often concern a small subset of these law areas. For instance, a document containing a court verdict on tax evasion may be assigned to the *tax law* as well as *criminal law* areas of law.

Many legal documents are created and indexed each year in the Netherlands, to provide an indication, 33,454 were indexed in the public governmental legal document repository *rechtspraak.nl*[1] in 2015 alone. There is no unified repository where these documents are published, and despite standardization efforts there is as of yet no single structure for legal documents and their meta-data in wide use[1][2]. A lawyer or other legal expert often investigates prior verdicts and other legal documents when dealing with a case. Specialized search engines offer a single point of entry for these various sources of legal documents.

The area of law meta-data is a valuable filter term when searching for legal documents. This meta-data is however not always present in a structured manner which a computer can parse or present at all. Assigning the correct area(s) of law to legal documents currently happens manually and is a laborious task.

Opsomer et al., in a 2009 paper, argue that it is necessary for legislators to annotate legislation with meta-data because it helps allow one to retrieve all legislation applicable to a certain issue[3].

They argue that *"While text classification is a mature domain, with very good results in e.g. the automated classification of news articles in classes such as "Politics" or "Sport", achieving accuracy rates of 90%+, the approach does not work very well for legal topics. This is mainly because legal concepts are too abstract to be easily learned by a classifier. In the past, there was always optimism that this problem would be overcome, but now we have indications that this will not be the case in the near feature"*.

To back up this statement, they refer to their own article titled *"Automated classification of legal text will never work"*, which was to be written and published, but unfortunately never was.

We see no reason to believe this sentiment. In this study we will investigate the possibility of automatically assigning legal documents their representative law areas, based solely on the textual content of the document.

Aside from labeling the documents the best we can, we also want to find a model which is not entirely a black box. As such, we train models which allow for probabilistic interpretation of the results. Two classifiers which allow for probabilistic interpretation of the classifications will be compared, namely a *Multinomial Naïve Bayes* model and a *Logistic Regression* model.

The research questions of this study read:

Is automatic classification of the law areas a legal document applies to possible with sufficient accuracy?

Is a more complex logistic regression model better suited than a multinomial naïve bayes model for this classification problem?

For the first research question it is hard to determine what a sufficient accuracy would be. We opt to take the (fairly arbitrary) measure of 90% accuracy and higher mentioned by Opsomer et al..

We hypothesize that this accuracy is attainable for legal documents. Although legal concepts may indeed be too abstract to easily be learned by a classifier, this thorough understanding is not necessary for classification to a law topology. Furthermore, we expect that the formal, exact language (sometimes called *legalese*) which is typical for legal documents will make for easier classifications as the vocabulary may be smaller.

Significance

Legal documents often lack structure of meta-data, and meta-data that is often missing is the area of law the document belongs to. Representative law area meta-data is important in legal search as it can be used as a filter option.

Currently, assigning legal documents to one or more law areas is a manual and time-consuming process.

[1]Artifical Intelligence student at Radboud University Nijmegen, student number: 4160703. Correspondence can be addressed to guido.zuidhof@student.ru.nl

This paper was written as part of the final project for the Text Mining course (LET-REMA-LCEX06-2015-PER1-V) taught at the Radboud University, Nijmegen.

[1]`http://www.rechtspraak.nl`

1. Related work

Extensive research is done in the area of legal information retrieval [4]. Most studies however, deal with the retrieval part in particular or with automatically classifying or recognizing certain concepts or language constructs within documents.

Opsomer et al. attempted to classify Belgian laws (written in Dutch) to a hierarchy of subjects by using a *support vector machine (SVM)* classifier[5]. They exploited some domain knowledge to improve their classification performance (such as recognizing certain patterns such as references to legal sources and using this as a feature).

For the top-level subjects in this hierarchy they achieved a 63.6% accuracy rate, and much lower for the lower level subjects. They concluded that it is not possible to automatically assign documents to this hierarchy accurately enough to be of practical use. Note that this is a multiclass classification problem as opposed to the multilabel classification problem posed in this study.

Maat et al. compared machine learning versus knowledge based classification of Dutch legal texts[6]. This classification was done at a more granular level (sentences as opposed to full documents) and aimed to classify sentences into categories such as definition, repeal, obligation etc..

Accuracy scores higher than 90% with both the machine learning approach as well as the knowledge based pattern recognizer were achieved. They conclude that the machine learning approach works just as well as the knowledge based method, however, it may generalize less well to different kinds of text. Also, they prefer the pattern based approach as it is less of a black box and its predictions can be used for further modeling the structure of these sentences.

An exciting approach by Goncalves et al. which predicts classes of legal documents involves training an SVM classifier for legal texts in different languages individually[7], which are then combined to form a single multilingual classifier which outperforms the single language classifiers.

2. Method

In this section we describe the dataset and how we use it to create a predictive model for labeling legal documents with their corresponding areas of law. Also, we will describe methods of evaluating the model. We will follow the standard steps involved in data mining: *selection, preprocessing, transformation, classification*, and finally *evaluation*.

2.1 Dataset. The dataset consists 357,819 legal documents labeled with their representative law areas (7.76GB). These were scraped from the Dutch governmental website *rechtspraak.nl* by legal search machine company *Legal Intelligence*[2]. The documents were provided in a proprietary structured (XML) format.

407 of these documents were not in the above-mentioned XML format, but were instead the source HTML page or document PDF. These documents were removed from the dataset, leaving us with 357,412 valid documents.

The most frequent label, *Bestuursrecht (Corporate law)*, is a label of half the documents, whereas two-thirds of the labels describe less than a percentage of all documents. See table 1. These varying frequencies of labels make for a very unbalanced dataset.

Law area label	*Frequency*
Bestuursrecht	192074
Civiel recht	102906
Strafrecht	65293
Socialezekerheidsrecht	51088
Belastingrecht	40590
Vreemdelingenrecht	15129
Personen-en familierecht	14832
Omgevingsrecht	9765
Ambtenarenrecht	7429
Insolventierecht	3525
Bestuursstrafrecht	1130
Arbeidsrecht	743
Verbintenissenrecht	705
Ondernemingsrecht	502
Burgerlijk procesrecht	321
Bestuursprocesrecht	269
Europees bestuursrecht	230
Materieel strafrecht	207
Intellectueel-eigendomsrecht	113
Aanbestedingsrecht	97
Internationaal publiekrecht	75
Internationaal privaatrecht	72
Strafprocesrecht	58
Goederenrecht	35
Mededingingsrecht	33
Europees civiel recht	20
Penitentiair strafrecht	5
Mensenrechten	4
Internationaal strafrecht	3
Europees strafrecht	3
Volkenrecht	1

Table 1: Labels found in the dataset and their respective frequencies. Note that a single document may have multiple labels.

Most documents are either assigned one or two labels (21090 and 143427 respectively) with 3168 documents assigned three labels, 26 documents assigned four labels and a single document assigned 5 labels.

Label topology. The labels are part of a law topology, where some labels always co-occur with their parent labels. This topology is the one used by *rechtspraak.nl*, see table 2.

Bestuursrecht	***Civiel recht***
— Ambtenarenrecht	— Aanbestedingsrecht
— Belastingrecht	— Arbeidsrecht
— Bestuursprocesrecht	— Burgerlijk procesrecht
— Europees bestuursrecht	— Goederenrecht
— Mededingingsrecht	— Intellectueel-eigendomsrecht
— Omgevingsrecht	— Internationaal privaatrecht
— Socialezekerheidsrecht	— Mededingingsrecht
— Vreemdelingenrecht	— Ondernemingsrecht
	— Personen- en familierecht
Strafrecht	— Verbintenissenrecht
— Europees strafrecht	
— Internationaal strafrecht	***Internationaal publiekrecht***
— Materiaal strafrecht	— Mensenrechten
— Penitentiair strafrecht	— Volkenrecht
— Strafprocesrecht	

Table 2: Law topology as used in the legal document repository rechtspraak.nl.

[2] `https://www.legalintelligence.com/`

If the *subclass* label *Ambtenarenrecht* is present, it's *superclass* label *Bestuursrecht* will always be present as well. Note that it is also possible for only the *Bestuursrecht* label to be present.

Remark 1. *There is one label, Mededingingsrecht, that is part of two superclass labels (Bestuursrecht and Civiel recht). For this label the above does not hold.*

2.2 Selection. The legal documents can be divided into *verdicts* and *conclusions*. Although this subdivision contains information about the area of law the document is part of, we wish to create a model which solely works on the plain text found in the document, and as such both types were selected.

2.3 Preprocessing. The documents are provided in an XML format, whereas we are interested in only the textual content of the text and the law area labels. Also, methods of dimensionality reduction are applied to the plain text.

Plaintext extraction. The first step in the preprocessing of the XML files is the extraction of the labels and the text data. This is performed by parsing the XML tree and extracting the relevant parts (the verdict or conclusion section).

Frog NLP analysis. Next, the plain text files are processed by *Frog*[3][8]. Frog is an integration of open source *NLP (natural language processing)* modules for Dutch text.

Frog is capable of many advanced NLP tasks, such as *named entity recognition*. This analysis, however, takes a long time to perform. As such, we only enable the tokenization, lemma and *part-of-speech (PoS)* features. See table 3 for example output with these settings.

#	Token	Lemma	Part-of-speech tag
1	Zij	zij	VNW(pers,pron,nomin,vol,3p,mv)
2	hebben	hebben	WW(pv,tgw,mv)
3	gesteld	stellen	WW(vd,vrij,zonder)
4	dat	dat	VG(onder)
5	het	het	LID(bep,stan,evon)
6	huwelijk	huwelijk	N(soort,ev,basis,onz,stan)
7	duurzaam	duurzaam	ADJ
8	is	zijn	WW(pv,tgw,ev)
9	ontwricht	ontwrichten	WW(vd,vrij,zonder)
10	.	.	LET()

Table 3: Example frog output with only tokenization, lemma and PoS features enabled. The first column indicates the token number in the current sentence. A confidence value of the PoS tag is also present in the output, this column was omitted for space purposes.

Tokenization. A piece of text has to be cut into pieces in order to transform it into a feature vector. This can be done in many different ways, such as on a character or word basis. Here, we opt for tokenization on a word level. This was performed by Frog, which uses the rule-based tokenizer Ucto[4]. It separates punctuation, words, symbols and sentences.

Lemmatization. Lemmatization is the act of reducing words to their base form known as the lemma. See the table 4 for examples of tokens and their lemma. Lemmatization reduces the dimension of the resulting feature vectors as different words are mapped to the same base form. We apply this to all words in the dataset.

Original token	Lemma
van	van
blijkt	blijken
tevens	tevens
minderjarige	minderjarig
ingekomen	inkomen
De	de
is	zijn
partijen	partij
;	;
wetboek	wetboek

Table 4: Examples of tokens and their respective lemmas.

PoS filtering. Words in a text have a certain type. A word can for instance be a verb with a certain conjugation or a noun. Often, to determine this the context the word is in is required as well. For example, the word *bait* can be both a noun or verb. We use Frog to automatically determine the part-of-speech tags of the tokens in the legal documents.

Frog uses *mbt*, which is a is a memory-based tagger-generator and tagger in one[5] for part-of-speech tagging. According to the documentation Frog assigns PoS tags based on the *CGN (Corpus Gesproken Nederlands)* tag set[9][10]. Twelve main tag groups are defined in this tag set, which can be found in table 5.

Remark 2. *Two tags appear in the frog output which are not part of the CGN tag set, namely SPEC(afkorting) and SPEC(symbool). This leads me to believe it is instead tagged according to the D-coi tag set[11], which is a superset of the CGN tagset with the addition of these two tags.*

PoS tag	Token type	Token type translated
ADJ	Adjectief	Adjective
BW	Bijwoord	Adverb
LET	Letterteken	Punctuation
LID	Lidwoord	Determiner
N	Naamwoord	Noun
SPEC	Speciaal	Special
TSW	Tussenwoord	Interjection
TW	Telwoord	Cardinal/ordinal
VG	Voegwoord	Conjunction
VNW	Voornaamwoord	Pronoun
VZ	Voorzetsel	Preposition
WW	Werkwoord	Verb

Table 5: Part-of-speech main tag groups assigned by Frog.

We filter certain tokens by their assumed part of speech category. After empirical investigation we found that removing certain types of tokens improves classification accuracy. This is a type of feature selection.

The types we remove are *LET* (punctuation), *TW* (cardinals and ordinals), *LID* (determiners), *VG* (conjunctions), *VZ* (prepositions), and *SPEC(symb)* (symbols).

[3] `https://languagemachines.github.io/frog/`
[4] `https://languagemachines.github.io/ucto/`
[5] `https://languagemachines.github.io/mbt/`

2.4 Transformation. From the selected and preprocessed terms we now have to construct a feature vector. One can opt for simply using the term counts as feature vector. However, the presence of some terms may hold more information than others, and thus using different weights for terms can improve performance [12].

The most often used document representation is *tf−idf* [12]. This is bag-of-words representation which is calculated by multiplying the *term frequency* with the *inverse document frequency* of a term t in document d as follows:

$$tf - idf_{t,d} \;=\; tf_{t,d} \times idf_t \;=\; tf_{t,d} \times log(\frac{N}{df_t}) \qquad (1)$$

with $tf_{t,d}$ the frequency of term t in document d, df_t the amount of documents term t occurs in, and N the amount of documents.

More elaborate weighting schemes exist [13], some even specifically tailored to multilabel text classification [14]. These are, however, outside of the scope of this study.

2.5 Classification. The dataset is split into a training set consisting of 50% of the documents and a test set containing the other half. We are using a *one-versus-rest* classification scheme. This means that we are training a binary classifier for each label, and to come to a multilabel prediction, we combine the output of these classifiers.

We are comparing two types of binary classifiers, namely a *Multinomial Naïve Bayes* model and a *Logistic Regression* model. The *scikit-learn*[6] [15] implementation was used of both models.

Here a short introduction is given for either model, and the use thereof is motivated. Afterwards, we describe two additional post-processing steps which are performed on the classifications generated by these models.

Multinomial Naïve Bayes. Multinomial Naïve Bayes is a simple model which can be considered the baseline for many classification problems, and in particular text classification problems. It determines the probability of a sample belonging to either the positive or the negative class by looking at the frequencies of that term occuring in either instance.

Logistic Regression. Linear support vector machines present the state-of-the-art in text classification[16]. It works by determining the best linear decision boundary which separates the data in the true and false class. There is however no probabilistic interpretation of the classifications, which is a prerequisite for this study. As such, we use a logistic regression model which also which attempts to determine a decision boundary which best separates the two classes. This linear regression is used to estimate the probability of a sample being part of either of the two classes. Both have very similar performance and scale well to large datasets.

Enforcing at least one label. Because of the one-versus-rest nature of the classifications, it is possible that not a single label is assigned to a document. We know that every document should have at least one label. If a document is not assigned any label, we assign the most likely label - the one with the highest probability - instead.

Exploiting the topology. As the law areas of a document follow a topology, where subclasses are always part of its superclass. For instance, if the label *"Europees civiel recht" (European civil law)* is present, the label *"Civiel recht" (Civil law)* is always present too.

We infer this topology from the data (the train set), and can come up with implications such as:

$$\textit{Europees civiel recht} \implies \textit{Civiel recht}. \qquad (2)$$

To generate these *topology rules* we consider all pairs of labels. If a label p always coincides with label q we infer that q is the superclass of p, or more formally:

With all training set labellings Y and the set of possible labels L:

$$\begin{aligned} &\forall p \in L, \forall q \in L, \\ &\quad \forall y \in Y, \\ &\qquad (p \in y \implies q \in y) \\ &\qquad \implies \\ &\qquad (p \implies q) \end{aligned} \qquad (3)$$

As a final step, we apply these rules to the classifications.

Remark 3. *Note that not the whole topology may be inferred this way, and also faulty implications may be inferred. Even if all implications are correct, applying these rules may still lead to a worse classification when the subclass was incorrectly assigned (as it will also be assigned the superclass).*

2.6 Evaluation. Our goal is to create a classifier which labels documents according to their respective law areas. To evaluate the performance, we make use of certain metrics.

Precision, recall, F-score. Often, accuracy is used as the go-to metric for classifier evaluation. The problem here is however more akin to information retrieval. If 99% of documents are not described by a certain label, an accuracy of .99 could be achieved by never assigning the label.

Better suited are *precision* and *recall* metrics. Precision captures the ability of the classifier to not assign a label to a document that should not have this label. Recall is the ability to label those which should be assigned a certain label.

Often a trade-off between precision and recall has to be made. The F_β-score *"measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision"* [17], as such we can use it as a weighted combination of precision and recall.

Remark 4. *If one uses the classifications of this model as a filter for a search engine, it may very well be that one values recall more highly than precision. We do not want to make this assumption however, and thus will use $\beta = 1$ henceforth, which is the harmonic mean of precision and recall. This is also called the F_1-score.*

With A the predicted *(sample, label)* pairs and B the true *(sample, label)* pairs, we define precision as

[6] `https://scikit-learn.org/`

$$P(A,B) := \frac{|A \cap B|}{|A|}, \tag{4}$$

recall as

$$R(A,B) := \frac{|A \cap B|}{|B|}, \tag{5}$$

and F_β-score as

$$F_\beta(A,B) := (1+\beta^2) \cdot \frac{P(A,B) \cdot R(A,B)}{\beta^2 P(A,B) + R(A,B)}. \tag{6}$$

Remark 5. *Note that when A or B are empty in the above formulas, $P(A,B)$ and $R(A,B)$ respectively are ill defined (divide by zero). We handle this by defining $P(\emptyset, B) := 0$ and $R(A,\emptyset) := 0$.*

The above precision, recall and F_β-score metrics are for a single classification problem. We are solving a multilabel classification problem and as such have to combine these values of every individual label's classifier.

For this we use the *weighted average* precision, recall and F_β-score metrics. The precision, recall and F_β-score of every binary classifier are weighted by the amount of true samples for that label.

With $\hat{y}$ the true *(sample, label)* pairs, y the predicted *(sample, label)* pairs, y_l the subset of y with label l, and L the set of labels, we define weighted precision as

$$\frac{1}{\sum_{l \in L} |\hat{y}_l|} \cdot \sum_{l \in L} |\hat{y}_l| \cdot P(y_l, \hat{y}_l), \tag{7}$$

weighted recall as

$$\frac{1}{\sum_{l \in L} |\hat{y}_l|} \cdot \sum_{l \in L} |\hat{y}_l| \cdot R(y_l, \hat{y}_l), \tag{8}$$

and weighted F_β-score as

$$\frac{1}{\sum_{l \in L} |\hat{y}_l|} \cdot \sum_{l \in L} |\hat{y}_l| \cdot F_\beta(y_l, \hat{y}_l). \tag{9}$$

Loss metrics. Finally, we introduce two metrics more akin to accuracy.

Zero-one classification loss is the fraction of documents that were not perfectly labeled. It is given by

$$1 - \frac{1}{|D|} \sum_{d \in D} \mathbb{1}(\textit{Predicted}_d = \textit{True}_d), \tag{10}$$

where D is the set of documents, $\textit{Predicted}_d$ and $\textit{True}_d$ are the predicted and true labels for document d as binary string. $\mathbb{1}$ is the indicator function, which in this case is equal to 1 when all values in the binary string are equal, and 0 otherwise.

Zero-one classification loss ignores the notion that the labeling of a document can be partially correct. Schapire et al. proposed the use of *Hamming loss* for multilabel classification problems[18]. Hamming loss is the fraction of labels that are incorrectly predicted. This is a more forgiving metric, it penalizes on individual labels. In our one-versus-rest classification scheme, one can see it as the the fraction of incorrect binary decisions of all binary classifiers combined. It is upper-bounded by the zero-one classification loss. It is given by:

$$\frac{1}{|D|} \sum_{d \in D} \frac{\textit{Predicted}_d \oplus \textit{True}_d}{|L|} \tag{11}$$

where D is the set of documents, L the set of possible labels, $\textit{Predicted}_d$ and $\textit{True}_d$ are the predicted and true labels for document d as binary string. $\oplus$ is the *XOR* operator.

These metrics are both loss functions that are between 0 and 1, where lower is better.

Comparison to human classification. A subset of 24 randomly selected documents was manually labeled by two legal experts. This allows us to get an indication of how the automatic classification method compares to manual annotation. Care was taken to not have these 24 documents as part of the training set.

4. Results

We tested the model by predicting labels of the test set of documents (which makes up 50% of all documents), and comparing this to the actual labels. In this section we present the results.

Classifier	*Zero-one*	*HL*	*Precision*	*Recall*	*F-Score*
Logistic Regression					
With postprocessing	0.0694	0.0025	0.9748	0.9628	0.9685
No postprocessing	0.0709	0.0025	0.9754	0.9617	0.9682
Multinomial NB					
With postprocessing	0.3400	0.0124	0.8969	0.7542	0.7870
No postprocessing	0.3902	0.0135	0.9097	0.7147	0.7707

Table 6: Classifier performance on test set consisting of 178706 documents. HL is the Hamming Loss metric.

4.1 Classifier performance. The *Logistic Regression* classifier performed better than the *Multinomial Naïve Bayes* classifier in all metrics (see table 6).

It appears that when we look at the precision, recall and F-scores of the individual labels in table 7 and 8, the Naïve bayes model is much more conservative when it comes to assigning labels. It assigns fewer labels, and seems to only ever assign labels which occur the most often (the top 5 more or less). The unbalanced dataset seems to pose a problem for the Naïve Bayes model. This appears to be less problematic for the Logistic Regression model, although it also nearly never assigns the very infrequently occuring labels (those describing less than .1% of all documents).

4.2 Post-processing performance increase. After making predictions with the model, we apply post-processing steps which make use of domain knowledge. More specifically, in the first step we made sure there was at least label assigned to each document, and second we enforced the inferred *topology rules* (see section 2.5).

This resulted in 13501 labels added in the first step, and 2 labels in the second step added to the classifications of the Multinomial Naïve Bayes model. 371 and 87 labels for the first and

Label	Precision	Recall	F-Score	Support
Bestuursrecht	0.9960	0.9960	0.9960	96158
Civiel recht	0.9927	0.9816	0.9871	51478
Strafrecht	0.9920	0.9887	0.9903	32505
Socialezekerheidsrecht	0.9585	0.9536	0.9561	25582
Belastingrecht	0.9927	0.9794	0.9860	20278
Vreemdelingenrecht	0.9602	0.9322	0.9460	7535
Personen- en familierecht	0.8891	0.8615	0.8751	7437
Omgevingsrecht	0.8284	0.6723	0.7422	4891
Ambtenarenrecht	0.9286	0.8209	0.8714	3707
Insolventierecht	0.8185	0.6840	0.7452	1747
Bestuursstrafrecht	0.9484	0.8342	0.8877	573
Arbeidsrecht	0	0	0	377
Verbintenissenrecht	0	0	0	349
Ondernemingsrecht	0.7544	0.3660	0.4928	235
Burgerlijk procesrecht	0	0	0	164
Bestuursprocesrecht	0	0	0	140
Europees bestuursrecht	0	0	0	118
Materieel strafrecht	0	0	0	108
Intellectueel-eigendomsrecht	0	0	0	63
Aanbestedingsrecht	0	0	0	50
Internationaal publiekrecht	0	0	0	39
Internationaal privaatrecht	0	0	0	28
Strafprocesrecht	0	0	0	25
Goederenrecht	0	0	0	20
Mededingingsrecht	0	0	0	17
Europees civiel recht	0	0	0	12
Penitentiair strafrecht	0	0	0	2
Mensenrechten	0	0	0	3
Internationaal strafrecht	0	0	0	3
Europees strafrecht	0	0	0	0
Volkenrecht	0	0	0	1
avg / total	**0.9748**	**0.9628**	**0.9685**	**253645**

Table 7: Performance metrics of one-versus-rest Linear Logistic Regression classifier on the test set.

Label	Precision	Recall	F-Score	Support
Bestuursrecht	0.9463	0.9961	0.9706	96158
Civiel recht	0.9950	0.8910	0.9401	51478
Strafrecht	0.9902	0.9580	0.9738	32505
Socialezekerheidsrecht	0.9891	0.4462	0.6150	25582
Belastingrecht	0.9985	0.3488	0.5170	20278
Vreemdelingenrecht	1.0000	0.0012	0.0024	7535
Personen- en familierecht	0	0	0	7437
Omgevingsrecht	0	0	0	4891
Ambtenarenrecht	0	0	0	3707
Insolventierecht	0	0	0	1747
Bestuursstrafrecht	0	0	0	573
Arbeidsrecht	0	0	0	377
Verbintenissenrecht	0	0	0	349
Ondernemingsrecht	0	0	0	235
Burgerlijk procesrecht	0	0	0	164
Bestuursprocesrecht	0	0	0	140
Europees bestuursrecht	0	0	0	118
Materieel strafrecht	0	0	0	108
Intellectueel-eigendomsrecht	0	0	0	63
Aanbestedingsrecht	0	0	0	50
Internationaal publiekrecht	0	0	0	39
Internationaal privaatrecht	0	0	0	28
Strafprocesrecht	0	0	0	25
Goederenrecht	0	0	0	20
Mededingingsrecht	0	0	0	17
Europees civiel recht	0	0	0	12
Penitentiair strafrecht	0	0	0	2
Mensenrechten	0	0	0	3
Internationaal strafrecht	0	0	0	3
Europees strafrecht	0	0	0	0
Volkenrecht	0	0	0	1
avg / total	**0.8969**	**0.7542**	**0.7870**	**253645**

Table 8: Performance metrics of one-versus-rest Multinomial Naïve Bayes classifier on the test set.

second step respectively were added to the classifications of the logistic regression model.

For both models this improved the predictive accuracy of the model (see table 6). This improvement was larger for the multinomial naive bayes model, which is to be expected given the large amount of labels that were added in the first step. The model was very conservative in its predictions, i.e. not many labels were assigned. Having no labels is certainly wrong, so ensuring there is at least one label (with the highest probability) is a fairly surefire way to improve performance.

4.3 Comparison with human classifications. A subset of 24 documents were labeled by two legal experts. See table 9. The trained predictive model outperformed the legal experts in all metrics, however, this may be an unfair judgement. The legal experts were not aware of the topology of law they were to follow. For instance, if the more specific *Burgerlijk procesrecht (Civil procedural law)* label was assigned, they did not know it should also be assigned the superclass *Civiel recht (Civil law)*.

If we add these superclass labels for all assigned subclass labels, we find that the performance is very similar to the predictive model. In the table look at the *"Legal expert TE"* row for the performance on the *topology enforced* version of their classifications.

Classifier	Zero-one	HL	Precision	Recall	F-Score
LR model	0.4167	0.0174	0.7674	0.6976	0.7224
Legal expert	0.7083	0.0403	0.7578	0.4651	0.5377
Legal expert TE	0.4583	0.0282	0.7164	0.7209	0.7107

Table 9: Classifier performance on subset of 24 legal documents. HL is the Hamming Loss metric. TE stands for Topology Enforced, LR is the logistic regression model.

5. Discussion and conclusion

To rehash, the two research questions of this study:

"Is automatic classification of the law areas a legal document applies to possible with sufficient accuracy?"

"Is a more complex logistic regression model better suited than a multinomial naïve bayes model for this classification problem?"

Legal documents can be automatically classified to a great accuracy. We can calculate the accuracy by taking the reciprocal of the zero-one classification loss. This leaves us with an accuracy of >93% for the best performing model (the logistic regression model).

This allows us to falsify the sentiment of Opsomer et al. that legal documents can not be automatically classified. The small pilot comparison indicates that there is reason to believe that the performance of the predictive model is equal of greater than that of manual classifications by legal experts.

Comparing the loss metrics and the recall, precision and f-score of the Logistic Regression model and the Multinomial Naïve

Bayes model, we can conlude that the more complex logistic regression model outperforms the Naïve Bayes model on all measures. This answers the second research question.

Suitability as filter in a legal search engine. The current system may only be suitable for use as a filter in a legal search engine for the labels with sufficiently high precision and recall (which fortunately captures a large fraction of all documents). In table 7 one could see that although the average metrics of the model are very good, half of the labels is poorly classified. This is likely due to the extreme imbalance of the labels found in dataset.

5.1 Future research. Despite the good performance, there are still areas where improvement may be found. Here we discuss a few.

Domain knowledge. In the current approach domain knowledge is used, in particular, the post-processing step where superclass labels are added if a subclass label is present. One can apply this domain knowledge more effectively, i.e. much more elaborate classification schemes are possible. We know that a document always has one of the 4 main legal area labels. One can train one-versus-rest binary classifiers on these four classes, and employ a separate one-versus-rest binary scheme for each superclass's subclasses.

No distinction was made between the types of legal texts (*verdicts* and *conclusions*) because the goal here was to have a system which generalizes to both. However, it could still be valuable to train a separate classifier for both types of documents, or it could be used as a feature. Aside from this metadata, there may be more metadata present which can be used as a feature.

Document representation and feature selection. We used the simple *tf−idf* document representation, future work could investigate the use of more advanced weighting of terms such as *tf−rfl*[14]. Specific types of words were removed in the preprocessing step, with the *tf−idf* document representation this yielded better performance. This may not be the case for other document representations and should be re-considered.

Non bag-of-words. In this study only unigram features were considered (due to memory constraints), any information contained in the ordering of words was not captured in the feature vector. A non bag-of-words (bigram) approach could also be worth investigating.

Acknowledgments

I would like to thank Legal Intelligence for providing the extensive dataset, the manual classifications, and providing additional computing resources for processing the data. In particular, I want to thank Tjerk de Greef who was my contact there and organised all of this.

References

[1] Alexander Boer et al. "Metalex: Legislation in xml". In: *Legal Knowledge and Information Systems. Jurix* (2002), pp. 1–10.

[2] Alexander Boer et al. "Proposal for a dutch legal xml standard". In: *Electronic Government*. Springer, 2002, pp. 142–149.

[3] Rob Opsomer, Geert De Meyer, and Greet Van Eetvelde. "Why it is necessary for legislators to annotate legislation with meta-data". In: *Frontiers in Artificial Intelligence and Applications*. Vol. 205. Proceedings of the 22nd Annual conference on Legal Knowledge and Information Systems. Amsterdam, The Netherlands: IOS Press, 2009, p. 5.

[4] Marie-Francine Moens. "Innovative techniques for legal text retrieval". In: *Artificial Intelligence and Law* 9.1 (2001), pp. 29–57.

[5] G Governatori. "Exploiting properties of legislative texts to improve classification accuracy". In: *Legal Knowledge and Information Systems: JURIX 2009, the Twenty-second Annual Conference*. Vol. 205. IOS Press. 2009, p. 136.

[6] Emile de Maat, Kai Krabben, and Radboud Winkels. "Machine Learning versus Knowledge Based Classification of Legal Texts". In: *Proceedings of the 2010 conference on Legal Knowledge and Information Systems: JURIX 2010: The Twenty-Third Annual Conference*. IOS Press. 2010, pp. 87–96.

[7] Teresa Gonçalves and Paulo Quaresma. "Polylingual text classification in the legal domain". In: *Informatica & Diritto Journal* (2011), pp. 203–2016.

[8] A. van den Bosch et al. "An efficient memory-based morphosyntactic tagger and parser for Dutch". In: *LOT Occasional Series* 7 (2007), pp. 191–206.

[9] I. Schuurman and M. Schouppe. "CGN, an annotated corpus of spoken Dutch". In: *Proceedings of 4th International Workshop on Language Resources and Evaluation*. 2003, pp. 340–347.

[10] F. Van Eynde. "Part of speech tagging en lemmatisering". In: *Corpus Gesproken Nederlands, project internal document* (2000).

[11] F. Van Eynde. "Part of speech tagging en lemmatisering van het D-COI corpus". In: *Dutch Language Corpus Initiative, project internal document.* (2005).

[12] Fabrizio Sebastiani. "Machine learning in automated text categorization". In: *ACM computing surveys (CSUR)* 34.1 (2002), pp. 1–47.

[13] Mostafa Keikha et al. "Document representation and quality of text: An analysis". In: *Survey of Text Mining II*. Springer, 2008, pp. 219–232.

[14] Rodrigo Alfaro and Héctor Allende. "A new input representation for multi-label text classification". In: *International Conference on Instrumentation, Measurement, Circuits and Systems (ICIMCS 2011)*. ASME Press. 2011.

[15] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[16] Safdar Sardar Khan and Divakar Singh. "A Survey on Effective Classification for Text Mining using one-class SVM". In: *International Journal of Computer Applications* 65.23 (2013).

[17] C. J. Van Rijsbergen. *Information Retrieval*. 2nd. Newton, MA, USA: Butterworth-Heinemann, 1979. ISBN: 0408709294.

[18] R. E. Schapire and Y. Singer. "BoosTexter: A boosting-based system for text categorization". In: *Machine learning* 39.2 (2000), pp. 135–168.