university of groningen

faculty of arts

Predicting Applicable Laws for Court Cases

Exploiting the inherent structure of case transcriptions and the law to improve multi-label classification

Tim Kreutz

Supervisors:
Prof. dr. G.J.M. van Noord
dr. M.B. Wieling

Groningen, August, 2017

Abstract

Recent trends that promote open availability of public sector information have resulted in new data sources for natural language processing research. The Dutch court system has embraced the trend and with it the Open Data format for publicizing anonymized court cases. This thesis aims to automatically classify a collection of more than 10,000 court transcriptions to determine which of the 72 possible laws apply to each. The multi-label classification task is approached by looking at which existing techniques in Natural Language Processing and Machine Learning can provide the best outcome, and by exploiting the characteristics specific to court cases as a data source and the law as subject matter.

A study of relevant literature suggests that the Binary Relevance method is an effective and flexible solution for multi-label prediction, and after testing a variety of possible machine learning implementations it is clear that a collection of decision trees provides the best predictions. With a combination of features that leverage both linguistic and meta information about the documents, an average F-score of .7485 is achieved.

It is further shown that modelling the relationships between labels, by reusing the predictions of the model as input for repredictions, can further boost performance. This is confirmed with the overall best performance being measured after three repredictions (F-score .7522).

Although some of the unique qualities of legal data for natural language processing were not properly leveraged, this thesis provides optimistic results in predicting applicable laws for court cases and a useful insight into the interdisciplinary field of computational linguistics and law.

Preface

This paper is the final hurdle in the Information Science masters degree and for me it represents the end to a five year journey. It certainly has not been easy. I would like to thank my dad for getting me on the right track in writing this thesis through vague yet helpful advice. I would also like to thank my supervisor Gert-Jan van Noord for the feedback. Finally, I owe thanks to Tom Bouwhuis, Elvira Slaghekke, Martijn Wieling, Lucinda Fernandez Gonzalez and my mom.

Contents

1 Introduction

In 2003, the European Parliament issued a directive to promote accessibility and re-use of public sector information. Although the directive did not obligate the European Union member states to publicize their documents freely and directly, it urged them to promote such government transparency and set out the initial guidelines. Most importantly, nations were to restrict legislative, financial and technical limitations on the availability of government documents. In other words, citizens or organizations should upon request of any open government written text, audiovisual file or database, not be faced by unnecessarily difficult procedures, unreasonable fees or unusable file formats [2].

In the past decade most European member states have passed some form of the directive into national law. Public sector bodies on national and local levels are obliged to timely and correctly publish their open data upon request, with very few exceptions. Beyond that, some government bodies are actively publishing (part of) their documents as Open Data.

1.1 What is Open Data?

Open Data, according to OpenDefinition.org, *"is data that can be freely used, re-used and redistributed by anyone - subject only, at most, to the requirement to attribute and sharealike"* [4]. Without a clear example of such data this definition remains very broad, but exploring some of the most important characteristics of Open Data can limit the scope.

Open Data needs to adhere to the following characteristics. With regards to access, the data must be published completely and in a convenient, editable format. The costs should be reasonable and at most cover actual costs of reproduction and publication on a per-user base. The data should be provided under terms that allow for re-use, redistribution and altering. Restrictions on the data cannot be selective, Similarly, no individual users should get exclusive access. In the specific case of government information some cases national security restrictions may apply and it is important that data is made non-personal.

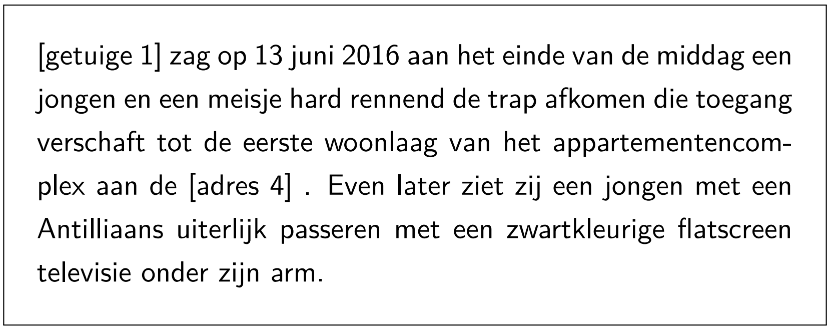
These characteristics assure that anyone can use open data freely and intermix with other datasets. Interoperability is the strength of open data and it allow companies of individuals to develop better ideas, product and services [4].

If we take this broadly defined benefit of the distribution of Open Data and apply it to government, the improved ideas become new government policies and better products and services elude to better functioning citizen registration, waste collecting or even road traffic safety.

1.2 Court Cases as Open Data

The Dutch government has been progressive in embracing Open Data and the European Parliament initiative. As early as 1997 it started an initiative towards more openness of government information through the use of Information Technology [5].

In 1999 it launched the Rechtspraak.nl website which contained a databank of Dutch court cases. Since then, this databank has grown not only in the absolute number of freely accessible cases, but also in the portion of total court cases that were processed and anonymized for online publication (Figure 1.1). In 2004 less than one percent of all court cases found their way to rechtspraak.nl, but in 2015 the number had risen to more than three percent, or 25.000 court cases [11][12].



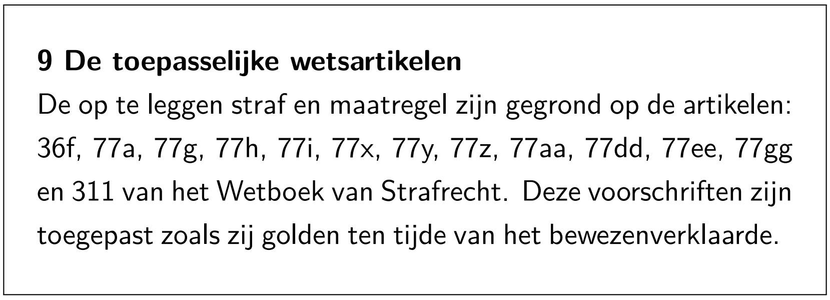
[getuige 1] zag op 13 juni 2016 aan het einde van de middag een jongen en een meisje hard rennend de trap afkomen die toegang verschaft tot de eerste woonlaag van het appartementencomplex aan de [adres 4] . Even later ziet zij een jongen met een Antilliaans uiterlijk passeren met een zwartkleurige flatscreen televisie onder zijn arm.

Figure 1.1: Fragment of an anonymized court case

The early start to the design of data in an open format and the significant uptake in publication of court cases has lead to a substantial collection of Dutch court cases. Beyond the collection's size, court cases are made up of long sections of written (near-natural) text transcripts, and thus should be seen as excellent subject matter for analyses using Natural Language Processing (NLP).

1.3 Natural Language Processing on Legal Data

This thesis will focus on a specific case of using NLP and Machine Learning techniques on legal big data obtained from Rechtspraak.nl. As a broader perspective, we will analyze some of the challenges that go into the automatic processing of court case transcriptions. From a case perspective, we choose to focus on the prediction of laws that apply to individual court cases. For certain court cases the basis of the verdict is neatly annotated as in Figure 1.2 and can readily be used as a reference.



9 De toepasselijke wetsartikelen

De op te leggen straf en maatregel zijn gegrond op de artikelen: 36f, 77a, 77g, 77h, 77i, 77x, 77y, 77z, 77aa, 77dd, 77ee, 77gg en 311 van het Wetboek van Strafrecht. Deze voorschriften zijn toegepast zoals zij golden ten tijde van het bewezenverklaarde.

Figure 1.2: A court case explicitly mentioning the applicable laws.

Since there can be multiple laws that influence an individual court case (Figure 1.2), the problem at hand is a multilabel classification problem. In this sense we can research similar problems to understand existing solutions to multilabel prediction. In another way, dealing with a familiar problem can illustrate the uniqueness in our particular dataset. We accordingly focus some of our research questions on the exploitation of features inherent to court cases towards a better prediction.

Our formalized research questions read:

1. *How can we predict applicable laws for transcribed court cases?*
2. *How effective are existing techniques for predicting applicable laws?*
3. *What can we learn from court cases as a data set to better predict applicable laws?*

In the next chapter, relevant texts on NLP in law will be discussed to gain a better understanding of the work that was previously done. Further, relevant articles that similarly aim to predict multiple labels for documents will be consulted to give a general idea of useful approaches to answering the research question.

2 Related Work

This chapter will describe previous work to give a better frame for the posed research questions. First and foremost this will be work that has used Natural Language Processing and Machine Learning techniques to analyze legal texts. Beyond a mere description we will go into the challenges that such papers highlight and use this in going forward with our own research.

Secondly, we will seek out literature that gives a more methodological direction. As is the main focus of our second research question, we will examine tasks that deal with multi-label classification to understand and employ existing techniques. In this way, our methods will not be based solely on a trial-and-error approach.

2.1 Jurimetrics

Quantitative methods are rarely used in legal research. More prevalent are the annotation and analyses of single court cases in the context of existing legal theory. The latter methodology is commonplace, on the one hand, because its suitability to expand academic knowledge in the legal domain is tried-and-tested. On the other hand, there is a certain bias in its prevalence since the study of quantitative techniques are mostly absent in legal scholarship.

Research that uses large collections of legal data and descriptive statistics has distinct usefulness over the aforementioned method, however, in being able to surmise trends that are harder to determine by human annotators. This distinct usefulness is noted in a large variety of papers that go back as early as the mid nineteenth century [3]. The American jurist Lee Loevinger coined the term 'Jurimetrics' [6] in an appeal to further legal knowledge using empirical methods rather than speculative methods. Jurimetrics has evolved over time, mostly becoming more specific in its meaning. The two main components are: the empirical study of legal phenomena (1) with the aid of mathematical models (2) [3].

2.2 Natural Language Processing and Machine Learning

If we look at work that relates more specifically to the proposed research questions, there are some good examples of articles that adhere to the principles of jurimetrics. Beyond the mentioned 'aid of statistical models', the combined use of Natural Language Processing (NLP) and Machine Learning (ML) yields researchers powerful methods to analyze any collection of texts, legal discourse being no exception [1].

2.2.1 Predicting outcomes for human rights court cases

In a study by Aletras et. al [1] an attempt is made at predicting judicial descisions of court cases. Similar to the Dutch judicial system, the European Court of Human Rights (ECHR) publishes a percentage of case transcriptions on a dedicated domain [7]. Aletras et. al [1] took 600 case transcription and tried to predict their outcome.

The research is in many ways similar to what is proposed in this thesis, and demonstrates some of the considerations to be made with regards to data selection. For one, there is a balance to strike between data scarcity and representability. Scarcity is a large concern with the relatively small ECHR dataset, and leads to an oversimplification of predicting outcomes of court cases:

> "We focus on [...] three articles for two main reasons. First, these articles provided the most data we could automatically scrape. Second, it is of crucial importance that there should be a sufficient number of cases available." (quote)

The authors have to narrow the focus of the paper to predicting the violation of only three articles set out in the European Convention of Human Rights. On the one hand this is done to prevent data scarcity: the chosen articles provided the most data. On the other, it limits the representability of the study and its results: the high overall accuracy of 79 percent does not represent the ability to reliably predict the outcome of any case that comes to the ECHR.

A second simplification can be found in the relatively simple feature set and ML approach taken by the authors. A comparison of different feature sets and ML algorithm could provide more insight into using court cases as a corpus for training.

In addition to the pitfalls that we have described there are more explicit lessons to be learned from the 2016 paper. A large section is dedicated to the distinctive structure of court judgments. This structure is not incidental but adheres to Rules of the Court [7], which guarantees the presence of distinct textual sections with a standardized order. Although this structure is not as standard in the Dutch court rulings, some order to the textual sections is still provided depending on the type of court case (for examples see [8] and [9]).

2.2.2 Simple domain adaptation

It makes sense to use the structure that is specific to our legal data to better predict applicable laws. Similarly, it makes sense to adapt to domain-specific language. Court transcriptions do not follow the same conventions as other domains. Sentences often include excessive enumerations to account for all possibilities and uncertainties in a described situation (Figure 2.1). Legal texts are also riddled with elements that would not be expected in conventional text like references to other cases or legal jargon (Figure 2.2).

> hij op of omstreeks [datum] te [plaats] , op de openbare weg [openbare weg] , althans op een openbare weg, tezamen en in vereniging met een of meer anderen, althans alleen, met het oogmerk van wederrechtelijke toe-eigening heeft weggenomen een portemonnee

Figure 2.1: Excessive enumerations typical to court transcriptions.

> Er is geen omstandigheid aannemelijk geworden waardoor de wederrechtelijkheid aan het bewezenverklaarde zou ontbreken. Het bewezenverklaarde is derhalve strafbaar.

Figure 2.2: Use of domain-specific wording and jargon in court transcriptions.

Problems arise when a system heavily relies on tools that were previously trained or tuned on other discourse, or when there is no way to deal with unexpected elements [15]. [16] identifies two other weaknesses when applying existing NLP techniques to a new domain. First, metadata that is available for the new domain is rarely leveraged. Second, traditional NLP systems often treat texts from different authors as the same. The described problems of domain adaptation can easily be translated to the task of predicting applicable law for the Dutch court transcriptions. It specifically introduces new considerations when it comes to data selection, processing and feature selection.

2.3 Multi-label classification

The legal dataset brings up a lot of challenges and considerations in and of itself, but the task of multi-label classification is not straightforward either. It goes beyond regular single-label classification and asks not only which label applies, but also how many labels apply for a given document.

Generally there are two approaches to multi-label classification. The first approach, also known as Binary Relevance (BR), is to train a binary classifier per label to predict whether that label applies to a document or not. Though a single classifier may achieve a high performance, the overall performance is a result of all classifiers' errors and may be low. The second approach does not split the problem but rather requires extra effort to determine the amount of labels to predict per document. For instance, all applicable labels are merged to form one label that can be predicted by a single-label classifier [13].

2.3.1 Classifier Chains

Over the past decade, variants of both approaches have developed to take into account dependencies between labels. Dependencies certainly exist when we look at laws that apply in court cases. There will be a lot of cases that mention laws that target both burglary and theft, for example.

A special way of doing BR is the Classifier Chain model (CC). CC is an ensemble method that takes predicted labels of one classifier and uses it as input for other classifiers to the end of modelling label dependencies [10]. Although this would model label dependencies, the order in which the individual classifiers are fed to the chain is really important to the performance of the CC. With only a few labels, this order can be randomized a number of times to test which order performs best. With a lot of labels however, such an approach is less feasible.

2.4 On Methodology

To surmise this chapter on related work it should be said that there is a lack of work that both closely resembles this thesis and thoroughly experiments with methodological configurations. Our own methodology will on the one hand be inferred from broader considerations we have described; on data selection and label dependencies for example. On the other hand we will try a lot of configurations to determine the best approach to the particular data set and the task at hand.

3 Method

This chapter will look at the methods that were used to come to our research results. First, I will describe the data that was used. To come to a feasible research that is understandable and reproducible, some choices with regards to data selection are made. I will explain the considerations and substantiate the choices.

In the second section I will go into the techniques that were used to come to a multi-label classification. This section contains both the ML-algorithms that were tested and the feature sets I developed and tested.

In the last section I will go more into depth by describing the system that I developed to leverage the domain-specific qualities of our legal data. As described in section 2.2.2, research using a distinct discourse can have certain weaknesses. Since most of the improvements that can be made to cope with these weaknesses require custom solutions, and most of the work that was done went into these solutions, an in-depth description is warranted.

3.1 Data

The Dutch court cases can be collected via two manners. To simply obtain the complete text of court cases, some website scraping software could be used. The whole collection of court cases (from 1913 to the current date) can also be downloaded all at once in a useful XML-scheme. We use the latter for two reasons. The OpenDataUitspraken project clearly asks users to not 'hammer the server' using simultaneous page requests [12]. This appeal is supported by the argument that the project aims to make Open Data available for a marginal cost and increasing server capacity conflicts this aim. Although we are not interested in most of the meta-data that the schema provides, a second reason for using the complete collection are some of the tags contained in it. As we will see, there is a wide range of type of court cases, and the tags will help by making a selection that is coherent and understandable.

3.1.1 Data selection

There is a lot to consider when limiting the data to make up a coherent and understandable collection. I will try to discuss these considerations by order of importance (or simply by the relative amount of court cases they filter when applied).

The recency of court cases is important to consider. Since our aim is to predict which articles in Dutch law are relevant to a certain court case, it can logically be assumed that changes in the law will muddle our system and its results. Studying changes in the law in the past century is beyond the scope of this paper. We will thus try to limit the risk of big alterations to articles by using recent court cases, whilst assuring a data set large enough to train a system. If we consider only court cases from January 2012 until June 2017 this leaves us a sizeable collection of 865,078 court cases.

A second important delimitation on our data set is to consider only court cases that deal with criminal law. The reasoning here is twofold. First and most important is the much needed limitation on type of articles that we will consider as labels. As seen previously [1], predicting only a few articles simplifies a research to the point of being experimental only, without any practical use or representability. By focusing on a very specific (but substantial) domain of Dutch law we keep the research feasible while insuring representability. It is perfectly imaginable that the research at hand can be repeated for other domains of Dutch law. The reason for choosing criminal law specifically is that it appeals to any readers imagination. Although such a consideration should be secondary in most cases, in a first of its kind research such as this the use of understandable examples can be very important.

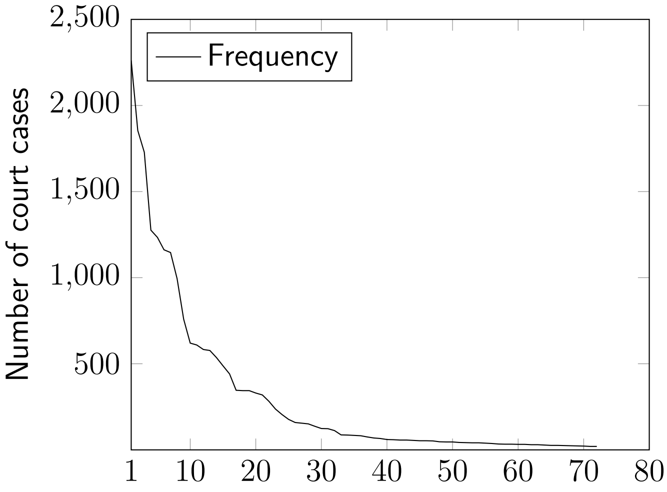


Figure 3.1: Ranked labels and their occurrence in our data set. The most common label leftmost, the least common rightmost.

Table 3.1: List of labels and their frequencies

Rank	Article no.	Common description	Number of court cases
1	310	Theft	2,262
2	311	Qualified theft	1,855
3	312	Violent theft	1,728
4	287	Manslaughter	1,276
5	300	Physical abuse	1,162
..	..	..	..
68	131	Sedition	24
69	363	Passive bribery	23
70	177	Bribery	22
71	365	Coercion by a civil servant	20
72	316	Theft between partners in matrimony	20

Following this reasoning we will also only consider first instance cases. Although this delimitation is meant to only filter appeal cases or similarly successive cases that go into the functioning of the court and its judges, we find that a surprisingly large portion of the XML-files lack the tag that denotes whether a case is a first instance case. We conclude that the remaining cases from each of our filters do not represent the share that realistically would remain if all tags were correctly filled out. For instance, we believe that court cases in criminal law should realistically make up more that roughly six percent of all court cases.

Aside from selections based on the type of court cases, we have to analyze our data from a label perspective. It would not be useful to have labels appear only in single instances of court cases. To cope with very rare instances, we remove the distinction between sub-articles in the law. For instance, both variants of burglary mentioned under articles 138a and 138b of Dutch criminal law will be merged and hence be interpreted as label 138. After using this clustering of articles in the law, it is very rare to find single instances of labels. We can implement a cut-off point for labels by removing those that occur in less that twenty court cases without drastically reducing the amount of court cases (see Figure 3.1 and Table 3.1).

We are still left with a sufficient collection of over 10,000 court cases after filtering files that do not clearly mention the applicable laws or mention applicable laws that fall outside our selection of 72 criminal law articles (Table 3.2).

[1] Some of the remaining court cases did not contain any text. Others did not refer to any of the labels that remained after applying our cut-off point. This table can be seen as a reflection not only on which selection criteria can be applied but also of how much of the data gets filtered because it is inconsistent or incomplete.

Table 3.2: Data selection overview

Year	All court cases	criminal law cases	first instance cases	other filters[1]
2012	189,933	12,009	3,612	2,348
2013	173,508	10,229	3,090	1,898
2014	154,790	8,497	2,908	1,696
2015	149,664	8,408	3,234	1,861
2016	143,238	7,164	2,764	1,681
2017	53,945	2,874	1,382	857
Total	865,078	49,181	16,990	10,341

3.1.2 Data preprocessing

In its current XML-format our documents contain a lot of meta-information. Only the unstructured data, the textual transcript, will be used to predict labels. As a first preprocessing step we will extract the textual information only.

In most cases, elements such as dates and locations and replaced by placeholders (see Figure 2.1). We are not interested in this specific information because it does not have any relation with the labels we try to predict. In cases where we still find dates or location specified by cross referencing them with lists of words, we replace them with the '[datum]' and '[locatie]' placeholder respectively.

The court cases also frequently contain reference to legal articles. Since we want to imply applicable laws from the description of cases, not by plainly retrieving this information from the text, we look up any reference to numbers in the text and remove these. This may seem as somewhat of an aggressive approach, since it may remove other information from the text. However, no type of number, albeit dates or imposed fines, is of any interest to the research at hand since they are too variable to generalize from.

Table 3.3: Description of document collection after preprocessing

Total lines	2,360,972
Total tokens	58,483,464
Total types	324,509
Average lines per document	231.81
Average tokens per document	5742.12
Average types per document	996.67

The 10,341 plain text files that remain are easy to tokenize and hence surmize (see Table 3.3).

3.2 Multi-label classification

We will use the Binary Relevance (BR) approach from the chapter on related work as our way of dealing with multi-label classification. BR transforms the multi-label classification problem by splitting the problem into multiple parts. These parts are binary classifiers that were trained to identify a single applicable law in a court case. The overall results that BR yields are the (macro) averaged performances of the binary classifiers combined.

3.2.1 Binary Relevance

Document names follow the naming convention of European Case Law Identifiers (ECLI) which denotes location (NL for Netherlands, RBUTR for Rechtbank Utrecht), year and unique case number. In multi-label classification, any document is linked to a list of labels as the first step in the figure below.

With the BR method, the list of labels that a document is linked to is binarized. If our labelset is limited to the example above (i.e. the set {157, 225, 300, 310, 311}), we can transform the labels to the second step.

```
ECLI_NL_RBUTR_2012_2953.txt      {300,310,311}
ECLI_NL_RBLIM_2013_8507.txt      {225,310,311}
ECLI_NL_RBAMS_2015_7662.txt      {157,300}
```

```
ECLI_NL_RBUTR_2012_2953.txt      {0,0,1,1,1}
ECLI_NL_RBLIM_2013_8507.txt      {0,1,0,1,1}
ECLI_NL_RBAMS_2015_7662.txt      {1,0,1,0,0}
```

```
ECLI_NL_RBUTR_2012_2953.txt      1
ECLI_NL_RBLIM_2013_8507.txt      0
ECLI_NL_RBAMS_2015_7662.txt      1
```

Figure 3.2: Problem transformation for Binary Relevance classifiers

And for a classifier that gets trained to predict a single label (article 300 for example), labels are further stripped to become step three.

There are a lot of candidate Machine Learning algorithms for these single classifiers. We will only look into the possibilities the Scikit-Learn framework offers. Scikit-learn is a framework for the Python programming language and offers a wide range of machine learning implementations. Scikit-learn is also selected because I am familiar with the framework, and it will allow for the easy comparison between classifiers and classifier configurations that we mentioned in section 2.4.

3.2.2 Features

The developed features aim to cover both linguistic and meta information from the documents. Some are proven to work, like word (co-)occurrences [1], while others were conceived simply from reading the court cases. Beyond common sense suppositions we developed features on the basis of previous work that states weaknesses of domain adaption [16], namely the assumption of same authorship and failing to leverage available meta data. To tackle both, we use the location of the court that processed the case as a feature. We also go into the way a case has been anonymized by counting the specific anonimity tags. The features can roughly be divided into two categories, namely content features and structure features:

3.2.3 Content features

Content features were developed to capture the content of a document. Usage of specific words, and specific legal jargon for court cases, are signals of certain laws applying to a court case. Although the literature does not suggest any specific configuration of features, we will try to find the best way to use and combine the developed features. Content feature do not necessarily help to indicate how many laws apply to a document, which is why we separately implement structure features.

Word n-grams

We use word occurrence (unigrams) to come to a Binary Relevance baseline. This feature will be informative in ranking words and co-occurrences of words that are key to signifying specific articles of the law. For our final implementation we use single words, bigrams and trigrams.

Skipgrams

Skipgrams are useful to tackle scarcity in an n-grams feature. If combinations of word are particularly uncommon, the n-gram feature may actually hurt performance. A skipgram feature also considers n-grams of words not directly adjacent thus potentially increasing the count of uncommon co-occurrences. Court cases specifically contain a lot of filler words (see for example Figure 2.1) that could be skipped to increase the amount of useful bigrams. We use bigrams with up to two skips.

Adjective count

Frequent use of adjectives can signify a more descriptive text. We argue that descriptiveness in court cases plays a bigger role when handling certain articles of the law. There are more sophisticated ways of implementing part-of-speech in predicting applicable laws, but the distinct sentence structure of court cases should first be studies extensively to understand how it would affect POS-taggers. For now, we just count the amount of tagged adjectives per documents and add this count as a feature.

Tagged entities

As a preprocessing step of the courts that publish the case transcriptions, named entities are replaced by a range of tags. We have extended upon this preprocessing as described in section 3.1.2. Use of certain tags is argued to signify a type of event which in turn relates to violations of articles of the law. For example the frequent mention of a licence plate number can mark the importance of a vehicle. We count the occurrence of the tags mentioned in Table 3.4 as features to the model. There are other more uncommon tags but their use seems arbitrary rather than related to a certain type of case.

3.2.4 Structure features

The structure features were developed with a simple idea in mind: which characteristics of a document can be used to determine how many laws apply to the document. Beyond words usage, we argue this will relate to the length of a document and the other meta-features.

Document length

The length of a document can signify a longer court procedure. We think the length or complexity of the procedure can relate to the complexity of the law being discussed, hence implies a certain area of criminal law. This feature is operationalized as the number of tokens the document contains.

Type token ratio

The number of word types (unique words) used in a court case relative to the number of words used says something about the specificity of language and use of jargon. We argue that complicated cases have a higher ratio and similarly correlate with certain types of cases.

Number of paragraphs

The number of paragraphs further indicates complexity in a case or the number of distinct topics that are discussed. This feature is operationalized as the number of newlines a document contains.

Table 3.4: Anonimity tags in court cases that serve as features

Tag	Denotes
[verdachte]	suspect name
[nummer]	any number
[(bedrijf) naam]	arbitrary name (company or venue)
[adres]	address
[slachtoffer]	victim name
[aangever]	declarant
[kenteken]	licence plate number

Court location

The location of the court is usually not explicitly mentioned in the unstructured textual data. The meta tag can be relevant however, if not in the type of case the transcription described, in the style of the transcription. This feature can hold on of eleven possible features in our current dataset: 'rechtbank amsterdam', 'rechtbank den haag', 'rechtbank gelderland', 'rechtbank limburg', 'rechtbank midden-nederland', 'rechtbank noord-holland', 'rechtbank noord-nederland', 'rechtbank oost-brabant', 'rechtbank overijssel', 'rechtbank rotterdam' or 'rechtbank zeeland-west-brabant'.

3.2.5 Feature sets

Since we have a relatively small set of features, any combination of the described features can be evaluated to find the optimal feature set. N-grams and skipgrams however, are expected to capture the same information and will not be used in conjunction. The additional features are mostly experimental and may or may not be beneficial to performance.

3.3 Modelling label dependencies

We propose an approach to multi-label classification that leverages knowledge of the structure of the law to make better predictions of applicable laws. The binary relevance method assumes a relationship between our developed features and single labels. In court cases where more than one law applies however, we also imagine a relationship between labels in the labelset.

It is no large stretch of the imagination that some crimes co-occur more often than others and even that some crimes can entail other crimes. For instance, theft is described in article 310 of the Dutch Wetboek van Strafrecht but features a more general description of taking property that belongs to another. A more detailed description of types of theft can be found in article 311 and 312. Article 312 describes a more severe conviction for a combination of theft and assault. This in turn makes it likely for article 312 to co-occur with article 300 (assault).

3.3.1 Conditional probabilities

We can also quantify the expected dependencies between labels by calculating their conditional probability. This conditional probability of a label given another label can then be compared to its regular probability to see how they differ. Labels that have a high conditional probability given another label but a relatively low occurrence rate in the collection of documents, can be said to have a very strong relationship with the given label [14]. In a general sense, if we find a lot of these discrepancies, it mean that labels are not independent. What follows is that the (probable) occurrence of one label, can be useful information for predicting another label.

Table 3.5: Top ten conditional probabilities offset against non-conditional probabilities

Applicable law (Label A)	Applicable law (Label B)	P(B|A)	P(B)
Insult of an official (267)	Insult (266)	0.8808	0.0201
Assault of family (..) (304)	Assault (300)	0.8604	0.1212
Embezzlement at work (322)	Embezzlement (321)	0.8248	0.0340
Repeated fencing (417)	Fencing (416)	0.8070	0.0598
Collusion of nat. sec. (96)	Murder (289)	0.7188	0.0572
Extortion of property (317)	Violent theft (312)	0.7173	0.1697
Manslaughter accomp(..) (288)	Manslaughter (287)	0.7143	0.1253
Qualified theft (311)	Theft (310)	0.6933	0.2221
Insult (266)	Insult of an official (267)	0.6488	0.1821
Collusion of nat. sec. (96)	Crime of general safety (157)	0.6250	0.1125

Table 3.5 shows us that the expected dependency between labels does exist. We have used simple names for describing the co-occurring laws to give some idea of the reason behind high conditional probabilities. The table can be read as follows: Given a document in which article 267 applies, there is an 88 percent chance of article 266 also

applying. Given any document, independent of which article applies, the chance that 266 applies is approximately two percent.

> Aan verdachte is — na meerdere wijziging ter terechtzitting — kort gezegd, cumulatief/alternatief ten laste gelegd dat hij zich heeft schuldig gemaakt aan medeplegen van voorbereiding en-/of bevordering van moord en/of doodslag met een terroristisch oogmerk;

Figure 3.3: Typical fragment from a court case with applicable articles 96 and 289.

Strong relationships are supported by the content of the co-occurring laws. Some cases can more easily be expected than others. We recognize that persecutable insults are usually insults of government officials in function, assault happens most often in the familial context and embezzlement happens in a professional context. More interesting is the common co-occurrence of the collusion against national security and manslaughter. Closer inspection of the documents that feature both laws shows us that all court cases are relatively recent (between 2015 and 2017) and concern trials of "Syriegangers" or foreign fighters in the Syrian Civil War (see Figure 3.3).

3.3.2 Limitations of Classifier Chains

In the chapter on literature, one method of modelling label dependencies has already been mentioned. Classifier Chains (CC) can improve performance in multi-label classification tasks where there are dependencies between the labels. Since it relies on chaining classifiers that directly influence the next in the chain, successfully modelling label dependencies becomes a lot harder and computationally expensive when there are more possible labels. Ideally, each chain order is tested and the results are averaged to come to the best classifications, but in a problem with 72 possible labels there are 72! possible combinations. As a result, using CC in the task at hand is unfeasible.

3.3.3 Proposed method: Stacking classifiers

Instead of using the prediction of a single classifier as extra input for the next, it would be more useful to get a predictions for all classifiers and using this as extra input to repredict new labels. I propose a method that uses the designed classifier, not to predict one or multiple labels for the court cases, but to predict probabilities of each of the labels for a court case and inserts these probabilities as a feature for retraining. This way, if the initial classifier deems it very likely that a certain label is present in a court case, this likelihood itself could serve to influence its prediction of other labels in the court case.

In other words, to deal with the idea that the prediction of one law may affect whether another becomes more or less likely to occur, we will iteratively retrain the selected model using our original features and the predicted probabilities of the previous iteration. The proposed method of stacking classifiers is further visualized in Figure 3.4.

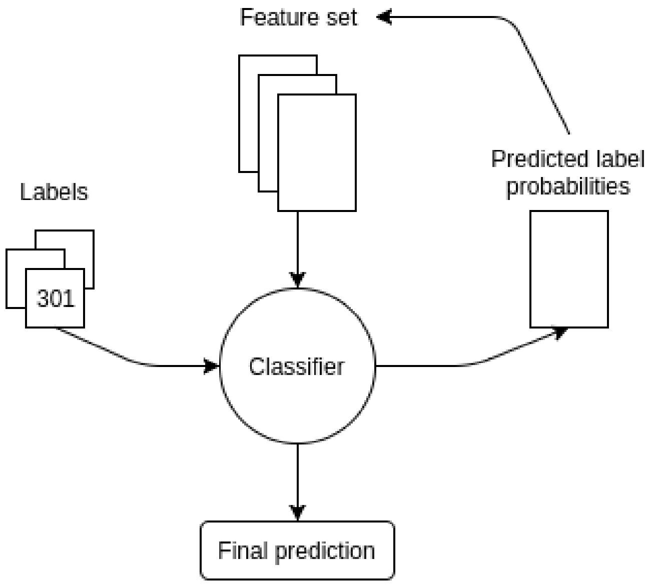


Figure 3.4: Stacking approach to model label dependencies

Using this custom approach also gives us the opportunity to better answer research question 3: *What can we learn from court cases as a dataset to better predict applicable laws?* We will gain insight into the effect of the (probable) presence of one law for better predicting the presence of another, thus leveraging the inherent structure of the law to come to a better prediction.

4 Results

This chapter describes the results that our methods yielded. To keep the chapter understandable as a whole and focus on what the results mean for our research questions, we incrementally describe partial results and the direction they gave to further analyses. For example, the choice for the ML-algorithm can be made in an early stage using a basic feature. This makes further analyses more delimited because we then focus solely on the best algorithm. Whereas the chapter on our methods mostly expanded on the possible routes to take in answering our research question, this chapter reversely abandons some paths that prove unfruitful. In addition, we will go more into some of the results when deepening them can help us gain understanding as to why something worked or did not work, so that we can implement these notions in our chapters that discuss and conclude our research.

4.1 Binary Relevance

We use the Binary Relevance method to make the best predictions about which laws apply in court cases. In this section we will choose the best machine learning algorithm for the task at hand and see which combination of features yield the best results.

4.1.1 Machine Learning algorithms

There is some trouble in coming to a baseline approach in multi-label classification. The literature offers no consensus either. In multiclass classification, one easy way to determine a baseline is to let the baseline model predict the most frequent label in all cases. By splitting the problem into a binary classifier per label (Binary Relevance), given that no single label applies to more than half the documents, such a baseline would predict no labels per document at all.

But we should not get stuck in a theoretical discussion on what constitutes a good baseline. Instead, we can compare machine learning algorithms by letting them train a basic feature. We choose a word count vector to represent the documents for our baseline, so that we can compare the ML-algorithms in a Binary Relevance approach.

Table 4.1: Performance of different Machine Learning algorithms with unigram features

Algorithm	Precision	Recall	F-score
Stratified Dummy Classifier	0.0286	0.0291	0.0287
Naive Bayes	0.3202	0.6334	0.3838
Linear Support Vector Machine	0.7120	0.5303	0.5917
3-layer Perceptron classifier	0.5682	0.6604	0.6108
Decision Tree Classifier	0.6637	0.6169	0.6304
Adaboosted Decision Tree	**0.8030**	**0.6362**	**0.6878**

Table 4.1 shows the tested algorithms in order of performance. We used 75 percent of the data for training the algorithms and 25 percent for testing. Precision, recall and F-score are useful metrics because they give us insight into the reason behind low overall performance. The Stratified Dummy Classifier is useful as a baseline classifier to get an idea of the relative difficulty of determining the labels, even when respecting label distribution (which should have a considerable effect, see Figure 3.1).

The decision tree algorithms perform the best overall with a good balance between precision and recall. If we plot the decision boundaries in a two-dimensional feature space we can see the different ways the best performing algorithms deal with classifying some testing documents. We use the most important features, which in our feature correspond to the occurrences of the two most determining words in a document (Figure 3.4). We find that these words are often used in common phrases that relate to court cases dealing with fraud (Figure 4.2).

Linear support vector machines make linear distinctions in the feature space. Its merit is perhaps not fully demonstrated using only a small portion of the features, but a large set of linear distinctions end up yielding high precision but a lower recall. The 3-layer perceptron neural network requires very specific tuning to perform well at the task. The best performance was measured when not making the second layer too extensive (to limit overfitting) but with frequent retraining iterations (Appendix A). Figure 3.4 shows us that it does not look much different from the linear distinction made by the SVM, but its performance shows us it features a slightly more sophisticated way of separating the binary classes.

The decision tree classifier makes hard cutoffs for each of the features in its decision boundaries. It shows great performance with the unigram features when combining all binary classifiers. Using Adaboost to retrain the decision tree on cases with low certainty further improves performance. We will continue using it to evaluate our designed features from the previous chapter and to predict the probabilities that we use to stack in our custom approach.

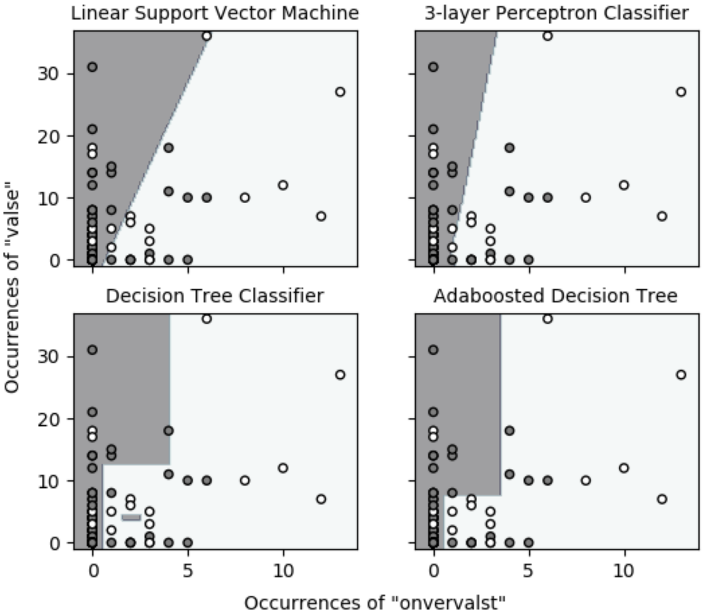


Figure 4.1: Decision boundaries for the classifier of cases with 'fraud' using its two most important features.

> (...) het opzettelijk gebruik (laten) maken van de valse of vervalste passen of kaarten als ware deze echt en onvervalst dan wel het opzettelijk zodanige passen of kaarten afleveren, voorhanden hebben, ontvangen, zich verschaffen, vervoeren, verkopen of overdragen (...).

Figure 4.2: Example of word occurrence in court cases about 'fraude'.

4.1.2 Feature sets

The feature sets were constructed to contain a linguistic aspect as well as meta-information on each document. By adding features to the sets one by one, we see the impact of each. However, we mentioned that we find no (theoretical) merit in combining n-grams with skipgrams and we alternate between both to form the basis of each feature set.

N-grams perform better than skipgrams but the difference is relatively small. We also see only small differences when adding one of the other features to the n-gram features. Only in the case of structure features (namely document length, type token ratio and number of paragraphs) does the F-score increase marginally. The addition of court location to the feature set helps with recall, but lowers the precision of the model.

Table 4.2: Performance of the feature sets

Featureset	Precision	Recall	F-score
N-grams	0.8310	0.7146	0.7463
Skipgrams	0.8213	0.6873	0.7273
N-grams + adjective count	0.8103	0.6993	0.7345
N-grams + tagged entities	0.8339	0.7119	0.7460
N-grams + document length	0.8290	0.7155	0.7467
N-grams + type token ratio	**0.8358**	**0.7170**	**0.7485**
N-grams + number of paragraphs	0.8316	0.7153	0.7469
N-grams + court location	0.8309	0.7146	0.7463
All features combined (with n-grams)	0.8334	0.7033	0.7430

The other linguistic features only hurt the F-score. The full table has been added in Appendix B.

Though differences are marginal, we will select the N-gram + type token ratio feature set (emphasized in Table 4.2) for our stacking classifier.

4.2 Stacking classifiers

In the previous chapter, we described a method to deal with potential dependencies between labels. Now that we have tested a range of machine learning algorithms and chosen a best-performing feature set we will further test iterative stacking of our classifier to model the relationship of labels.

4.2.1 Stacking iterations

We can use the AdaBoosted Decision Tree to directly predict labels for a document, or predict the probabilities for all labels. Using the same example as above, if one classifier predicts article 267 but not 266 to apply, we can refeed the predicted probability of article 267 to the classifier to learn the importance of the occurrence of article 267 for 266. This way of leveraging label dependencies can widely be applied without feeding any specific non-conditional probabilities into a classifier. It can also be repeated until no significant boost in performance is measured.

The overall best performing iteration is after three stacks. The most typical improvement in the first stacking iteration is displayed Figure 4.3. The figure reflects the increased recall we observe in the table by finding additional applicable laws. False negatives hence turn to true positives in most cases where there is an improvement.

After the third iteration there are more and more cases in which labels are incorrectly

Table 4.3: Performance of stacking iterations

Iteration	Precision	Recall	F-score
No iteration	**0.8358**	0.7170	0.7485
1	0.8267	0.7223	0.7506
2	0.8153	0.7269	0.7519
3	0.8021	0.7302	**0.7522**
4	0.8010	**0.7303**	0.7520
5	0.8004	0.7303	0.7512

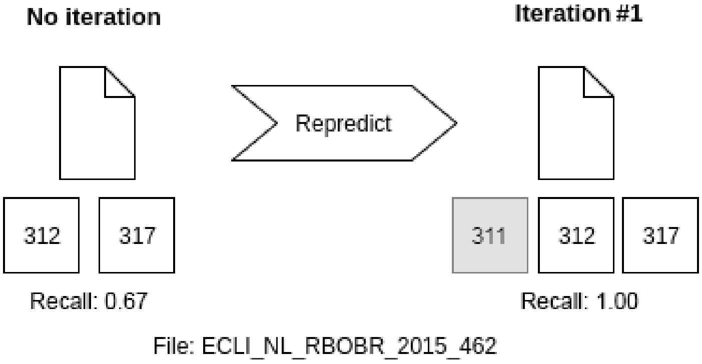


Figure 4.3: Improving recall after repredicting applicable laws.

added to the prediction or removed from the prediction, in a few cases some correct labels are even replaced by incorrect labels (see Figure 4.4), replacing a true positive by a false positive.

Again, looking into the content of changes to the label prediction is insightful to gain understanding of the data and label dependencies. Figure 4.3 shows the correct prediction of violent theft (article 312) and extortion of property (article 317). By feeding the high probabilities of both predicted labels, as well as the relatively high probability for the occurrence of qualified theft (312) the reprediction now also features qualified theft.

Conversely, Figure 4.4 shows a completely correct prediction at the third stacking iteration. The case at hand features rape (article 242) and assault (article 300). Reprediction in the fourth iteration sees the wrongful replacement of article 242 by crimes against personal freedom (article 285).

4.2.2 Final results

The final model is not overly complex but was formed through a range of different steps. First, we determined the best performing machine learning algorithm for binary

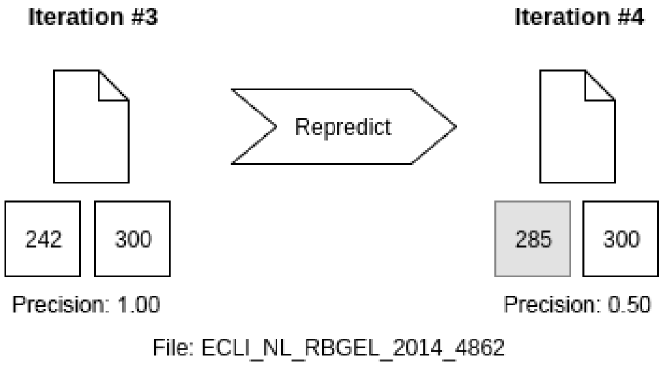


Figure 4.4: Hurting precision after repredicting applicable laws.

relevance by comparing a dummy classifier, a naive bayes algorithm, linear support vector machine and 3-layer perception classifier and two decision trees (one regular, one with Adaboost). We used counts of tokens as a basic feature and determined that the Adaboosted Decision Tree significantly outperformed all others and should therefore be used for the rest of the tests.

Combinations of features were made to assess the best configuration for predicting label probabilities. N-grams up to three words in combination with the type token ratio of the document marginally outperformed other feature sets and was used to predict the initial label probability distribution.

We then iteratively fed the best performing feature set and the repredicted label distribution to the same classifier to measure whether this boosted performance and which iterations performed the best. The best trade off between precision and recall was found after three iterations. The final F-score of .7522 is a great improvement over the results we reported at the beginning of the chapter.

5 Discussion

In this chapter we will critically assess the results from the last chapter and how they were acquired. Just like any other research, there are directions that could have been taken and we can make suggestions for future improvements off the back of these. First, I will discuss the features that were developed, how they aimed to capture structure in the court case data and what could be improved. The second section will discuss the stacking approach and possible alternatives. Lastly, I will go into the final results of the system and how we assess them in a broader context. Examples will accompany the reasoning behind the discussion.

5.1 Features

In subsection 3.2.2, the features were developed on the basis of our studies literature and manual inspection of the court case data. In retrospect, the developed features that were aimed to capture subtleties in court cases as a unique collection are rather shallow. This is mostly exposed due to their failing to boost performance in consolidation with the n-gram models, but remains a point of discussion when inspecting court cases and the underlying reasoning to constructing such features.

5.1.1 Paragraph structure

From Figure 5.1 it is clear that court cases follow a distinct pattern that separates the applicable laws by paragraph. The "number of paragraph" and "document length" features were constructed to not model this inherent structure, but rather to indicate that laws which occur in longer court cases may apply. This approach would work if it could indicate that structure of the court case is indeed of importance to predict applicable laws, but this is not the case in the research at hand. Conversely however, the inability of most structure features to boost performance does not dismiss the notion the court case structure matters, and could instead be caused by the shallow operationalization of these features.

A better approach then, would be to not take court cases as a unit (document) but to consider granularity of text units as a setting in coming to a model. In other words,

> De rechtbank acht op grond van voornoemde bewijsmiddelen wettig en overtuigend bewezen dat verdachte het ten laste gelegde heeft begaan, met dien verstande dat:
> 1. hij op te Middenmeer, gemeente Hollands Kroon, opzettelijk en wederrechtelijk meerdere lampen en een kledingkast, toebehorende aan [benadeelde partij], heeft vernield door de lampen van het plafond te rukken en door de deur van de kledingkast af te rukken
> 2. hij op te Middenmeer, gemeente Hollands Kroon, opzettelijk mishandelend zijn vader tot wie hij in familierechtelijke betrekking stond, te weten [benadeelde partij], meermalen te slaan en te stompen en te duwen tegen het lichaam waardoor hij ten val is gekomen, waardoor deze pijn heeft ondervonden.

Figure 5.1: Different paragraphs referencing separate applicable laws.

predictions could be made on the sentence or paragraph level and combined to come to better predictions.

It should be noted however that between using shallow features and extensively mapping court case textual structure there is no semi-shallow approach that both better tests the importance of sentence structure and deals well with the constraint of time that is on the research in a master thesis. Future research could improve on operationalizing paragraph structure in court cases as was done in [1], but for the research at hand I am content with the balance struck.

5.1.2 Tagged entities

Of the features that were left out of the eventual model, the tagged entities represented another unique characteristic of court cases that could be modelled. In its current implementation, tagged entities had a great overlap with the n-gram feature and did not improve the F-score. But more than the mere usage of tags in court cases, a tagged entities feature can represent a semantic factor in the court cases that is otherwise not leveraged. Fragments like 5.2 are not uncommon and especially in court cases with trivial perpetrator, victim and object roles, we can imagine mapping relationships to roles and coming to an automatic understanding of the semantics of a crime.

Again, the consideration to not make use of semantics is based on the constraint of the master thesis scope. Nonetheless, the unique possibility that tagged entities offer for a deeper semantic analysis of legal data should be explored and could lead to better predictions of applicable laws for court cases.

> Uit de getuigenverklaring van [getuige] volgt dat verdachte [verdachte] in elk geval eind in haar woning aan de [adres] te Zoetermeer verbleef.

Figure 5.2: Use of tagged entities in court transcriptions.

5.2 Stacking classifiers

The iterative stacking of our best performing classifier was successful overall and showed that this approach can model label dependencies up to a certain degree. Initially, the approach was conceived as an alternative to Classifier Chains which had the limitation of becoming overly complex and expensive in a setting with a lot of possible labels.

5.2.1 An ontology of law

The stacking approach to predicting applicable laws consciously leaves out a component that dictates how labels relate. The reason behind this is that the study of the structure of the law is beyond the scope for this paper, would require involvement of legal experts and even then carries a predominantly qualitative essence. But it is interesting that beyond the conditional probabilities (Table 3.5), there are hierarchical relationships between articles of the law that could influence what we expect to apply in a given court case. As such, implementing a way to understand a structure of law, albeit through a network with nodes and edges or an ontology of the law, that carries weight in a predictive model would provide great insight into what constitutes a case in court and how this differs over time and between areas of jurisdiction. Even if the described implementation goes far beyond the sophistication of the research at hand, this thesis has provided some evidence that relationships within legislation do matter, and supports their relevance.

5.3 Results

The next chapter will conclude the results of this thesis by interpreting them as answers to our posed research questions. It is harder to see how the results can be generalized since there is no easy comparison to be made with previous research or real world situations. Some of the chosen directions in the method section also indirectly influence the numbers we presented in last chapter. It is important to understand these influences and the forthcoming knowledge that the retrieved scores house no inherent conclusions.

5.3.1 Data selection

The section on data selection features thorough motivation of the limits that were put on the data. However, in some cases my own discretion is of influence solely because no optimal limit can be found. Subsection 3.1.1 is an example of this when it is decided that we will only consider labels that are occur at least twenty times throughout the data collection. There is no trivial way to decide if this cut off point is high or low, but we can compare it to a similar paper [1] on predicting judicial court decision for court cases of the European Court of Human Rights. Here the authors opt to only consider articles that occurred eighty times or more, which led to only three laws being considered. The article reports an overall accuracy of 79 percent in predicting which law applies to a case, but with a most frequent class baseline of 43 percent this can be considered low.

Putting our results in perspective, I think we have modelled a system that is much closer to a real-world setting. In other words, the developed system could be accurate in predicting which laws apply for any new court case that is filed in the area of criminal law on Rechtspraak.nl [12].

5.3.2 Data quality

A study of data quality is obstructed by the required reading and annotation of all court cases. Here, time is not necessarily a constraint, but expertise certainly is since different views on court cases and the law may influence annotation. Instead, section 1.3 features an assumption that the paragraph on applicable laws (Figure 1.2) when present, is accurate and complete.

Although we have argued that testing the quality of the automatic annotation of court cases would be non-trivial and time consuming, this chapter allows us to look into some of the cases in which we can assume that applicable laws should co-occur, and critically analyze documents that do not mention both labels. We cannot quantitatively express inconsistencies in labelling documents without the aforementioned annotation, but expect that any court case in which qualified theft (article 311) or violent theft (article 312) applies, general theft (article 310) should also apply. Similarly, in any court case in which premeditated assault (article 301) or aggravated assault (article 302) applies, general assault (article 300) should also apply.

Table 5.1: Occurrences of labelsets.

Labelset	Number of articles
$311 \cup 312$	3.182
$(311 \cup 312) \setminus 310$	1.265

Labelset	Number of articles
$301 \cup 302$	1.172
$(301 \cup 302) \setminus 300$	761

Table 5.1 shows that the data is inconsistent in 39.7 percent of the cases in which theft applies, and in 64.9 percent of the cases in which assault applies. We can expect poor overall data quality having just cited these examples, which undermines the representability of our measures on precision, recall and F-score.

Ideally, a collection of court cases is accompanied by a gold label set developed by legal experts. In research cases where the aim is to showcase use of a distinct data domain, lessons can still be learnt using the label set gained from the court cases themselves. In fact, commenting on the quality of court case transcription and placing this as a connotation to predictions in a multi-label setting is a lesson in itself. The results on the task of multi-label classification suffer because of data inconsistency, but this in turn demonstrates that predicting applicable laws for court cases is very achievable using existing techniques and some novel ways to model label dependencies. In the next chapter we will elaborate more on the meaning of the reported results for the posed research questions.

5.3.3 Adding separate test data

The methods were tested using one collection of documents. Ideally, the best performing system would be developed using a separate collection of documents (development data) rather than the test data. The current approach has the drawback that the system could be tweaked to fit the test data specifically and that the results were not gathered from a purely blind test. However, not a lot of tweaking went into the development of the system in the first place. The feature sets were developed beforehand and so were the methods for stacking the classifiers. It is expected therefore that the chosen configurations are still the best settings for classifying new data.

6 Conclusion

In this final chapter I will summarize the work that went into this thesis and the results it yielded. More importantly, these results will be explained in the context of the research questions they aimed to answer. Although the previous chapter already disclosed some of the ways the current research could be extended upon, we will reiterate some of the possible courses for future work and its relevance for the field of computational linguistics and law.

6.1 Summary

Motivated by the increasing availability of large open textual data in the legal domain and its possibilities for natural language research, we posed three research questions in our introductory chapter:

1. *How can we predict applicable laws for transcribed court cases?*
2. *How effective are existing techniques for predicting applicable laws?*
3. *What can we learn from court cases as a data set to better predict applicable laws?*

The questions cover both the general aim of this thesis to reach high overall performance in a multi-labelling task (through RQ 1 and RQ 2) and a more specific ambition to study court case transcriptions and learn how legal data might differ from other NLP corpora (RQ 2 and RQ 3). We will sequentially answer the research questions in the following sections.

6.1.1 Predicting applicable laws as a multi-label classification task

The literature exhibited several existing techniques for handling multi-label classification. Specifically the Binary Relevance approach, which trains binary classifiers for each of the possible labels, has been used and its flexibility allows for adaptation to any multi-label task. One adaptation which respects label dependencies, feeds the prediction of one binary classifier to the next in a chain. Classifier Chains have limited potential for tasks where the amount of labels is very large (over 70 for applicable laws) since this minimizes the chance of an arbitrary label influencing the occurrence of the next. However, the notion that modelling label interdependence could benefit performance in some multi-labelling task and the inference that this would be the case for predicting articles of the law was important in designing the eventual predictive model.

A range of Binary Relevance Adaboosted decision trees greatly improved the baseline approach with an averaged precision of .8030 and recall of .6362. By extending the feature set with bigram and trigrams and the document type token ratio these were further improved to .8358 and .7170 respectively.

6.1.2 Leveraging the structure of law to better predict labels

Most of the features that were designed to leverage the distinct structure of court cases did not boost performance. As discussed in section 5.1, we do not take this to mean the the structure of court cases in unimportant but rather that the operationalization of structure features lacked the sophistication to leverage court case structure.

Performance was boosted, however, by iteratively stacking the Adaboosted decision tree with a feature that captured label probabilities predicted by its previous iteration. The averaged recall for all labels benefited from these stacks and this yielded and overall better performing system after three iteration, with an F-score of .7522.

It is non-trivial to compare these results to anything else, considering this thesis concerns a new field of research and we have remarked that data quality was sub-optimal (see subsection 5.3.2). But the boost in performance that was gained from using the label probabilities can serve as a demonstration of laws inherent structure and how it can easily influence predictive tasks such as ours.

6.2 Future work

This leaves the suggestion for future work to overcome some of the limitations to this master thesis. For one, a deepened semantic understanding of court cases, how roles in crimes relate, and a way to model this that is both applicable in a wide variety of court cases and advantageous to performance, would further benefit understanding of the importance of legal structure and allow for studies of how court cases differ both between fields of law and over time.

Predicting applicable laws for court cases is an interesting field by itself and one that could further aid the trend of digitalization and openness of government information. It would be interesting to see a predictive model capable of predicting applicable laws be used in a real-world setting. For instance, an initial step to filing a grievance can see a potential plaintiff enter a textual description of the relevant situation, suggest applicable laws, and have the plaintiff further explore these laws to learn if he has any legal ground in taking the situation to court.

6.3 Closing remarks

Our suggestions for future work show that there is lot left to explore and even more to learn. This is also true for the paper at hand. The questions that were raised at the beginning have been answered in its conclusion, yet more questions were raised along the way. It will be interesting to see how the use of computation linguistics in the legal domain evolves and I hope it is successful in providing insightful research that furthers both the field of information science and law.

Bibliography

[1] Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93, 2016. 6, 10, 14, 28, 30

[2] European Commission. European legislation on reuse of public sector information. https://ec.europa.eu/digital-single-market/en/european-legislation-reuse-public-sector-information, 2017. [Online; accessed 11-July-2017]. 1

[3] Richard de Mulder. Jurimetrics please! *European Journal of Law and Technology*, 1(1), 2010. 5

[4] Open Definition. Open Definition 2.1. http://opendefinition.org/od/2.1, 2017. [Online; accessed 11-July-2017]. 1, 2

[5] Ministerie van Binnenlandse Zaken Directie Interbestuurlijke Betrekkingen en Informatievoorziening. Naar toegankelijkheid van overheidsinformatie. beleidskader voor het vergroten van de toegankelijkheid van overheidsinformatie met informatie- en communicatietechnologie. https://zoek.officielebekendmakingen.nl/dossier/20644/kst-20644-30, 1997. [Online; accessed 11-July-2017]. 2

[6] Lee Loevinger. Jurimetrics–the next step forward. *Minn. L. Rev.*, 33:455, 1948. 5

[7] European Court of Human Rights. Hudoc database. http://hudoc.echr.coe.int/eng, 2017. [Online; accessed 13-July-2017]. 6

[8] Overheid.nl. Wetboek van strafvordering - artikel 348. http://wetten.overheid.nl/jci1.3:c:BWBR0001903&boek=Tweede&titeldeel=VI&afdeling=Vierde&artikel=348&z=2017-06-17&g=2017-06-17, 2017. [Online; accessed 13-July-2017]. 7

[9] Overheid.nl. Wetboek van strafvordering - artikel 350. http://wetten.overheid.nl/jci1.3:c:BWBR0001903&boek=Tweede&titeldeel=VI&

afdeling=Vierde&artikel=350&z=2017-06-17&g=2017-06-17, 2017. [Online; accessed 13-July-2017]. 7

[10] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011. 8

[11] De Rechtspraak. Kengetallen gerechten 2015. https://www.rechtspraak.nl/Organisatie-en-contact/Organisatie/Raad-voor-de-rechtspraak/Jaardocumenten, 2016. [Online; accessed 11-July-2017]. 2

[12] Rechtspraak.nl. Open data van de rechtspraak. https://www.rechtspraak.nl/Uitspraken-en-nieuws/Uitspraken/Paginas/Open-Data.aspx, 2017. [Online; accessed 26-July-2017]. 2, 9, 30

[13] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. A review of multi-label classification methods. In *Proceedings of the 2nd ADBIS workshop on data mining and knowledge discovery (ADMKD 2006)*, pages 99–109, 2006. 8

[14] Yale University. Conditional probability. http://www.stat.yale.edu/Courses/1997-98/101/condprob.htm, 1998. [Online; accessed 20-August-2017]. 17

[15] Marc Vilain. Language-processing methods for us court filings. In *LTDCA-2016. Proceedings of the Workshop on Legal Text, Document, and Corpus Analytics*, pages 81–90. University of San Diego Law School, 2016. 7

[16] Yi Yang. *Robust Adaptation of Natural Language Processing for Language Variation*. PhD thesis, Georgia Institute of Technology, 2017. 7, 14

A Perceptron classifier configuration

Table A.1: 3-layer perceptron configuration in Scikit-learn

Setting	Value
Hidden layer size	20
Solver	lbfgs
Maximum iterations	100
Random state	1

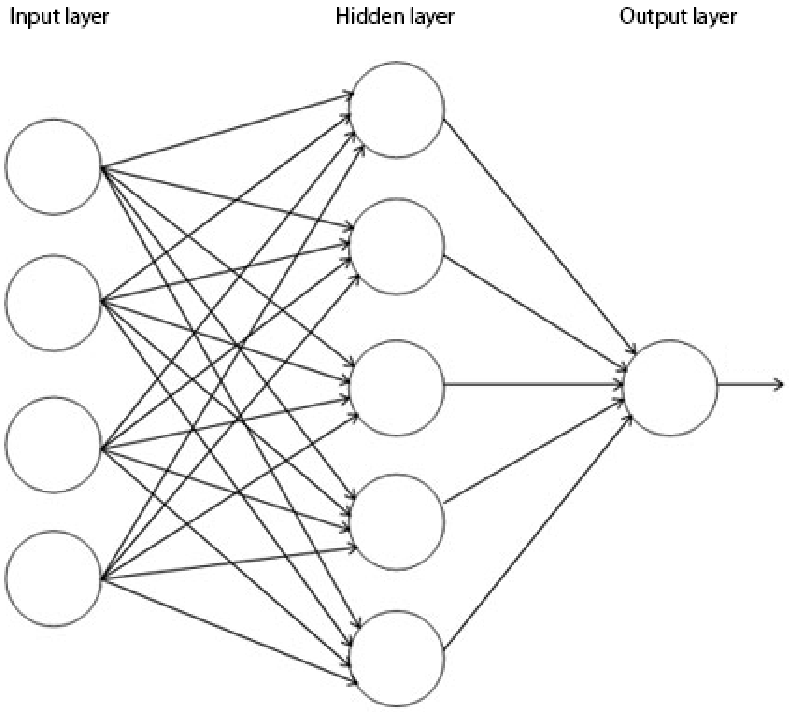


Figure A.1: 3-layer Perceptron Diagram

B Feature sets

Table B.1: Performance of the feature sets

Featureset	Precision	Recall	F-score
N-grams	0.8310	0.7146	0.7463
N-grams + adjective count	0.8103	0.6993	0.7345
N-grams + tagged entities	0.8339	0.7119	0.7460
N-grams + document length	0.8290	0.7155	0.7467
N-grams + type token ratio	0.8358	0.7170	0.7485
N-grams + number of paragraphs	0.8316	0.7153	0.7469
N-grams + court location	0.8309	0.7146	0.7463
All structure features (with n-grams)	0.8306	0.7168	0.7475
All content features (with n-grams)	0.8337	0.7012	0.7418
All features combined (with n-grams)	0.8334	0.7033	0.7430
Skipgrams	0.8213	0.6873	0.7273
Skipgrams + adjective count	0.8199	0.6835	0.7228
Skipgrams + tagged entities	0.8192	0.6846	0.7232
Skipgrams + document length	0.8213	0.6873	0.7273
Skipgrams + type token ratio	0.8208	0.6839	0.7251
Skipgrams + number of paragraphs	0.8204	0.6890	0.7287
Skipgrams + court location	0.8168	0.6857	0.7252
All structure features (with skipgrams)	0.8153	0.6821	0.7216
All content features (with skipgrams)	0.8195	0.6849	0.7249
All features combined (with skipgrams)	0.8136	0.6826	0.7224

C Fragment figure sources

Hyperlinks work for the digital version of this paper. ECLI documents can otherwise be retrieved by prepending the ECLI source with the 'http://deeplink.rechtspraak.nl/uitspraak?id=' snippet.

Table C.1: Figures and their sources.

Figure	ECLI source (URL)
1.1	ECLI:NL:RBDHA:2016:12314
1.2	ECLI:NL:RBDHA:2016:12314
2.1	ECLI:NL:RBNHO:2017:2181
2.2	ECLI:NL:RBNHO:2013:CA2169
4.2	ECLI:NL:RBZLY:2012:BX2192
3.3	ECLI:NL:RBAMS:2016:4123
5.1	ECLI:NL:RBALK:2012:BW8953
5.2	ECLI:NL:RBDHA:2015:3442