PREDICTION OF OUTCOMES IN HIGHER COURTS OF TURKEY USING NATURAL LANGUAGE PROCESSING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Emre Mumcuoğlu
April 2022

PREDICTION OF OUTCOMES IN HIGHER COURTS OF TURKEY USING NATURAL LANGUAGE PROCESSING

By Emre Mumcuoğlu

April 2022

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Memduh Haldun Özaktaş (Advisor)

Aykut Koç (Co-Advisor)

Tolga Çukur

Arzucan Özgür Türkmen

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

PREDICTION OF OUTCOMES IN HIGHER COURTS OF TURKEY USING NATURAL LANGUAGE PROCESSING

Emre Mumcuoğlu
M.S. in Electrical and Electronics Engineering
Advisor: Memduh Haldun Özaktaş
Co-Advisor: Aykut Koç
April 2022

The use of Natural Language Processing (NLP) in the field of law has become a topic of interest in the recent years. Applications to Turkish law, however, have remained unexplored to this day. In this thesis, first, a review of existing NLP applications in law is provided, and then, the problem of predicting Turkish court decisions is studied using NLP techniques. An extensive corpus that consists of case texts from Turkish higher courts, namely, the Constitutional Court and District Courts, is compiled. In addition, a numerical analysis and comparison of NLP methods at predicting the outcomes of these higher court cases is provided. The methods used for prediction are based on Decision Trees, Random Forests, Support Vector Machines and various deep learning models; specifically Gated Recurrent Units, unidirectional and bidirectional Long Short-Term Memory networks, and their attention-integrated counterparts. Prediction results for all algorithms are presented comparatively across all courts. The results show that decisions of Turkish higher courts can be predicted with high accuracy, especially with deep learning-based methods. Similar performance to existing work in the literature on case outcome prediction, which focus on different languages and different legal systems, is achieved.

Keywords: Natural Language Processing, Law, Machine Learning, Deep Learning, Artificial Intelligence and Law.

ÖZET

DOĞAL DİL İŞLEME YÖNTEMLERİ KULLANILARAK TÜRK YÜKSEK MAHKEMELERİNDE KARAR TAHMİNİ

Emre Mumcuoğlu
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Memduh Haldun Özaktaş
İkinci Tez Danışmanı: Aykut Koç
Nisan 2022

Doğal Dil İşleme'nin (DDİ) hukuk alanındaki uygulamaları araştırmacıların son yıllarda artarak ilgisini çekerken Türk hukuku için olası uygulamaları inceleyen bir çalışma bulunmamaktadır. Biz, ilk olarak DDİ'nin halihazırda literatürde olan hukuki uygulamalarını gözden geçiriyor, sonra da Türk mahkemeleri özelinde dava metinlerinden mahkeme kararını tahmin etme problemini ele alıyoruz. Bu tezin alana başlıca iki katkısından ilki, Türk yüksek mahkemelerinden Anayasa Mahkemesi ve İstinaf Mahkemelerinin karar metinlerinden kapsamlı bir derlem oluşturulmasıdır. İkinci katkısı ise, bu dava metinleri üzerinden mahkemenin kararını tahmin etmede çeşitli DDİ yöntemlerinin nicel olarak incelenmesi ve kıyaslanmasıdır. Karar tahmin etmede kullanılan algoritmalar Karar Ağaçları, Rastgele Ormanlar, Destek Vektörü Makineleri ve muhtelif derin ögrenme yöntemlerine dayanmaktadır. Sözü geçen derin ögrenme yöntemleri Geçitli Mükerrer Hücreler, tek yönlü ve çift yönlü Uzun-Kısa Vade Hafıza ağları ve bunların dikkat mekanizması eklenmiş çeşitleridir. Bütün bu algoritmaların karar tahminindeki performanslarını her bir mahkeme için mukayeseli bir biçimde ortaya koyuyoruz. Türk yüksek mahkemelerinin kararlarının özellikle derin öğrenme yöntemleri kullanılarak yüksek isabet oranlarıyla tahmin edilebileceğini gösteriyoruz. Diğer ülkelerin mahkemeleri için farklı dillerdeki dava metinlerinde yapılmış benzer çalışmalara yakın sonuçlar elde ediyoruz

Anahtar sözcükler: Doğal Dil İşleme, Hukuk, Makine Öğrenmesi, Derin Öğrenme, Yapay Zeka ve Hukuk.

Acknowledgement

I want to express my gratitude to my supervisors Haldun M. Özaktaş and Aykut Koç, who by their conception of the idea and by their valuable efforts throughout the last two years have made this thesis possible. I also want to express my gratitude to my teammate Ceyhun Emre Öztürk for his contributions towards this work.

I thank my friends for the fruitful exchange of ideas, and for their company during the direst of times.

I thank my family for their support and guidance at all times.

I also thank my university and my department for providing the environment and support that made this work possible.

This work is supported by TÜBİTAK within the scope of 2210/A scholarship, and in part by TÜBİTAK 1001 grant (120E346).

Contents

List of Figures

List of Tables

List of Publications

This thesis includes content from the following publication(s):

1. E. Mumcuoğlu, C. E. Öztürk, H. M. Ozaktas, and A. Koç, "Natural language processing in law: Prediction of outcomes in the higher courts of Turkey," *Information Processing & Management*, vol. 58, no. 5, p. 102684, 2021

Chapter 1

Introduction

This thesis tries to answer the question whether decisions of Turkish higher courts can be predicted using machine learning methods. In this thesis, in the light of common approaches in the literature, we formulate case outcome prediction as a classification problem. We build our models to process raw text from case descriptions to predict the outcome of a case. To make use of case texts in a machine learning setting, we deploy Natural Language Processing (NLP) techniques. As our work is the first of its kind in Turkish law, we aim to provide a baseline for more extensive future applications.

1.1 Motivation

Law is as old as civilization, and it has been constantly evolving to meet the changing needs of society. Systems of law need to adapt to the changes and developments in culture, politics, economy and technology. This naturally results in the rapid growth and change in legislation, and more so in the body of court rulings. The change is speeding up all the more as time advances. Both the transformation of legislation and the increase in the number of cases encumber a growing burden on professionals from judiciary, legislative, and even executive

institutions. The question hence arises whether any machine assistance can be of help in the field of law.

The most predominant legal systems around the world are civil law and common law. In the former, the main legal sources are usually a constitution and other laws made by the legislature, whereas in the latter, decisions of courts are primarily authoritative. Most legal systems rely on the combination of some form of legislation together with precedent court rulings. It is obvious that professionals need to investigate a large corpus of existing statutes and court decisions to act in accordance with the law and the principle of *stare decisis*, respecting precedents. Computers can aid in quickly scanning and analyzing such a vast amount of legal documents. Computerized systems can be utilized to retrieve laws that are relevant to a given case, decide whether a previous ruling is binding for the ongoing case, find precedent cases, analyze a case text by breaking it down into components, and perhaps aid judges in giving verdicts. By facilitating the job of prosecutors, lawyers, judges and other professionals, such systems may contribute to the greater good of the public by saving time, reducing error and improving consistency.

The logical and verbal nature of law makes it both a good target for computer involvement, but at the same time a challenging one to address. Natural Language Processing is well suited for legal applications since documents in this field are written entirely in natural language. NLP is a field at the junction of linguistics, statistics and computer science that tries to provide methods for computers to work on human language. NLP techniques are vital for converting legal texts into a machine readable form and thereafter analyzing them. In fact, law, from an NLP perspective has recently received the attention of researchers. A brief history and an overview of the existing approaches to the processing of legal documents is given in Chapter 2, with most of the focus on NLP applications. Some popular areas of interest in the literature are legal Named Entity Recognition [2, 3, 4, 5], legal feature annotation and classification [6, 7, 8], retrieving relevant law articles [9, 10, 11, 12, 13] and case outcome prediction [14, 15, 16, 17, 18, 19].

No such research has been done, however, to the best of our knowledge for

Turkish law. As an appropriate starting point, we have decided to address the case outcome prediction task. The aim is to predict the result of a case by looking solely at the fact descriptions written by the court. We use prevalent NLP methods to predict the outcomes. By doing so, we are hoping to lay a groundwork for future research in Turkish law, where new and more complicated models or low-level improvements can be tested against our results to see whether they provide actual improvement compared to our baseline methods.

1.2 The Case of Turkish Law

The Republic of Turkey is a unitary and constitutional state, whose legal system shares many features with those of continental Europe, including the adoption of a *separation of powers*. The three major *powers* are legislative, executive and judiciary. The Grand National Assembly of Turkey assumes the legislative power. The executive power is held by the president, together with his cabinet and other subsidiary directorates. The judiciary branch consists of independent courts. The Constitutional Court (*Anayasa Mahkemesi*) oversees the legislative branch and ensures its conformity to the constitution. Apart from the Constitutional Court and the Court of Jurisdictional Disputes (*Uyuşmazlık Mahkemesi*), Turkish courts can be grouped under two main categories by area of law: judicial courts and administrative courts. Both of these divisions exhibit a three-level hierarchy, namely, supreme courts, district courts and first instance courts. The supreme courts have the highest authority in appellate jurisdiction: Court of Cassation (*Yargıtay*) in judicial law, and Council of State (*Danıştay*) in administrative law. Below them are District Courts of Ordinary Jurisdiction (*Bölge Adliye Mahkemeleri* or *BAM*) and District Courts of Administrative Jurisdiction (*Bölge İdare Mahkemeleri* or *BİM*), respectively. District Courts evaluate appeals against first instance court decisions. Each higher court has the authority to modify or remove the decision of a lower court. The decisions of higher courts therefore tend to be binary, resulting in either the admission or rejection of the appeal, sometimes with partial decisions or modifications [20].

Since first instance court cases may have results that cannot easily be partitioned into discrete classes, and since the cases themselves are not fully available online, we have chosen to work on higher court cases. Namely, we work on decisions of the Constitutional Court and District Courts, for these are the higher courts that provide online case texts which are sufficiently long and explicitly organized in sections.

As mentioned before, District Courts are divided into District Courts of Ordinary Jurisdiction and District Courts of Administrative Jurisdiction. Courts falling into these two groups are further subdivided according to the area of law they are concerned with. The former group is divided into Civil Courts of Appeals (*BAM Hukuk Daireleri*) and Criminal Courts of Appeals (*BAM Ceza Daireleri*), and the latter is divided into District Administrative Courts (*BİM İdare Daireleri*) and District Tax Courts (*BİM Vergi Daireleri*). All these divisions comprise many regional departments. In our study, we do not distinguish between regional origins of case texts, and treat all decisions from a single division as one large corpus. As a result, we work on five corpora: one of Constitutional Court and four of District Courts (see Chapter 3 for details).

Since we work on higher court cases, we can formulate the problem of predicting the outcome as a binary classification problem. This is because higher court decisions are in the form of admission or rejection of the application. Individual applications to the Constitutional Court can result in either the violation of a constitutional right or no violation thereof. These results correspond, respectively, to the admission or the rejection of the application, thus forming our two classes. Decisions of District Courts result in either the removal or approval of the original ruling, corresponding to the admission or rejection of the appeal. It is commonplace that a higher court may approve the lower court's ruling, thereby rejecting the appeal, but still make minor corrections on the original decision. In our study, such decisions are still considered as rejection. However, there are partial decisions where some of the verdicts in the original ruling are approved, and the rest are removed. These decisions do not fall into either one of our binary classes. Such decisions are left out of our experiments in order to keep our work consistent with the literature, and to use binary classification metrics and

compare them to existing works. Chapter 3 details how we label court cases as one of two outcomes, and when we leave them out.

1.3 Our Contribution

The fundamental contributions of the work explained in this thesis are two. The first is the compilation of an extensive Turkish legal corpus for the first time that contains processed and labeled case texts. This corpus is made available for researchers[1] and is thus hoped to constitute a benchmark for future research in NLP in Turkish law.

The second major contribution of our work is the detailed comparison and evaluation of various NLP methods at case outcome prediction on multiple case text corpora. Our work is, to the best of our knowledge, the first where NLP is applied to Turkish law. For predicting the outcomes of cases, we utilize both traditional learning methods, namely, Decision Trees (DTs), Random Forests (RFs) and Support Vector Machines (SVMs); and deep learning methods, namely, Gated Recurrent Units (GRUs), unidirectional and bidirectional Long Short-Term Memory networks (LSTMs), and their attention-integrated counterparts. Our methodology is given in detail in Chapter 4. We have shown that decisions of Turkish higher courts can be predicted with high accuracy, especially with deep learning. We have achieved at the end of our experiments accuracy scores reaching 93%, and more importantly, F1 scores reaching 0.87, which are on par with the best results in the literature.

A lesser contribution of our work is related to experimentation methodology. The literature on predicting case outcomes is quite fragmented. Most of these studies are limited in scope as they usually deal with cases from one court and utilize a single machine learning algorithm. Given that the legal systems they

[1]The corpus that we have created is available at the GitHub repository "koc-lab/law-turk" [21].

study are already different, it is difficult to make any comparative studies or derive generalizations. Therefore, in this study, we have aimed to cover as many courts as possible and compare several machine learning methods. By doing so in a systematic and comprehensive way, our work should not only constitute a baseline for further studies of the Turkish legal system, but by characterizing classical machine learning methods and deep learning methods, it can also provide a framework within which other studies on different legal systems can be compared.

This thesis is based on work previously presented in [1].

Chapter 2

An Overview of Related Work

In this chapter, related work on the subject matter is discussed. We start from the historical origins by giving the earliest applications of artificial intelligence to the domain of law. We give an overview of NLP in law, especially in case outcome prediction, with a separate section for deep learning. The relevance of mentioned studies to our work is discussed in detail.

2.1 Origins

The use of artificial intelligence (AI) in law has a long history. The conception of the idea that these two fields could be brought together goes back to the 1970s. Buchanan and Headrick [22] speculated such a relationship, offering a multitude of areas where AI might contribute to law. According to Bench-Capon et al. [23], however, the birth of an active community of AI and law research is marked by the year 1987 when the first International Conference on AI and Law (ICAIL) was held. Early research focuses on exploring and utilizing logical structures in legal argumentation, also making use of a knowledge base consisting of legal cases. Finding and distinguishing precedents in legal discourse [24, 25] or factors that, for example, might favor a side [26] attracted research, motivated by *case based*

reasoning (CBR) systems [27]. These systems are designed to work based on a knowledge of previous cases, with a variety of possible applications. Extensive research has been done involving the design, improvement and utilization of CBR systems [27, 28, 29, 30, 23]. By being able to model legal arguments and making it possible to do better indexing and retrieval depending on the structures of cases, these models have many applications [31]. Similar rule-based approaches are used for evaluating cases and predicting court rulings [32, 33]. One special advantage is the interpretability of the results they provide. For further details on the topic, one may refer to the overview written by Bench-Capon et al. [23]. While similar research remains active, recent developments in NLP and deep learning have also found their way into the field of law, increasing the number of applications. The analysis of legal documents with NLP involves carrying common NLP methods into the legal domain and possibly further customizing and specializing them to better fit different tasks at hand.

2.2 Overview of NLP in Law

A primary problem is to extract features from a legal text. One such task is Named Entity Recognition (NER). NER systems specific to the legal domain have been studied in the literature [2, 3, 4, 5], including NER in Turkish legal texts at the time this thesis is published [34]. Another study which can set an example of utilizing previous knowledge and improving it on legal texts is the work of Elnaggar et al. [35]. The authors show that a transfer learning approach which incorporates training a Named Entity Linking system first on non-legal data, and then further training it on legal data, improves performance compared to solely training it on legal data. Although there are studies also on Turkish NER [36, 37, 38, 39], there exists no work specific to the legal domain.

Another important task in extracting features from legal texts is detecting word or sentence level law-specific features such as facts, obligations, prohibitions and principles [33, 40, 41, 6, 7]. For the annotation of legal features, supervised learning is commonly used, where every sentence in a text is manually annotated as

belonging to one or more of the given classes [33, 6, 7]. In the work of Ashley and Brüninghaus [33], case sentences are automatically represented by features called *factors* (parts of text that matter for the result) using a nearest-neighbor algorithm. A hand-coded algorithm decides on the labels of new cases, by comparing them to existing cases, with 92% accuracy in predicting the outcome. Shulayeva et al. [6] have used a multinomial Bayesian classifier to decide whether a given sentence contains a legal fact/principle or not, allowing the detection of facts and principles with 85% accuracy. Similarly, the work of Sleimi et al. [7] does not utilize any learning method, but achieves a 0.86 F1 score at classifying texts into one of seventeen classes (action, agent, condition, constraint etc.) using hard-coded decision rules that search for certain sequences of part-of-speech (POS) tags and specific words. Extracted features like those listed above can then be used for retrieval or reasoning systems [32, 42].

A different problem addressed in the literature is detecting the logical relations between texts, such as deciding whether a given plea is made based on the law. For instance, Nguyen et al. [8] have devised a deep learning model consisting of cascaded neural structures to break a sentence into its requisite and effectuation parts, and have achieved 0.78 F1 score on annotated test data. A second example of relations between texts that is considered in the literature is logical entailment. Finding such relations is addressed via modeling the task as a classification problem where a classifier decides, for a given pair of sentences, whether one entails the other. In the legal domain, models have been developed to decide whether there is an entailment between a given query and a law article. A direct application of this would be to automatically find and cite supporting law articles [9, 10, 11, 12, 13].

2.3 Case Outcome Prediction with Machine Learning

We now look at studies that are relevant to the main purpose of this thesis, namely, predicting legal case outcomes. Our prediction methods, which we comparatively evaluate as a second purpose of this thesis, are adopted from these similar studies. Aleven [32] and Ashley and Brüninghaus [33] have developed systems to predict the outcomes of cases by using rule-based algorithms that make their decisions based on the results of similar cases, and compared these algorithms to simple machine learning methods. Similar cases are retrieved by finding the nearest neighbors of a given case in terms of previously extracted legal features such as aforementioned *factors*. They both achieved accuracy scores reaching around 92%. The fact that the decision of a court can be predicted by using only a few parameters was demonstrated in *The Supreme Court Forecasting Project* [43, 44]. In this project, Decision Trees were used whose decision criteria were manually extracted from case descriptions. These trees were trained on a collection consisting of more than 600 cases. The simple and statistically determined decision hierarchy of these trees was able to predict case outcomes with 75% accuracy, and was sometimes able to surpass experts at predicting justice votes. More recently, a number of different legal systems and languages have been studied with the purpose of trying to predict court decisions using machine learning techniques [14, 15, 16, 17, 18, 19].

Traditional machine learning techniques using language features, usual word and n-gram frequencies, have proven quite useful for the case outcome prediction task. Katz et al. [14] consider more than 28,000 cases of the US Supreme Court, spanning a period of nearly two centuries, and aims to predict the outcomes both at case level and justice vote level. In their work, they train online-growing Random Forests on categorical variables that were partly obtained from a database and partly engineered. They achieve 70% accuracy at the binary classification task of predicting the outcomes of cases. Support vector machines (SVMs) have

also been shown to be successful in a similar task. Aletras et al. [15] have compiled a corpus consisting of cases of the European Court of Human Rights, using more than 500 cases related to articles 3, 6 and 8 of the European Convention on Human Rights. They attempted to classify cases as 'violation' or 'no violation' using solely textual features. Using most frequent n-grams of up to order four as features, they trained SVMs and achieved 79% average accuracy in this binary classification. Another example is the work of Şulea et al. [16] that used SVMs to predict outcomes of French Supreme Court cases. On a collection consisting of over 130,000 documents, they extracted unigrams and bigrams from each case, and assigned a corresponding label in terms of the result. They trained SVMs to predict the outcomes. In addition to their temporal analyses of the results, they report an overall accuracy reaching 97% and an F1 score reaching 0.97 on predicting case outcomes. However, Şulea et al. [16] study the problem without the usual binary classification (accept/reject) model but use multi-label classifications with 6 or 8 labels. Experiments have also been performed by Virtucio et al. [18] to test the performance of Random Forests and SVMs on predicting outcomes of Philippine Supreme Court cases. They were able to reach 59% accuracy with a Random Forest classifier.

2.4 Deep Learning in Law

A very powerful machine learning framework applied in NLP is deep learning, specifically Gated Recurrent Units (GRUs), Long-Short Term Memory networks (LSTMs) [45] and their variants combined with vector representations of words known as word embeddings [46, 47, 48]. Training word embeddings that are suitable to legal applications is a subject that needs to be addressed on its own. Chalkidis and Kampas [9] provide the first publicly available legal word embeddings called *law2vec*, which are trained on a large legislation corpus in English consisting of 492M tokens. They also provide an overview of the deep learning techniques used in the legal domain, focusing on three issues: text classification, information extraction and information retrieval. As for classification, based on the work of O'Neill et al. [41] that aims to classify sentence modality, Chalkidis et

al. [40] developed a state-of-the-art modality classifier using several LSTM based methods operating on law-specific word embeddings provided by Chalkidis and Androutsopoulos [49]. In another instance of text classification, Branting et al. [50] trained a neural model to predict administrative adjudications. Deep learning has also been utilized for information extraction in legal texts. Examples include recognizing parts of sentences that are labeled as requisite and effectuation [8] and the work of Chalkidis and Androutsopoulos [49] which focuses on extracting contract elements based on the dataset provided by Chalkidis et al. [51]. Yet another area that deep learning has proven successful in is information retrieval in the legal domain. Such applications include finding law articles related to a query [13, 11, 10, 12], matching cases with law provision [52] and finding fact assertions in cases related to a given query [53].

Deployment of deep learning for case outcome prediction has been introduced by Long et al. [17]. For this purpose, they have developed a prediction system called *AutoJudge*. This model consists of three bidirectional Gated Recurrent Units (BiGRUs) that each encode pleas, fact descriptions and relevant laws separately, a pairwise attention mechanism that they themselves have designed to turn these encodings into more meaningful representations, and finally a Convolutional Neural Network (CNN) layer that turns these representations into the final vectors used for classification. They trained this model and other simpler deep learning models on 100,000 divorce proceedings of the Supreme People's Court of the People's Republic of China for predicting whether they are granted divorce or not. They achieved the best results of 82% accuracy and 0.83 F1 score with *AutoJudge*. Kowsrihawat et al. [19] have utilized a similar system consisting of BiGRU encoders for encoding facts and laws, followed by further attention and hidden layers for case outcome prediction. They evaluated this model on a collection consisting of more than 1,000 cases of the Thai Supreme Court for binary classification. The best Macro-F1 score of 0.63 was obtained with this model.

2.5 Relevance to Our Work

Although sharing a common ground, aforementioned related works have their differences compared to each other in terms of the data and methods used. Ruger et al. [43] and Katz et al. [14], for instance, utilize categorical features and their derivatives, unlike others that use features like word or n-gram counts. Şulea et al. [16] work with six or eight class labels whereas all others formulate the task as binary classification. Long et al. [17] and Kowsrihawat et al. [19] take texts of relevant laws into account in addition to case descriptions provided by the court. Our work mostly shares common ground with these works. We aim to predict outcomes of cases by looking solely at case descriptions (and at relevant laws where available). We use textual features and classify cases into one of two possible results, admission or rejection. Although the works mentioned above may have some differing aspects, together they roughly constitute a criterion of success that we can compare our work to and draw conclusions from.

While these pioneering works provide proofs of concept of the viability of machine-learning-based prediction, most of them are limited in scope, as they usually deal with only one type of court or only a single algorithm, and thus provide a fragmented picture of what can be achieved. This can be better appreciated by examining Table 2.1 that summarizes previous results, which we observe to be sparse. One study that brings together data from a variety of courts and legal systems is the most recent work of Chalkidis et al. [54]. They have created a benchmark dataset called *LexGLUE*, incorporating existing NLP datasets concerned with various legal tasks in a single collection. LexGLUE includes only legal texts in English for ease of adaptation. In the present work, we also consider a variety of courts (Turkish courts in this case) and a variety of algorithms, and compare these with earlier studies. By providing a comprehensive study for the legal system of Turkey, we not only provide baseline results for this particular case, but also a framework for future studies.

Table 2.1: Summary of previous work (accuracy scores are in percentages). Scores in each work are obtained using a different collection of legal data from mentioned courts.

	Court	Machine Learning Methods							
		DT		RF		SVM		DL	
		Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
[15]	European Court of Human Rights					79.0			
[14]	US Supreme Court			70.2	0.69				
[19]	Thai Supreme Court						0.61		0.63
[17]	Supreme People's Court of People's Republic of China					55.5	0.56	82.2	0.83
[43]	US Supreme Court	75.0							
[16]	French Supreme Court					96.9	0.97		
[18]	Philippine Supreme Court			59.0		55.0			

Chapter 3

Creation of the Corpus

In this chapter we describe the Turkish legal corpus compiled for this study. We analyze the decisions of the Constitutional Court (*Anayasa Mahkemesi*), Civil Courts of Appeals (*BAM Hukuk Daireleri*), Criminal Courts of Appeals (*BAM Ceza Daireleri*), District Administrative Courts (*Bİ̇M İdare Daireleri*) and District Tax Courts (*Bİ̇M Vergi Daireleri*). The District Courts that are mentioned consist of many regional subdivisions. The corpus we have constructed spans all the local subdivisions, and is not restricted to one region or court.

The texts we have compiled are obtained from data made available online. Structures of the case texts differ for each court. The details will be given in this chapter, along with how they are pre-processed and labeled.

3.1 The Constitutional Court

The major duty of the Constitutional Court is the *a posteriori* constitutional review for newly made legislation. However, since 2010, individual applications regarding human rights violations have also been allowed. This has provided abundant court case data for our work. Furthermore, for all individual applications, in addition to the text of the court case, a convenient *case overview table*

including information on the topic, the verdict and relevant law is also provided.

In our study, all 6,485 court cases for individual applications to the Constitutional Court available at the time of our study were used. On the Constitutional Court website, these court cases are provided with case overview tables. These tables include the constitutional rights at stake, verdicts on whether there is a violation or not for each right, and the laws that are relevant to the case, together with a full textual explanation. Relevant law articles are later used as features for classification (see Section 4.1). The texts of the cases are processed as described in Section 4.1, and are also utilized for classification. These texts consist of five parts: a brief overview of the topic, details of the application, a description of facts, the review and the result. The first, second and third parts that constitute the case description were extracted from each text, and the rest (the decisions) were discarded to not reveal the result to our learning algorithms. Documents that could not be divided as such were omitted. Figure 3.1 shows the distribution of extracted case description texts according to their lengths in word counts. These case descriptions were used in training. The case outcomes are also provided in the case overview tables, removing any need to extract them from the text. This information was then used to label each case as "violation" or "no violation". Cases that resulted in the irrelevancy or ill-foundedness of the application or were filed after the statute of limitations deadline and therefore excluded from further consideration by the court were not used in our study.

Cases of the Constitutional Court can be subdivided into categories according to the constitutional right whose violation is being brought into question (it should be noted that a single case might appear in more than one of these categories if multiple rights are in question). When this categorization was done, a total of 41 categories emerged, 21 of which contain less than a hundred cases, and 7 which contain less than ten. With the further removal of cases that did not result in either "violation" or "no violation" from these categories, only 7 categories remained with a sufficient number of cases so that even when further partitions that are later described in this chapter are carried out, there exist samples from both "violation" and "non-violation" cases in each partition. The number of cases in these seven categories add up to a total of 2,673 (counting

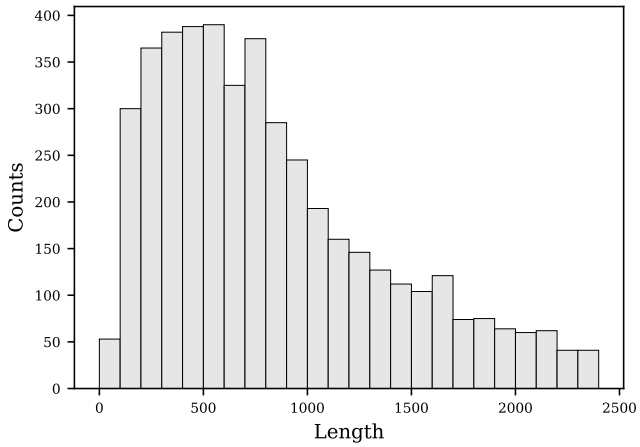


Figure 3.1: Histogram demonstrating lengths (in words) of the case descriptions for Constitutional Court cases.

duplicates if a case appears in more than one category). The numbers are low for some categories, but still enough to apply simple classification methods. Table 3.1 shows the number of court cases in each category.

A unified corpus was also created from the Constitutional Court cases to allow for further experiments by providing a larger collection. In this unified corpus, all cases that resolve in "violation" or "no violation" were brought together regardless of the constitutional right in question. Cases that brought multiple rights into question where each resulted in a different verdict were discarded (if a case in Table 3.1 appears in more than one category with different results, it is considered *mixed* and is discarded). These procedures led to a lower number of cases than the total number in Table 3.1. A collection consisting of 1,290 cases was obtained as a result of this compilation, 149 with no violation and 1,141 with violation. Table 3.2 gives a numerical overview of this collection.

Table 3.1: Number of "violation" / "no violation" Constitutional Court cases by constitutional rights discussed in the cases.

Constitutional Right	Violation	No Violation
Right to Equitable Trial	138	27
Freedom of Expression	155	32
Right to Trial within a Reasonable Time	987	49
Property Right	371	102
Right to Respect for Private and Family Life	192	70
Right to Access to Courts	203	43
Right to Personal Freedom and Security	196	108

3.2 Civil Courts of Appeals

The cases of District Courts do not contain readily extracted features such as an overview table or a list of relevant laws, which was the case with the Constitutional Court. Their documents also do not follow a strict pattern as there are many regional divisions. Thus, working on these cases is more complicated and harder than working with those of the Constitutional Court. However, there are still certain common keywords (or keyphrases), usually written in all capitals or as separate titles, that mark where the description of the facts ends and the justification for the decision begins. We search for these keywords and divide the document into two from the first line where one of these keywords occur. The part before the keyword is used as the case description (used for training), and the part after the keyword is used as the case decision (used for label extraction). There is no exact, non-changing set of such keywords, and the keywords we use are of our choice. The set of used keywords might change with respect to legal corpus or time. If none of the keywords are found in a document, the document is deemed unsplittable and is discarded. From those that are successfully split, a label is extracted from the case decision part with another keyword search. If label extraction fails, those documents are discarded as well. We choose the splitting and labeling set of keywords from fundamental expressions denoting an assessment or verdict. Our choice of keywords is such that most of our case

Table 3.2: Number of successfully split and labeled documents and their labels for all courts. Only rejected or admitted cases are considered in our study. For the Constitutional Court, rulings of "violation" are listed under admitted and of "no violation" are listed under rejected. (843 cases of the Constitutional Court that are not listed here have other results such as "Irrelevancy". Some of the cases that are listed as mixed and discarded here may appear in constitutional rights-based categories).

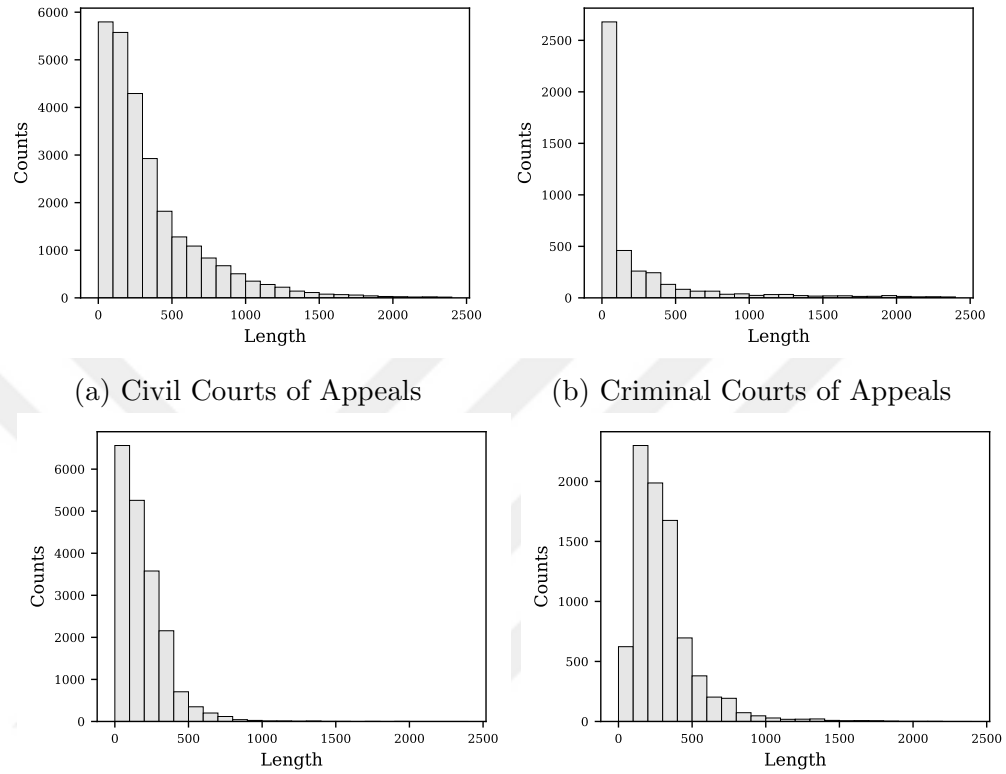
Court	Input Total	Split & labeled	Rejected	Admitted	Mixed
Constitutional Court	6,485	4,973	149	1,141	2,840
Civil Courts of Appeals	47,796	26,327	12,519	8,702	5,106
Criminal Courts of Appeals	9,241	4,385	1,891	420	2,074
District Administrative Courts	20,948	19,046	6,851	966	11,229
District Tax Courts	8,870	8,302	3,276	559	4,467

documents can be split and labeled successfully (see Table 3.2).

For the Civil Courts of Appeals, all 47,796 available case documents were used. The documents were split from the first occurrence of one of the following keywords: *'GEREKÇE'* (justification), *'KARAR'* (decision), *'DEĞERLENDİRİLMESİ / DEĞERLENDİRME'* (assessment), *'HÜKÜM'* (verdict), such that the rest would be kept hidden from the machine learning models. Figure 3.2 presents the lengths of case description texts extracted from the District Courts cases. Then, if the remainder of the text (the decision) contained one of the following keywords, the case was labeled as such: *'REDDİ'* (rejection), *'KABULÜ / KALDIRILMASI'* (admission). Cases with mixed/ partial decisions were discarded (in our circumstance those that include both of these keywords). If a document failed to meet these structural patterns (such as not containing the keywords) they were omitted as well. The resulting number of documents are listed in Table 3.2 with their corresponding labels.

3.3 Criminal Courts of Appeals

All 9,241 case documents from the Criminal Courts of Appeals were used. The same keywords were used for the split with the addition of *'GEREĞİ*



(a) Civil Courts of Appeals

(b) Criminal Courts of Appeals

(c) District Administrative Courts

(d) District Tax Courts

Figure 3.2: Histograms demonstrating lengths (in words) of case description texts for (a) Civil Courts of Appeals, (b) Criminal Courts of Appeals, (c) District Administrative Courts and (d) District Tax Courts.

DÜŞÜNÜLDÜ' (a verdict has been reached). When labeling, documents containing *'YER OLMADIĞINA'* (unjustifiable) were also considered rejected. Cases that could not be split or had partial decisions were omitted. The resulting number of documents can be seen in Table 3.2.

3.4 District Administrative Courts

All 20,948 case documents from the District Administrative Courts were processed. These documents follow a different structure. They begin with a description of facts, followed by the phrase *'TÜRK MİLLETİ ADINA'* (in the name of the Turkish People) and the decision and the reason thereof. These documents

were split from this point, and the preceding parts (case description) were used for training. For the labels, the same keywords as above were sought in the remaining portion of the text. The documents that could not be split or labeled were omitted. The resulting number of case documents can be seen in Table 3.2.

3.5 District Tax Courts

All 8,870 case documents from the District Tax Courts were used. The splitting and labeling process is the same as the one for administrative courts. The resulting number of documents can be seen in Table 3.2.

3.6 Overview

After all the steps described above were carried out for each court, the resulting corpora were finally split into training, validation and test sets. We use 60% training, 20% validation, and 20% test data splits for the constitutional rights-based corpora (to avoid having too few examples of a class in the validation and test sets), and 70%, 15%, 15% splits for other corpora (unified Constitutional Court corpus and District Courts corpora). An overview of the statistics of the created corpora after all the splitting, labeling and partitioning can be seen in Table 3.3.

Table 3.3: Overview of created corpora. For the Constitutional Court, rulings of "violation" are listed under admitted and of "no violation" are listed under rejected.

Corpus	Training	Validation	Test	Rejected	Admitted	Total
Right to Equitable Trial	99	33	33	27	138	165
Right to Freedom of Expression	112	37	38	32	155	187
Right to Trial within a Reasonable Time	621	207	208	49	987	1,036
Property Right	283	94	96	102	371	473
Right to Respect for Private and Family Life	157	52	53	70	192	262
Right to Access to Courts	147	49	50	43	203	246
Right to Personal Freedom and Security	182	60	62	108	196	304
Constitutional Court (unified)	902	194	194	149	1,141	1,290
Civil Courts of Appeals	14,854	3,183	3,184	12,519	8,702	21,221
Criminal Courts of Appeals	1,617	347	347	1,891	420	2,311
District Administrative Courts	5,471	1,173	1,173	6,851	966	7,817
District Tax Courts	2,684	575	576	3,276	559	3,835

Chapter 4

Methodology

In this chapter, our data preparation procedure, prediction methods and evaluation metrics are described.

4.1 Data Preparation

Case texts in our corpora differ in length, as demonstrated in Figures 3.1 and 3.2. Traditional machine learning methods, however, demand feature vectors of fixed length extracted from each document. For this purpose, word (unigram) frequencies were used as features to represent case texts. To extract the word frequencies, first, each case text was tokenized and stemmed using the Turkish NLP tool *Zemberek* [55]. To stem the words, this tool uses a hand-crafted algorithm as opposed to methods such as that in the work of [56], which are also suitable for agglutinative languages such as Turkish. Numbers, dates etc. were dropped so that only words remained. A vocabulary was created from these words, and very rare words whose number of occurrences throughout the corpus lay below a threshold of 50 were removed. This threshold is chosen by inspection, so that the words that lie below this threshold are mostly proper nouns. Each case text is thus turned into a sequence of stemmed tokens.

Let $D = \{s_1, s_2, ..., s_{\mathrm{N}}\}$ denote a corpus, and $V = \{w_1, w_2, ..., w_{\mathrm{M}}\}$ denote the vocabulary created from that corpus as described above. Each text in D is a sequence of tokens $s_n = \left(u_i\right)$ where $u_i \in V$. To represent each text numerically, vectors of word counts were created. The length of these vectors is equal to the size of the vocabulary, and their entries correspond to each word's number of occurrences in that text. In other words, if the number of occurrences of a word w in a text s is denoted by $\#_w(s)$, each text s_n was then represented by the word count vector

$$\mathrm{c}_n = \begin{bmatrix} \#_{w_1}(s_n) \\ \#_{w_2}(s_n) \\ \vdots \\ \#_{w_{\mathrm{M}}}(s_n) \end{bmatrix}$$

of fixed size $\mathrm{M} = |V|$. It was therefore possible to store the data from a corpus in a single matrix

$$\mathrm{C} = \begin{bmatrix} \mathrm{c}_1{}^{\mathrm{T}} \\ \mathrm{c}_2{}^{\mathrm{T}} \\ \vdots \\ \mathrm{c}_{\mathrm{N}}{}^{\mathrm{T}} \end{bmatrix}$$

of size $|D| \times |V|$. The sizes of the vocabularies created from each corpus can be seen in Table 4.1. One can observe that the sizes of these matrices are therefore very large.

On the Constitutional Court website, cases are provided together with a list of relevant laws in *case overview tables*. A total of 3,528 law articles are mentioned in all cases of the Constitutional Court. For each case, relevant articles were extracted from these case overview tables, and for each corpus, a set $L = \{l_1, l_2, ..., l_{\mathrm{P}}\}$ of all mentioned law articles was created. The texts of the law articles themselves were not used as they were not readily available. Each case's relevant law was then represented by a one-hot encoded vector, with one entry for each law article (whether it is relevant or not). In other words, if λ_n is the set of law articles mentioned in a case, that case's relevant law was represented

by the vector

$$\mathrm{r}_n = \begin{bmatrix} \mathrm{I}\left(l_1 \in \lambda_n\right) \\ \mathrm{I}\left(l_2 \in \lambda_n\right) \\ \vdots \\ \mathrm{I}\left(l_{\mathrm{P}} \in \lambda_n\right) \end{bmatrix}$$

of size $|L|$, where l_p is any article in L, and I is an indicator function. This procedure was followed for the rights-based Constitutional Court corpora. The sizes of the created law vectors for each of these corpora can be seen in Table 4.1. These one-hot encoded law vectors were appended at the end of previously created word count vectors. Each rights-based corpus was thus represented by an extended matrix

$$\mathrm{C} = \begin{bmatrix} {\mathrm{c}_1}^{\mathrm{T}} & : & {\mathrm{r}_1}^{\mathrm{T}} \\ {\mathrm{c}_2}^{\mathrm{T}} & : & {\mathrm{r}_2}^{\mathrm{T}} \\ & \vdots & \\ {\mathrm{c}_{\mathrm{N}}}^{\mathrm{T}} & : & {\mathrm{r}_{\mathrm{N}}}^{\mathrm{T}} \end{bmatrix}$$

of size $|D| \times \big(|V| + |L|\big)$. The same procedure was not followed in the unified Constitutional Court corpus to allow for comparison with District Courts corpora.

All these matrices were then standardized elementwise (by mean and variance) so that frequent words do not dominate, and to allow for calculation of sample covariance matrices (later mentioned in this section). Elementwise standardized matrices are denoted henceforth by $\bar{\mathrm{C}}$.

Since the vocabulary sizes are quite large, the resulting feature vectors were of very high dimensions, especially when compared to the small number of samples in some corpora (compare Table 3.3 with Table 4.1). However, most of the data can be correlated or uniform across cases. Therefore, the same information could possibly be represented using a reduced number of dimensions. To shrink the dimensions of the data matrices as much as possible, *Principal Component Analysis (PCA)* was performed. More precisely, each corpus data was represented by a new approximate matrix $\widehat{\mathrm{C}}$ that is the projection of the original data onto the principal eigenvectors of the sample covariance matrix $\Sigma = \bar{\mathrm{C}}^{\mathrm{T}}\bar{\mathrm{C}}$. Here, *principal eigenvectors* refer to the eigenvectors with largest corresponding eigenvalues

Table 4.1: Dimensions before and after PCA is applied.

Corpus	Vocabulary Size	Law Vector Size	Dim. after PCA
Right to Equitable Trial	4,252	245	38
Right to Freedom of Expression	7,159	143	24
Right to Trial within a Reasonable Time	7,300	1,051	120
Property Right	5,407	685	61
Right to Respect for Private and Family Life	5,012	318	29
Right to Access to Courts	4,295	384	73
Right to Personal Freedom and Security	6,566	198	18
Constitutional Court (unified)	10,606	n/a	334
Civil Courts of Appeals	12,946	n/a	587
Criminal Courts of Appeals	6,041	n/a	119
District Administrative Courts	8,123	n/a	607
District Tax Courts	4,795	n/a	250

(variances). The numbers of these principal components were chosen separately for each corpus such that 95% of the variance in the data would be preserved. This indeed yielded significant dimensionality reduction. Table 4.1 shows the original vocabulary and law vector sizes, and the resulting dimensions after PCA is applied.

For deep learning, on the other hand, most modern NLP applications take advantage of distributional representations of words, also known as *word embeddings* [46, 47, 48]. Word embeddings are trained on large corpora to place the words in a semantic vector space. The underlying assumption (named the *distributional hypothesis of linguistics*) is that the meaning of a word, to some degree, can be inferred from its statistical co-occurence with other words.

There are different methods for embedding words in a vector space such as *word2vec* [47], *GloVe* [48], or even into uncertain Gaussian distributions instead of point vectors such as in [57], or [58] which extend the idea to multimodal distributions. Word embeddings have performed well on evaluation tasks such as semantic similarity tasks [46, 59], and they have extensive applications such as detecting semantic change over time [60], analyzing and removing bias [61, 62] (including legal text [63, 64]). Imparting interpretability to the structure of these vector embeddings is also an active research area [65, 66, 67]. There is ongoing

research in these areas concerning Turkish word embeddings as well [61, 66].

To make use of word embeddings, we only performed tokenization and removal of non-words on case texts. Stemming, however, was not performed, although it decreases the size of the vocabulary. Here, unlike our previous methods, the complexity of deep learning models do not depend on the size of the vocabulary. More importantly, in agglutinative languages like Turkish, stemming a word may strip it from some of its meaning and grammatical function, both of which can be important in sequential models. Take the Turkish word "yargılanmadığını" for example, meaning "that he has not been tried". Stemming this word simply results in "yargı" (judgment). Although both of these words are represented by embeddings of the same size, the latter has lost some of its original function. Not stemming the words therefore ensures better quality of word representations, without suffering from increased dimensionality.

After tokenization was done, each token was replaced by 400-dimensional word2vec embedding vectors [47] trained on Wikipedia articles in Turkish, obtained from [68]. Cases in D were now represented therefore by sequences of vectors $\mathrm{e}_n = \left(\vec{\mathrm{v}}(u) \right)_{u \in s_n}$ where $\vec{\mathrm{v}}(\cdot) \colon V \to \mathbb{R}^{\mathrm{d}}$ denotes the function that maps tokens to their respective word embeddings.

For most of the court case corpora that we have created, the number of cases are relatively small, considering the large amount of parameters that deep learning models bring about. To address this issue, we enhanced the training sets by breaking each case text into 100-word chunks, and including these new chunks of text as additional training samples for our models, along with samples from the original training splits mentioned in Chapter 3. This data augmentation procedure increased the number of samples massively and slowed the training down. By simulating a larger collection of texts, although slowing down the training, it is aimed that the models would be forced to predict outcomes by looking at different parts of texts, instead of possibly inclining toward one irrelevant word or phrase in each text, which is likely given that our corpora are not very large. It is thus intended that in a way, this would prevent the models from *overfitting* the data. A more detailed discussion of the effects of this operation will be given

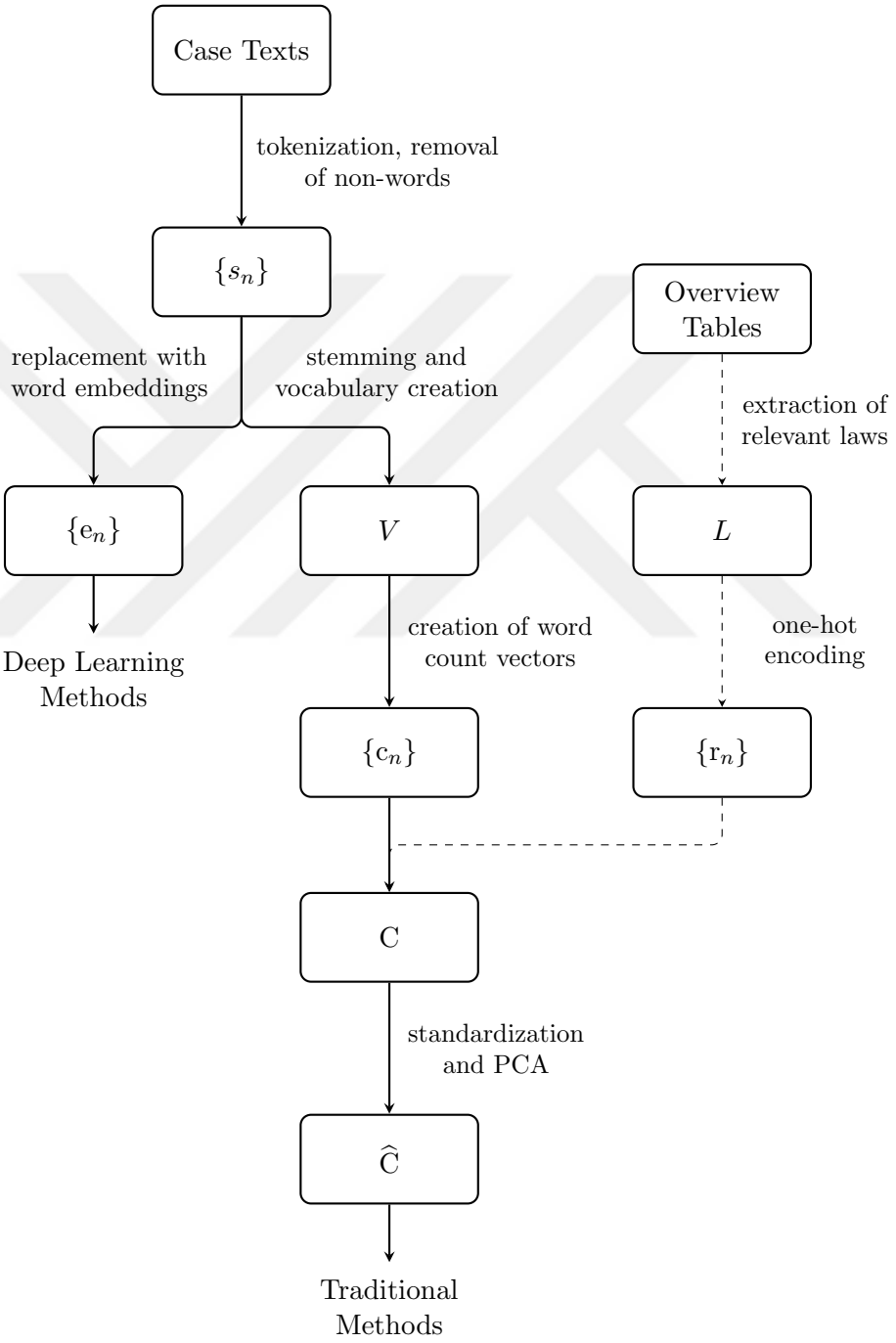


Figure 4.1: Summary of the data preparation process (dashed lines only for rights-based corpora).

in Chapter 5.

As for the labels, we simply labeled accepted (violation) cases with 1, and rejected (no violation) cases with -1. We denote the labels of each case with $y_n \in \{-1, 1\}$.

4.2 Classification Methods

The most prevalent methods in the literature for predicting case outcomes are Decision Trees, Random Forests and Support Vector Machines (SVMs) [14, 15, 16, 69].

4.2.1 Decision Tree

Decision Trees are structures wherein the output for a data sample is decided as a result of successive decisions. This decision process naturally constitutes a tree, where branches represent possible results of decisions. At each node, therefore, the data is split into subsets with respect to a decision, and subtrees are grown on these subsets in a similar fashion. A leaf node represents the end of this decision process, and has an associated output. Decision Trees can be used as a classification method, and can be grown from training data using appropriate decision rules. Decision Trees in such tasks are usually built top-down heuristically by splitting the data at decision nodes, usually according to either information gain or an impurity criterion, so that decisions split the data in groups that are as homogeneous as possible. Decision Trees make it possible to use the most important features of the data in order, which are ordered according to the increase in information or purity [70].

Ruger et al. [43] have shown that Decision Trees with even a few nodes can be effective at predicting decisions of the US Supreme Court. However, unlike our work where automatically extracted features are used (see Section 4.1), they

worked on manually crafted features and achieved 75% accuracy. We have therefore incorporated Decision Tree as our first learning algorithm in this thesis. On each corpus, a separate classification tree was grown using extracted features ($\widehat{\mathrm{C}}$) described in Section 4.1.

At each node of a tree, a split is to be made based on one feature. Since our features are numbers rather than a set of discrete values, our decisions take the form of setting a threshold, and splitting the data according to whether the selected feature is smaller or greater than this threshold (see Figure 4.2). Thus, if the data at a node is denoted by a set of indices $S = \{n \leq \mathrm{N}\}$, the choice of a feature k and a threshold β partitions the set into two sets:

$$S^- = \{n \in S : \ \widehat{\mathrm{C}}_{nk} \leq \beta\}$$
$$S^+ = \{n \in S : \ \widehat{\mathrm{C}}_{nk} > \beta\}$$

We use information gain as the measure of goodness of such a split, equivalently minimizing the expected entropy of the subsets created as the result of the split. The entropy of the class labels in a set is calculated as

$$\mathrm{H}(Y \mid S) = \sum_{y \in \{-1,1\}} -\mathrm{P}(y \mid S) \log_2 \mathrm{P}(y \mid S)$$

where Y is a Bernoulli random variable, and the probabilities $\mathrm{P}(y \mid S)$ are empirically calculated probabilities obtained by dividing the number of y in S by the cardinality of S. Lower entropy therefore means a more homogeneous set, where most of the corresponding cases are of the same class. The task is then to find the best decision criteria k and β for maximum information gain

$$\hat{k}, \hat{\beta} \ = \ \arg\max_{k,\beta} \ \mathrm{H}(Y \mid S) - \mathrm{H}(Y \mid S^{\pm})$$

or equivalently minimum entropy

$$\begin{aligned} \hat{k}, \hat{\beta} \ &= \ \arg\min_{k,\beta} \ \mathrm{H}(Y \mid S^{\pm}) \\ &= \ \arg\min_{k,\beta} \ \mathrm{P}(S^-)\, \mathrm{H}(Y \mid S^-) + \mathrm{P}(S^+)\, \mathrm{H}(Y \mid S^+) \end{aligned}$$

since $\mathrm{H}(Y \mid S)$ is independent of the choice of k and β. The reader should note that although in Figure 4.2 the decision boundaries are shown to be in between

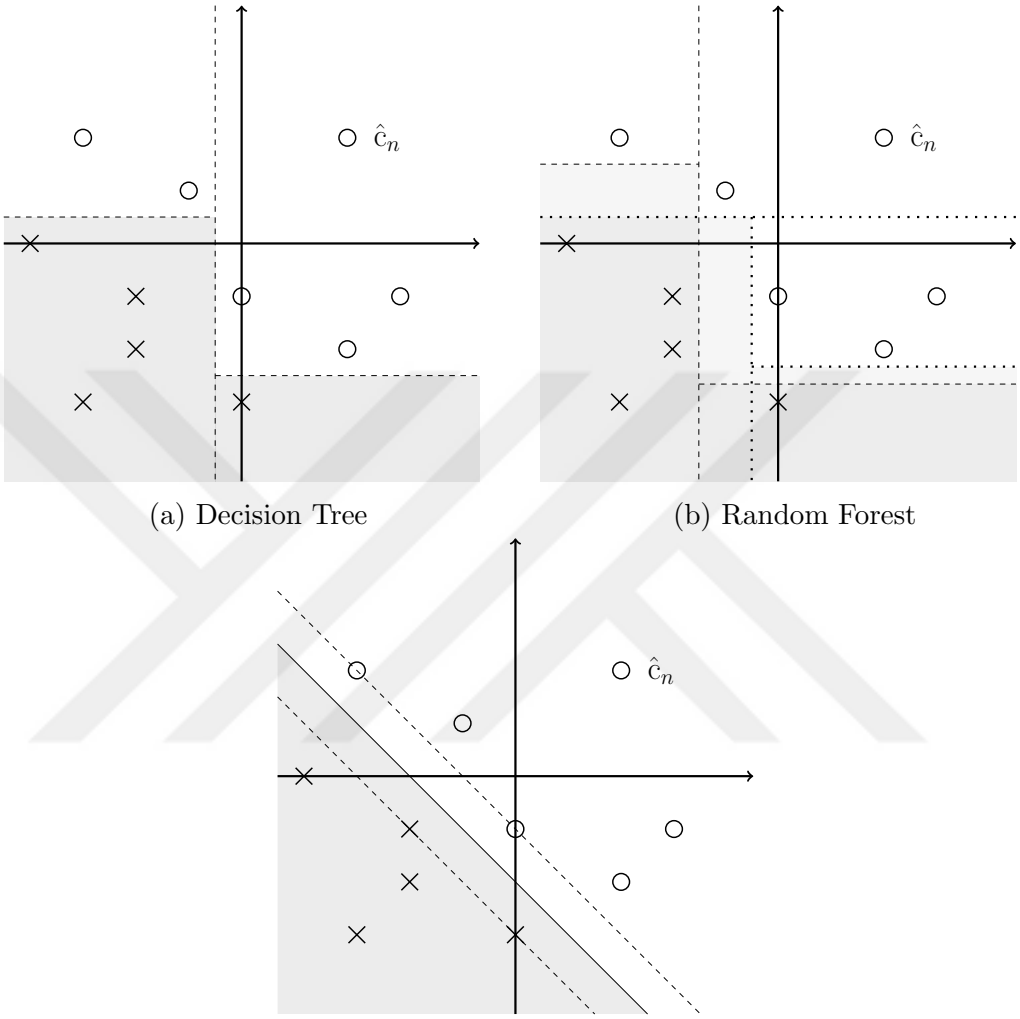
data points for clarity, the only meaningful decisions are made exactly at the sample points. It is of course impossible for a computer to try and evaluate every real number on a given dimension, and it is sufficient to evaluate as many splits as there are samples. The complexity of this operation therefore scales linearly with the number of samples. After the optimal decision criteria are found, the data is split accordingly into two subsets, and the same procedure can then be followed for both of these subsets, further splitting them until a completely pure node is reached or only a single point remains.

We use the *scikit-learn* implementation of Decision Tree [71], which allows us to use entropy as a homogeneity measure, and also lets us choose the minimum number of samples at leaf nodes. Although one can choose to grow the full tree, as the number of samples in nodes start to decrease, the decision splits start to become unreliable as they are made based on only a few samples. It is good practice therefore to stop earlier, and the growth can be limited by the minimum number of allowed samples per leaf (among others). We tune this parameter, based on the validation results, effectively pruning the tree if necessary. If, as a result, a leaf node is not pure anymore, the classification label can be decided by the majority class in that node.

4.2.2 Random Forest

A Random Forest is an ensemble learning technique, consisting of many decision trees that vote on the result (see Figure 4.2 for an illustration). Random Forests are widespread in NLP [72, 73, 74], and the reason thereof is that they, like Decision Trees, offer a way to classify texts considering only the most important features. In the legal domain, as mentioned earlier, Random Forests are used to predict the decisions of the US Supreme Court in the work of Katz et al. [14], and were proven to be useful with an accuracy of 70%.

A single decision tree is sensitive to the data it is trained on. Small changes in the data can alter all the structure of the tree, especially if this change affects the topmost splits in the tree where the best feature is only slightly better than the



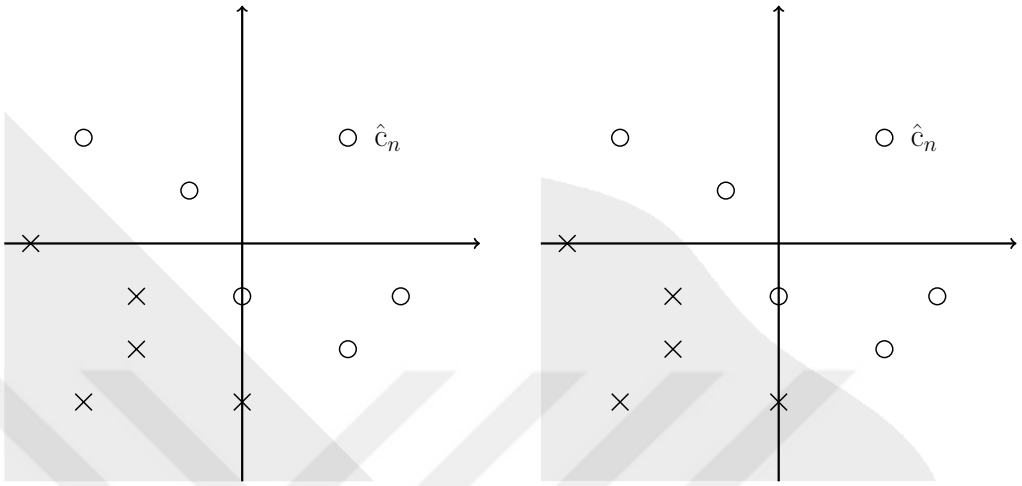
(a) Decision Tree

(b) Random Forest

(c) Support Vector Machine

Figure 4.2: Illustration of traditional classification methods.

second closest candidate. Random Forests aim to remedy this issue by growing multiple trees. The essential idea is that a compound of independent trees will likely outperform any single one of its constituents. The reason that training a forest does not simply yield a number of identical trees has to do with how the data is fed into each tree and how these trees handle the features. Each tree randomly samples, with replacement, elements from the dataset. The tree is then grown on this random subset of samples. Furthermore, each tree also randomly selects a subset of features to grow on. This prevents the ensemble from overestimating the importance of a single feature when making its decisions. This way, even if



(a) linear kernel (b) RBF kernel

Figure 4.3: Illustration of SVM decision boundaries with linear and Gaussian RBF kernels.

each tree has a probability of error, the overall probability of error will diminish multiplicatively [75, 69].

The training procedure of each tree is otherwise identical to the procedure described in the previous section. We again use the scikit-learn implementation [71], which requires the same parameters as decision trees, with the addition of number of trees in the forest. This implementation draws N random data samples for each tree by default, and the number of features that each tree considers is the square root of the number of all features. We therefore only tune the number of trees, based on validation performances, in addition to the minimum allowed number of samples per leaf.

4.2.3 Support Vector Machine

SVM is a classification method that is widely used for both general text classification and in the legal domain [76, 74]. SVMs are studied in the legal text processing domain in the works of Aletras et al. [15] and Şulea et al. [16]. The former achieved 79% accuracy on the European Court of Human Rights, and the latter achieved 97% accuracy on the French Supreme Court.

SVM tries to find the hyperplane that best separates data from different classes, the best being the one that has the largest distance to the nearest samples from each class (see Figure 4.2). The decision boundary thus takes the form $\omega^{\mathrm{T}}\mathrm{x} + b = 0$, characterized by ω and b. To the aim of separating the two classes, whether a sample lies on the correct side can be expressed compactly for both classes as $y_i\,(\omega^{\mathrm{T}}\mathrm{x} + b) \geq 0$. To also maximize the decision margin, the *Hinge loss* $\max(0,\, 1 - y_i\,(\omega^{\mathrm{T}}\mathrm{x} + b))$ is used, which is positive even if the sample is on the correct side but $\omega^{\mathrm{T}}\mathrm{x} + b \leq 1$. Or equivalently, it is positive when the sample x is not at least a distance of $\frac{1}{\|\omega\|}$ far away from the decision boundary. Since this margin is to be maximized, the objective is to also minimize $\|\omega\|$. However, if $\|\omega\|$ is chosen arbitrarily small, the Hinge loss will be non-zero for all samples. On the contrary, the Hinge loss can be made arbitrarily small by choosing $\|\omega\|$ large enough. A combination of these two objectives is therefore minimized, combined by a regularization parameter κ. Precisely,

$$\min_{\omega,b} \quad \|\omega\|^2 + \kappa \sum_{n=1}^{\mathrm{N}} \max(0,\, 1 - y_i\,(\omega^{\mathrm{T}}\hat{\mathrm{c}}_n + b))$$

is the objective that is to be minimized (minimized over principle component projections $\hat{\mathrm{c}}_n$ in our setup) [77, 69].

There is one extension of SVMs to data that is not linearly separable, and that is to transfer the data to another space where ordinary dot products ${\mathrm{x}_i}^{\mathrm{T}} x_j$ are replaced by a kernel function $k(\mathrm{x}_i, \mathrm{x}_j)$. If the *radial basis function (RBF)* kernel is used, for instance, dot products in the dual of the above problem are replaced by $k(\mathrm{x}_i, \mathrm{x}_j) = \exp(-\gamma\|\mathrm{x}_i - \mathrm{x}_j\|^2)$ [69]. Figure 4.3 illustrates example decision boundaries for the ordinary linear kernel and the RBF kernel. We use the scikit-learn implementation of SVMs [71]. We tune the regularization parameter κ, depending on the performance achieved on validation sets, and choose a kernel function from linear and RBF kernels. This implementation sets γ in the RBF kernel to $\frac{1}{(\mathrm{N}\,S^2)}$ by default (S^2 denoting sample variance).

Decision Trees, Random Forests and SVMs are trained on principle component projections of word count vectors (see Section 4.1 for all details). Parameters are tuned for each corpus separately by inspecting the validation accuracy of the

models. All models are trained with balanced class weights to prevent them from biasing towards the majority class.

4.2.4 Deep Learning

Deep Learning based methods are also widely used for text classification where there are sufficient data [78, 74, 79]. The usual approach is to replace words with their word embeddings, and then use a deep neural network, usually recurrent models, to process this information [17, 80]. Then a simple neural network classifies texts based on extracted features. Long Short-Term Memory Networks (LSTMs) [45] or Gated Recurrent Units (GRUs) [81], which are simpler alternatives to LSTMs, and their bidirectional variants are the most well-established approaches to the processing of text. These gated models, especially LSTMs, are well suited for the processing of long texts due to their internal memory mechanisms that help keep important information throughout the process [78]. However, for particularly long texts such as the ones in the legal text corpora we collected in this work, a state-of-the-art improvement is the use of attention mechanisms [82]. With this approach, either words themselves or the internal states of the neural units are weighted with an attention score and summed to obtain a final representation of the text (the case document in our situation). Attention mechanisms, although increasing the parameter count, allow distinguishing important parts of text, as well as ensuring better gradient flow during training.

Long et al. [17] have utilized deep neural networks with attention mechanism to predict case outcomes of the Supreme People's Court of People's Republic of China. They have achieved 82% accuracy and a 0.83 F1 score.

In our work, we apply GRUs, LSTMs and bidirectional LSTMs (BiLSTMs) as representative deep learning methods. Two alternatives for each method are considered, with and without the attention mechanism. In the case without attention, the sequences of embedding vectors representing cases (see Section 4.1) are fed into these models, and the classification is made by a dense layer (with

softmax activation) based on the output of the recurrent model. The following equations summarize this procedure

$$\begin{aligned} \mathrm{h}_t &= \mathrm{GRU}(\mathrm{e}_{n,t},\, \mathrm{h}_{t-1}) \\ \mathrm{p} &= \mathrm{softmax}(\mathrm{W}\,\mathrm{h}_{-1}) \end{aligned}$$

for GRU,

$$\begin{aligned} \mathrm{h}_t, \mathrm{c}_t &= \mathrm{LSTM}(\mathrm{e}_{n,t},\, \mathrm{h}_{t-1},\, \mathrm{c}_{t-1}) \\ \mathrm{p} &= \mathrm{softmax}(\mathrm{W}\,\mathrm{h}_{-1}) \end{aligned}$$

for LSTM, and

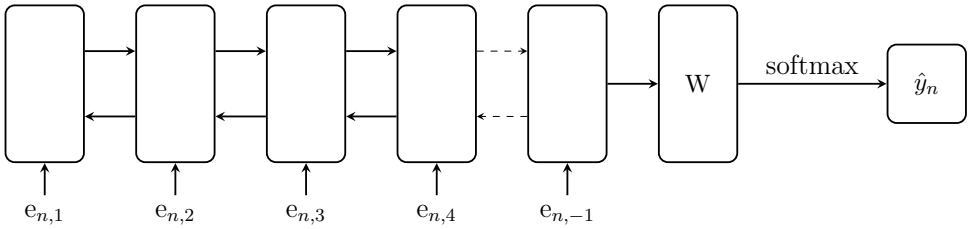
$$\begin{aligned} \mathrm{h}_t^f, \mathrm{c}_t^f &= \mathrm{LSTM}(\mathrm{e}_{n,t},\, \mathrm{h}_{t-1},\, \mathrm{c}_{t-1}) \\ \mathrm{h}_t^b, \mathrm{c}_t^b &= \mathrm{LSTM}(\mathrm{e}_{n,t},\, \mathrm{h}_{t+1},\, \mathrm{c}_{t+1}) \\ \mathrm{h}_{-1} &= [\mathrm{h}_{-1}^f; \mathrm{h}_1^b] \\ \mathrm{p} &= \mathrm{softmax}(\mathrm{W}\,\mathrm{h}_{-1}) \end{aligned}$$

for BiLSTM, where $\mathrm{GRU}(\cdot)$ and $\mathrm{LSTM}(\cdot)$ denote individual cells of each architecture, and h_{-1} denotes the last index.

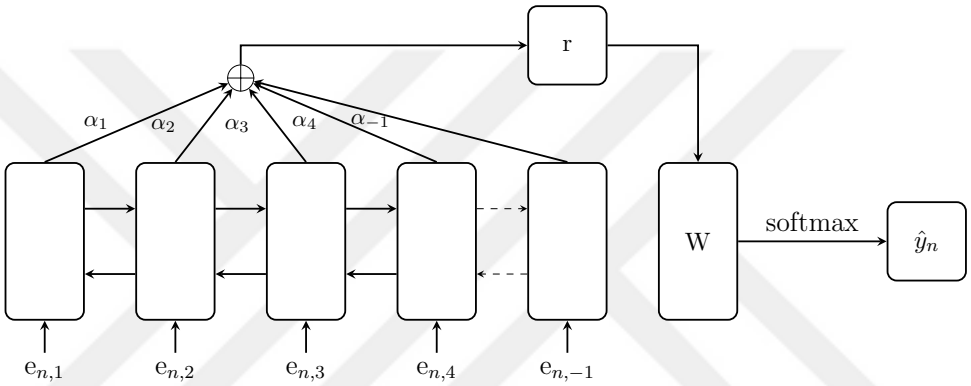
For the implementation of attention, since our problem is a classification problem with many-to-one architecture, it is enough to use a simple attention mechanism such as in [83]. The following equations describe this mechanism:

$$\begin{aligned} \mathrm{m}_t &= \tanh(\mathrm{h}_t) \\ \alpha_t &= \mathrm{softmax}(\mathrm{w}^{\mathrm{T}}\,\mathrm{m}_t) \\ \mathrm{r} &= \sum_t \alpha_t\,\mathrm{h}_t \\ \mathrm{p} &= \mathrm{softmax}(\mathrm{W}\,\mathrm{r}) \end{aligned}$$

This mechanism replaces the last layer of the previous models in the above equations. The attentive models therefore work on a weighted global sum of internal states, instead of only the last state. This is to facilitate classification on long case documents. As a result, there are a total of six models, three with attention



(a) without attention



(b) with attention

Figure 4.4: Illustration of the difference between deep learning models with and without attention.

and three without. Figure 4.4 shows the difference between models with and without attention.

The classification is then made depending on the output of the softmax layer:

$$\hat{y}_n = \begin{cases} +1 & \mathrm{p}_2 > \mathrm{p}_1 \\ -1 & \text{otherwise} \end{cases}$$

Since the output of the softmax layer represents class probabilities for "admit" and "reject", a document is simply labeled with the result that has the higher probability.

4.3 Evaluation Metrics

We use the most common evaluation metrics in the literature, the first one being accuracy [15, 14, 17, 16], defined as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN},$$

where TP, TN, FP and FN denote the numbers of true positives, true negatives, false positives and false negatives, respectively. However, most of our data is heavily imbalanced as can be seen in Table 3.3, and in some cases, achieving 80% accuracy is trivial. We therefore used two more metrics that take imbalance into account. The first one is balanced accuracy (BACC), which is the average of per-class accuracies defined as

$$BACC = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right)$$

where P and N denote the total numbers of positives and negatives. The other is the F1 score, which is commonly used in the literature [40, 35, 17, 10, 7, 16], defined as

$$F1 = 2\,\frac{Precision \cdot Recall}{Precision + Recall}.$$

However, F1 score assumes a positive and a negative class, and it does not make sense in our case to arbitrarily call a class positive, as it would depend on which one of the parties' perspective one looks from. We therefore used macro-averaged F1 score (MA-F1) as in the literature [19], which is the mean of per-class F1 scores that is calculated by using both of the alternatives.

Chapter 5

Results and Discussion

In this chapter, the results of our experiments are presented.[1] Reported scores were obtained from test runs over unseen data. Scores were compared to a baseline that only makes random guesses by using class label weights based on their frequency in the training set.

5.1 Results of Experiments

The first experiments for deep learning were done using data where the augmentation procedure described in Section 4.1 is not carried out. In that case, validation scores were at the vicinity of other methods, and the training that took orders of magnitude longer did not show any significant improvement in return. Especially for models without attention, gradients vanishing after a few hundreds of words would prevent getting the best out of training. Therefore, data augmentation was done by breaking each case text into 100-word chunks as described in Section 4.1. All of the reported results are for experiments on augmented data.

In the first experiments using the Constitutional Court corpus, cases for each

[1]The data and the code used to obtain these results are available at the GitHub repository "koc-lab/law-turk" [21].

Table 5.1: Constitutional rights-based classification results of Constitutional Court cases.

Constitutional Right	Method	ACC(%)	BACC(%)	MA-F1
Right to Equitable Trial	Baseline	72.0	72.0	0.42
	DT	78.9	72.9	0.73
	RF	78.9	60.0	0.60
	SVM	89.5	80.0	0.84
Freedom of Expression	Baseline	62.0	37.5	0.38
	DT	63.0	50.0	0.46
	RF	81.5	60.4	0.59
	SVM	81.5	60.4	0.59
Right to Trial within a Reasonable Time	Baseline	87.8	47.2	0.47
	DT	84.0	62.0	0.55
	RF	90.4	47.6	0.47
	SVM	92.9	49.0	0.48
Property Right	Baseline	68.1	47.6	0.48
	DT	50.7	55.0	0.40
	RF	50.7	36.6	0.36
	SVM	89.6	57.6	0.58
Right to Respect for Private and Family Life	Baseline	55.0	49.0	0.46
	DT	42.4	36.5	0.37
	RF	69.7	67.3	0.64
	SVM	63.6	46.3	0.46
Right to Access to Courts	Baseline	65.8	35.7	0.40
	DT	75.0	60.3	0.58
	RF	72.2	50.3	0.50
	SVM	88.9	60.0	0.64
Right to Personal Freedom and Security	Baseline	52.2	48.7	0.49
	DT	38.5	40.6	0.38
	RF	38.5	44.2	0.32
	SVM	53.8	56.4	0.54

constitutional right were considered separately. Since the number of cases in each category is not enough to train neural networks, only Decision Trees, Random Forests and SVMs were used for prediction and thus a comparison to deep learning models is not available. For these models, validation macro-F1 score was monitored as a success criterion. Parameters of models were tuned according to macro-F1 score to ensure a fair comparison. The test results are shown in Table 5.1. Since the number of instances in each category is very low, the results are not balanced and no one classification algorithm stands out clearly. Also, the prevalence of balanced accuracy scores close to 50% despite high accuracy scores may indicate that the classifiers are biased towards the majority class, and cannot detect instances of the smaller classes, which is "no violation". The high accuracy scores, therefore, do not say much about the selectivity of the classifier. This can be seen more clearly if one compares these to the numbers in Table 3.1 where the accuracies are almost consistent with the majority class ratio. The balanced accuracy and F1 scores, however, indicate at least some success. SVM has proven, in general, to be more useful than Decision Tree and Random Forest, since it can handle small data better. The inadequacy in scores is an indication of the insufficiency of the size of each dataset, where a prediction is being made with very few samples that have high dimensional features even after dimensionality reduction.

Observing the inadequacy of using a very small corpus, as a next step, we conduct experiments on the unified Constitutional Court corpus. All classifiers are trained on this unified corpus. Furthermore, relevant law articles were not used as features in this experiment to allow for comparison with the District Courts and to allow training of the deep learning based methods. This makes the task purely text-based unlike in the previous case, where we used one-hot encoded feature vectors corresponding to law articles in addition to textual features. Then, the same data preparation procedure that was performed for District Courts is applied. The results of prediction on the test set for all methods are shown in Table 5.2. The highest accuracy of 91.8% and F1 score of 0.67 is obtained with the LSTM model with attention. Models with attention have outperformed their

Table 5.2: Classification results on unified Constitutional Court corpus and District Courts corpora.

Court	Method	ACC(%)	BACC(%)	MA-F1
Constitutional Court (unified corpus)	Baseline	79.4	50.7	0.51
	DT	85.1	60.7	0.62
	RF	87.6	56.3	0.57
	SVM	83.5	59.9	0.61
	GRU	87.6	56.3	0.57
	GRU + attention	89.2	60.0	0.61
	LSTM	83.0	61.3	0.62
	LSTM + attention	91.8	64.2	0.67
	BiLSTM	89.7	54.6	0.56
	BiLSTM + attention	90.2	57.7	0.59
Civil Courts of Appeals	Baseline	50.8	49.2	0.49
	DT	61.5	60.3	0.60
	RF	68.7	67.3	0.67
	SVM	64.7	64.1	0.64
	GRU	66.8	63.5	0.64
	GRU + attention	66.7	66.1	0.66
	LSTM	66.7	65.7	0.66
	LSTM + attention	65.9	64.5	0.65
	BiLSTM	69.0	67.7	0.68
	BiLSTM + attention	67.3	65.0	0.65
Criminal Courts of Appeals	Baseline	69.5	48.6	0.49
	DT	82.4	75.0	0.73
	RF	81.8	74.6	0.73
	SVM	80.1	71.2	0.70
	GRU	82.4	75.6	0.74
	GRU + attention	82.7	75.8	0.75
	LSTM	85.0	77.8	0.77
	LSTM + attention	85.6	76.6	0.77
	BiLSTM	82.4	71.0	0.72
	BiLSTM + attention	82.4	74.0	0.74
District Administrative Courts	Baseline	78.9	51.0	0.51
	DT	86.3	73.0	0.72
	RF	86.7	74.3	0.73
	SVM	83.2	78.7	0.72
	GRU	90.0	71.3	0.74
	GRU + attention	89.6	69.2	0.72
	LSTM	90.1	75.2	0.77
	LSTM + attention	90.8	70.9	0.75
	BiLSTM	89.3	69.1	0.72
	BiLSTM + attention	91.1	72.8	0.76
District Tax Courts	Baseline	76.7	53.8	0.54
	DT	89.9	83.6	0.81
	RF	89.4	83.3	0.80
	SVM	92.4	90.0	0.86
	GRU	91.8	81.1	0.84
	GRU + attention	92.0	79.8	0.83
	LSTM	92.9	89.3	0.87
	LSTM + attention	92.9	84.3	0.86
	BiLSTM	91.7	78.8	0.82
	BiLSTM + attention	93.2	80.6	0.85

counterparts without attention. This is expected because cases of the Constitutional Court are quite long.

Then, the same experiments were performed for all District Courts. The results are reported in Table 5.2. For the Civil Courts of Appeals, 69.0% accuracy and a 0.68 F1 score are observed. In this largest corpus we have, the additional parameters that BiLSTMs bring seem to be useful since there are enough data to train on. On Criminal Courts of Appeals cases, 85.6% accuracy and a 0.77 F1 score are obtained. These highest results are achieved with the help of deep learning. Random Forest has overall performed better than SVM for these courts. On District Administrative Courts cases, 91.1% accuracy is obtained, together with an F1 score of 0.77. Again, deep learning models have mostly outperformed other methods in these scores. Finally, for cases of the District Tax Courts, for which the highest scores are obtained, an accuracy of 93.2% and an F1 score of 0.87 is achieved at best.

5.2 Discussion and Implications

In order to facilitate cross-comparison and provide an overall view of our results, the performance over all courts are tabulated in Table 5.3. On the deep learning column, we report the best out of all deep learning models. When an average is taken over the best results for each court, an average accuracy of 86.1%, average balanced accuracy of 75.7%, and average F1 score of 0.75 are obtained. These average results are on par with results obtained by other works in the literature reviewed in Section 2. The best scores we have obtained (accuracy 93.2%, balanced accuracy 90.0%, F1 score 0.87) surpass most earlier work.

Among all the experiments, the ones that utilize deep learning methods have overall proven to be more useful. It seems that these models can perform at least as well as others because they can capture dependencies between words in addition to information embedded in each word. When we look at previous work, it is difficult to reach a definitive conclusion as to whether there is a similar

Table 5.3: Overall results (accuracy scores are in percentages, F1 scores are macro averaged).

Court	Machine Learning Methods									
	Baseline		DT		RF		SVM		DL	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Constitutional Court	79.4	0.51	85.1	0.62	87.6	0.57	83.5	0.61	**91.8**	**0.67**
Civil Courts of Appeals	50.8	0.49	61.5	0.60	68.7	0.67	64.7	0.64	**69.0**	**0.68**
Criminal Courts of Appeals	68.5	0.49	82.4	0.73	81.8	0.73	80.1	0.70	**85.6**	**0.77**
District Administrative Courts	78.9	0.51	86.3	0.72	86.7	0.73	83.2	0.72	**91.1**	**0.77**
District Tax Courts	76.7	0.54	89.9	0.81	89.4	0.80	92.4	0.86	**93.2**	**0.87**

pattern. Long et al. [17] have obtained higher scores with their proposed deep learning model when compared to the other methods that they use. The difference is not as clear when compared to other works. Also, it can be seen that the results Kowsrihawat et al. [19] have obtained with deep learning are lower compared to other works. It is likely that there are other factors at play here. Works using high-level features or features specifically designed to capture word relations might perform better even with simpler classification methods. We argue though, by performing many experiments on cases of different courts, and by comparing methods trained on simple word features, that deep learning is overall more reliable in the processing of legal text for case outcome prediction.

Comparing deep learning models, we find that GRUs, being the simplest ones, generally performed worse. It is known that LSTMs are better for processing longer texts [78] and they are the ones that performed the best in our experiments. Although BiLSTMs, being more complicated, require the training of almost twice the parameters, they are the best performers in large corpora. For instance, in the Civil Courts of Appeals, which is our largest corpus, the BiLSTM performed the best as there were lots of data to train on. In other courts, however, where the corpora are smaller, BiLSTMs seem to be excessively complex to do reliable training. The use of such complex models can only be justified with the availability of large collections of data. This is the case for the work of Long et al. [17] where they use three bidirectional GRUs, but still obtain very good results, probably due to the size and quality of their data. Another observation

that needs to be made from our results is that the attention mechanism almost always improves the performance. Although attention itself requires the training of more parameters too, it can be said confidently that such a mechanism is essential for processing very long texts such as ours, as they provide a way of retaining information throughout and allow smoother training with propagating gradients where even LSTMs are not enough.

Some important factors that affect the results other than the actual methods used are statistics such as the lengths of case texts or number of samples in the corpora. In the literature, work that has used larger corpora have achieved better results. Examples are the works Şulea et al. [16] and Long et al. [17], which do training on around 130,000 and 100,000 case documents, respectively, (see Table 2.1). One should be aware, however, that it may not be appropriate to make direct comparisons between works dealing with different countries and languages. In our work, interestingly enough, the scores for the courts with shorter case texts and with less numbers of cases are higher. Scores for Constitutional Court cases, which are the longest, and cases of the Civil Courts of Appeals, which are the largest in number, are lower. Even though there is larger data to train on, these results might be an indication that to capture information in these longer texts, even more training data is required. It can be seen clearly that the performances of models suffer from working on very long texts, and while improvements such as the attention mechanism or bidirectional variants try to remedy this issue, an even larger corpus, if it was available, would perhaps be able to further elevate the performance of these models.

The numerical results that we have obtained have also implications beyond a simple comparison of algorithms. They tell us not only which algorithm performs better, but also which set of cases are easier or more difficult to predict. This, in turn, can give insight into the nature of cases in that set and the structure of the court they belong to. The performance for the Civil Courts of Appeals is considerably lower compared to the other district courts. This partly might have to do with the more balanced number of class labels (see Table 3.3), as the percent accuracy is naturally lower on a more balanced dataset, compared to a set where a single class dominates and high accuracy is trivial to achieve.

However, this explanation does not apply to the other scores which are designed to be less affected by class imbalance. Despite a larger corpus being available and used for the Civil Courts of Appeals, which, all else being equal should lead to better prediction performance, the opposite is observed. What could be the reason for this unexpected result? One factor that should not be overlooked is that the texts of the Civil Courts of Appeals cases are longer compared to the cases of other district courts (Figure 3.2). However, there may also be reasons related to the nature of the law and its enforcement. By nature, the Civil Courts of Appeals deal with civil cases that are very general and indefinite in terms of their content whereas the Criminal Courts of Appeals and District Tax Courts deal with more structured laws and cases. For example, the universal principle known as *the principle of no punishment without law* says that an act needs to be explicitly defined as a *crime* in the law for it to be punishable. Thus, the number of crimes are definite and their nature well defined. On the other hand, an indefinite number of civil cases can be built on an unpredictable variety of interactions among people and institutions. While the number of civil law articles and principles to be applied indirectly to them are finite, the potential situations to be judged are far from being predefined. Administrative law, while not being as structured as criminal law, nevertheless deals with a narrower scope of issues than civil law and thus may be argued to be more predictable. While these comments are preliminary and would require further work to substantiate, the fact that certain structural and content differences in different courts and the cases they deal with are reflected in prediction performance is meaningful. It highly suggests that the success rates of the algorithms are not merely about processing some texts regardless of content, but that the algorithms are actually doing something that is related to and that reflects the content of the texts, and perhaps more significantly the nature of the cases and the structure of the legal system.

Predicting the outcome of cases based on their descriptions can have a number of applications, some of which are bound to lead to ethical considerations and controversies. Attorneys may use this predictive tool to "weigh" the case in order to get a sense of how similar cases have been decided previously. Prosecutors

may use it to judge whether a case is worth pursuing; thereby concentrating their efforts on more promising cases. The use of an automated prediction tool by judges would probably be the most controversial one. Should they use it as an aid in making decisions, or should they not even see the result of such predictions as it may bias them towards a particular decision? Should such tools be limited to obtaining aggregate statistics for evaluative purposes, rather than making decisions in individual cases? Apart from the application to the practice of law, these algorithms and results can shed light on the working of the legal system, the extent to which it is consistent, and to understand whether it is aligned with our sense of fairness and justice. These are open problems for future research.

Chapter 6

Conclusion

Legal case outcome prediction is a machine learning and natural language processing application in law which has not received attention in the context of the legal system of Turkey. We have systematically studied the problem of predicting outcomes for court rulings in Turkey. Almost all possible court types have been studied in detail. Whereas almost all earlier work had used a single machine learning method, we have reported the results of several methods comparatively for several courts. Thanks to this breadth, we believe it will provide a reference point and baseline for further studies in this area.

The results show that higher court rulings in Turkey can be predicted with good accuracy, as shown by several alternative measures. Direct comparison with previous work is not possible because earlier systems were developed for other languages and very different legal systems. Yet, this work should contribute to setting general baselines. Among the various methods considered, deep learning has overall yielded the highest prediction scores.

There are several technical issues that may constitute the content of future work, such as more detailed feature extraction and sentence-level supervision for systems that are not end-to-end. Further experiments and research on the retrieval of precedent cases can also be addressed in the context of Turkish law.

Our results have implications beyond comparing algorithms and demonstrating their predictive power. There is a variation in results obtained for different courts, which has interesting potential interpretations. More work is needed to uncover the meaning of this difference but we hypothesize that it is related to the different content and structure of cases of different types of courts. One possibility is that certain courts have more predictable results because of the nature of the data or bookkeeping that is not substantially related to the content of the proceedings or the structure of the law. However, the results point to possible interpretations that predictability may be related to the actual content and structure. This in turn suggests that what the algorithms are doing goes beyond merely processing symbols or text, but is about the content of the case, and perhaps even about the structure of the legal system.

We also discussed some of the practical applications, and the legal and ethical implications of the use of such machine-based predictive systems.

Bibliography

[1] E. Mumcuoğlu, C. E. Öztürk, H. M. Ozaktas, and A. Koç, "Natural language processing in law: Prediction of outcomes in the higher courts of Turkey," *Information Processing & Management*, vol. 58, no. 5, p. 102684, 2021.

[2] P. H. L. de Araujo, T. E. de Campos, R. R. R. de Oliveira, M. Stauffer, S. Couto, and P. Bermejo, "LeNER-Br: a dataset for named entity recognition in Brazilian legal text," in *International Conference on the Computational Processing of Portuguese (PROPOR)*, pp. 313–323, Springer, 2018.

[3] C. Cardellino, M. Teruel, L. A. Alemany, and S. Villata, "A low-cost, high-coverage legal named entity recognizer, classifier and linker," in *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law (ICAIL 2017)*, (New York, NY, USA), pp. 9–18, Association for Computing Machinery, 2017.

[4] C. Dozier, R. Kondadadi, M. Light, A. Vachher, S. Veeramachaneni, and R. Wudali, "Named entity recognition and resolution in legal text," in *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*, pp. 27–43, Berlin, Heidelberg: Springer-Verlag, 2010.

[5] E. Leitner, G. Rehm, and J. Moreno-Schneider, "Fine-grained named entity recognition in legal documents," in *Semantic Systems. The Power of AI and Knowledge Graphs (SEMANTiCS 2019)* (M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack, and Y. Sure-Vetter, eds.), pp. 272–287, Springer International Publishing, 2019.

[6] O. Shulayeva, A. Siddharthan, and A. Wyner, "Recognizing cited facts and principles in legal judgements," *Artificial Intelligence and Law*, vol. 25, pp. 107–126, 3 2017. Open access via Springer Compact Agreement.

[7] A. Sleimi, N. Sannier, M. Sabetzadeh, L. Briand, and J. Dann, "Automated extraction of semantic legal metadata using natural language processing," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pp. 124–135, IEEE, 2018.

[8] T.-S. Nguyen, L.-M. Nguyen, S. Tojo, K. Satoh, and A. Shimazu, "Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts," *Artificial Intelligence and Law*, vol. 26, pp. 169–199, June 2018.

[9] I. Chalkidis and D. Kampas, "Deep learning in law: Early adaptation and legal word embeddings trained on large corpora," *Artificial Intelligence and Law*, vol. 27, pp. 1–28, 12 2018.

[10] R. Nanda, A. K. John, L. D. Caro, G. Boella, and L. Robaldo, "Legal information retrieval using topic clustering and neural networks," in *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017)* (K. Satoh, M.-Y. Kim, Y. Kano, R. Goebel, and T. Oliveira, eds.), vol. 47 of *EPiC Series in Computing*, pp. 68–78, EasyChair, 2017.

[11] A. Morimoto, D. Kubo, M. Sato, H. Shindo, and Y. Matsumoto, "Legal question answering system using neural attention," in *4th Competition on Legal Information Extraction and Entailment (COLIEE 2017), held in conjunction with the 16th International Conference on Artificial Intelligence and Law (ICAIL 2017) in King's College London, UK* (K. Satoh, M. Kim, Y. Kano, R. Goebel, and T. Oliveira, eds.), vol. 47 of *EPiC Series in Computing*, pp. 79–89, EasyChair, 2017.

[12] P. Do, H. Nguyen, C. Tran, M. Nguyen, and M. Nguyen, "Legal question answering using ranking SVM and deep convolutional neural network," 2017.

[13] M.-Y. Kim, Y. Xu, and R. Goebel, "Applying a convolutional neural network to legal question answering," in *New Frontiers in Artificial Intelligence*

(M. Otake, S. Kurahashi, Y. Ota, K. Satoh, and D. Bekki, eds.), pp. 282–294, Springer International Publishing, 2017.

[14] D. M. Katz, M. J. B. II, and J. Blackman, "A general approach for predicting the behavior of the Supreme Court of the United States," *PLOS ONE*, vol. 12, pp. 1–18, 04 2017.

[15] N. Aletras, D. Tsarapatsanis, D. Preotiuc-Pietro, and V. Lampos, "Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective," *PeerJ Computer Science*, vol. 2, p. e93, 2016.

[16] O.-M. Şulea, M. Zampieri, M. Vela, and J. van Genabith, "Predicting the law area and decisions of French supreme court cases," in *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP 2017)*, (Varna, Bulgaria), pp. 716–722, INCOMA Ltd., Sept. 2017.

[17] S. Long, C. Tu, Z. Liu, and M. Sun, "Automatic judgment prediction via legal reading comprehension," in *Chinese Computational Linguistics (CCL)* (M. Sun, X. Huang, H. Ji, Z. Liu, and Y. Liu, eds.), (Cham), pp. 558–572, Springer International Publishing, 2019.

[18] M. B. L. Virtucio, J. A. Aborot, J. K. C. Abonita, R. S. Aviante, R. J. B. Copino, M. P. Neverida, V. O. Osiana, E. C. Peramo, J. G. Syjuco, and G. B. A. Tan, "Predicting decisions of the Philippine supreme court using natural language processing and machine learning," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, pp. 130–135, IEEE, 2018.

[19] K. Kowsrihawat, P. Vateekul, and P. Boonkwan, "Predicting judicial decisions of criminal cases from Thai Supreme Court using bi-directional GRU with attention mechanism," *2018 5th Asian Conference on Defense Technology (ACDT)*, pp. 50–55, 2018.

[20] T. Ansay and D. Wallace, *Introduction to Turkish law*. Kluwer Law International, 2005.

[21] E. Mumcuoğlu, C. E. Öztürk, H. M. Ozaktas, and A. Koç, "law-turk." https://github.com/koc-lab/law-turk, 2021.

[22] B. G. Buchanan and T. E. Headrick, "Some speculation about artificial intelligence and legal reasoning," *Stanford Law Review*, vol. 23, pp. 40–62, 05 1970.

[23] T. Bench-Capon, A. M. Araszkiewicz, A. K. Ashley, K. Atkinson, F. Bex, F. Borges, D. Bourcier, P. Bourgine, J. G. Conrad, E. Francesconi, T. F. Gordon, G. Governatori, J. L. Leidner, D. D. Lewis, R. P. Loui, L. T. McCarty, H. Prakken, F. Schilder, E. Schweighofer, P. Thompson, A. Tyrrell, B. Verheij, D. N. Walton, and A. Z. Wyner, "A history of AI and Law in 50 papers: 25 years of the international conference on AI and Law," *Artificial Intelligence and Law*, vol. 20, pp. 215–319, 2012.

[24] K. D. Ashley, "Toward a computational theory of arguing with precedents," in *Proceedings of the 2nd International Conference on Artificial Intelligence and Law (ICAIL 1989)*, (New York, NY, USA), p. 93102, Association for Computing Machinery, 1989.

[25] G. Sartor, "A simple computational model for nonmonotonic and adversarial legal reasoning," in *Proceedings of the 4th International Conference on Artificial Intelligence and Law (ICAIL 1993)*, (New York, NY, USA), p. 192201, Association for Computing Machinery, 1993.

[26] K. D. Ashley and E. L. Rissland, "A case-based approach to modeling legal expertise," *IEEE Expert: Intelligent Systems and Their Applications*, vol. 3, p. 7077, Sept. 1988.

[27] K. D. Ashley, *Modelling legal argument: Reasoning with cases and hypotheticals*. PhD thesis, USA, 1988. Order No: GAX88-13198.

[28] K. D. Ashley, "Reasoning with cases and hypotheticals in HYPO," *International Journal of Man-Machine Studies*, vol. 34, no. 6, pp. 753 – 796, 1991.

[29] C. D. Hafner and D. H. Berman, "The role of context in case-based legal reasoning: Teleological, temporal, and procedural," *Artificial Intelligence and Law*, vol. 10, no. 13, p. 1964, 2002.

[30] A. Wyner, "An ontology in OWL for legal case-based reasoning," *Artificial Intelligence and Law*, vol. 16, no. 361, 2008.

[31] K. D. Ashley, "Case-based reasoning and its implications for legal expert systems," *Artificial Intelligence and Law*, vol. 1, pp. 113–208, 1992.

[32] V. Aleven, "Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment," *Artificial Intelligence*, vol. 150, pp. 183–237, 11 2003.

[33] K. D. Ashley and S. Brüninghaus, "Automatically classifying case texts and predicting outcomes," *Artificial Intelligence and Law*, vol. 17, pp. 125–165, June 2009.

[34] C. Çetindağ, B. Yazıcıoğlu, and A. Koç, "Named entity recognition in Turkish legal texts," *Natural Language Engineering*, pp. 1–29, 2022.

[35] A. Elnaggar, R. Otto, and F. Matthes, "Deep learning for named-entity linking with transfer learning for legal documents," in *Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference (AICCC 2018)*, (New York, NY, USA), pp. 23–28, Association for Computing Machinery, 2018.

[36] E. K. Akkaya and B. Can, "Transfer learning for Turkish named entity recognition on noisy text," *Natural Language Engineering*, vol. 27, no. 1, pp. 35–64, 2020.

[37] A. Güneş and A. C. Tantuğ, "Turkish named entity recognition with deep learning," in *2018 26th IEEE Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2018.

[38] O. Güngör, T. Güngör, and S. Üsküdarlı, "The effect of morphology in named entity recognition with sequence tagging," *Natural Language Engineering*, vol. 25, pp. 147–169, 01 2019.

[39] G. Tür, D. Hakkani-Tür, and K. Oflazer, "A statistical information extraction system for Turkish," *Natural Language Engineering*, vol. 9, no. 2, pp. 181–210, doi: 10.1017/S135132490200284X, 2003.

[40] I. Chalkidis, I. Androutsopoulos, and A. Michos, "Obligation and prohibition extraction using hierarchical RNNs," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 254–259, Association for Computational Linguistics, July 2018.

[41] J. O'Neill, P. Buitelaar, C. Robin, and L. O'Brien, "Classifying sentential modality in legal language: A use case in financial regulations, acts and directives," in *Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law (ICAIL 2017)*, (New York, NY, USA), pp. 159–168, Association for Computing Machinery, 2017.

[42] D. Sangeetha, R. Kavyashri, S. Swetha, and S. Vignesh, "Information retrieval system for laws," in *2016 Eighth International Conference on Advanced Computing (ICoAC)*, pp. 212–217, IEEE, 2017.

[43] T. Ruger, P. Kim, A. Martin, and K. Quinn, "The Supreme Court forecasting project: Legal and political science approaches to predicting Supreme Court decisionmaking," *Columbia Law Review*, vol. 104, pp. 1150–1210, 05 2004.

[44] A. D. Martin, K. M. Quinn, T. W. Ruger, and P. T. Kim, "Competing approaches to predicting Supreme Court decision making," *Perspectives on Politics*, vol. 2, no. 4, pp. 761–767, 2004.

[45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.

[46] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of Artificial Intelligence Research*, vol. 37, p. 141188, Jan. 2010.

[47] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv:1301.3781*, 2013.

[48] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[49] I. Chalkidis and I. Androutsopoulos, "A deep learning approach to contract element extraction," in *Legal Knowledge and Information Systems - (JURIX 2017): The Thirtieth Annual Conference, Luxembourg, 13-15 December 2017* (A. Z. Wyner and G. Casini, eds.), vol. 302 of *Frontiers in Artificial Intelligence and Applications*, pp. 155–164, IOS Press, 2017.

[50] K. L. Branting, A. Yeh, B. Weiss, E. Merkhofer, and B. Brown, "Inducing predictive models for decision support in administrative adjudication," in *AI Approaches to the Complexity of Legal Systems* (U. Pagallo, M. Palmirani, P. Casanovas, G. Sartor, and S. Villata, eds.), pp. 465–477, Springer International Publishing, 2018.

[51] I. Chalkidis, I. Androutsopoulos, and A. Michos, "Extracting contract elements," in *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law (ICAIL 2017)*, (New York, NY, USA), p. 1928, Association for Computing Machinery, 2017.

[52] G. Tang, H. Guo, Z. Guo, and S. Xu, "Matching law cases and reference law provision with a neural attention model," in *IBM China Research, Beijing*, 2016.

[53] I. Nejadgholi, R. Bougueng, and S. Witherspoon, "A semi-supervised training method for semantic search of legal facts in Canadian immigration cases," in *Legal Knowledge and Information Systems - (JURIX 2017): The Thirtieth Annual Conference, Luxembourg, 13-15 December 2017* (A. Z. Wyner and G. Casini, eds.), vol. 302 of *Frontiers in Artificial Intelligence and Applications*, pp. 125–134, IOS Press, 2017.

[54] I. Chalkidis, A. Jana, D. Hartung, M. Bommarito, I. Androutsopoulos, D. Katz, and N. Aletras, "LexGLUE: A benchmark dataset for legal language understanding in english," *arXiv*, 2021.

[55] A. A. Akın and M. D. Akın, "Zemberek, an open source NLP framework for Turkic languages." https://github.com/ahmetaa/zemberek-nlp, 2013.

[56] E. Tursun, D. Ganguly, T. Osman, Y.-T. Yang, G. Abdukerim, J.-L. Zhou, and Q. Liu, "A semisupervised tag-transition-based Markovian model

for Uyghur morphology analysis," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 16, pp. 8:1–23, Nov. 2016.

[57] L. Vilnis and A. McCallum, "Word representations via Gaussian embedding," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[58] B. Athiwaratkun and A. Wilson, "Multimodal word distributions," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Vancouver, Canada), pp. 1645–1656, Association for Computational Linguistics, July 2017.

[59] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013) - Volume 2*, (Red Hook, NY, USA), pp. 3111–3119, Curran Associates Inc., 2013.

[60] A. Yüksel, B. Uğurlu, and A. Koç, "Semantic change detection with Gaussian word embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3349–3361, 2021.

[61] N. Sevim and A. Koç, "Investigation of gender bias in Turkish word embeddings," in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2021.

[62] L. K. Şenel, F. Şahinuç, V. Yücesoy, H. Schütze, T. Çukur, and A. Koç, "Learning interpretable word embeddings via bidirectional alignment of dimensions with semantic concepts," *Information Processing & Management*, vol. 59, no. 3, p. 102925, 2022.

[63] E. Ash, D. L. Chen, and A. Ornaghi, "Gender attitudes in the judiciary : Evidence from U.S. circuit courts," qapec discussion papers, Quantitative and Analytical Political Economy Research Centre, 2021.

[64] N. Sevim, F. Şahinuç, and A. Koç, "Gender bias in legal corpora and debiasing it," *Natural Language Engineering*, p. 134, 2022.

[65] L. K. Şenel, I. Utlu, V. Yücesoy, A. Koç, and T. Çukur, "Semantic structure and interpretability of word embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1769–1779, 2018.

[66] L. K. Şenel, V. Yücesoy, A. Koç, and T. Çukur, "Interpretability analysis for Turkish word embeddings," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2018.

[67] L. K. Şenel, I. Utlu, F. Şahinuç, H. M. Ozaktas, and A. Koç, "Imparting interpretability to word embeddings while preserving semantic structure," *Natural Language Engineering*, vol. 27, no. 6, p. 721746, 2021.

[68] A. Köksal, "Turkish pre-trained Word2vec model." https://github.com/akoksal/Turkish-Word2Vec, 2018.

[69] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)*. USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[70] L. Rokach and O. Maimon, *Decision Trees*, pp. 165–192. Boston, MA: Springer US, 2005.

[71] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[72] M.-A. Kaufhold, M. Bayer, and C. Reuter, "Rapid relevance classification of social media posts in disasters and emergencies: A system and evaluation featuring active, incremental and online learning," *Information Processing & Management*, vol. 57, no. 1, p. 102132, 2020.

[73] L. Follett, S. Geletta, and M. Laugerman, "Quantifying risk associated with clinical trial termination: A text mining approach," *Information Processing & Management*, vol. 56, no. 3, pp. 516–525, 2019.

[74] J. Haneczok and J. Piskorski, "Shallow and deep learning for event relatedness classification," *Information Processing & Management*, vol. 57, no. 6, p. 102371, 2020.

[75] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 01 2001.

[76] A. Kumar, K. Srinivasan, W.-H. Cheng, and A. Y. Zomaya, "Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data," *Information Processing & Management*, vol. 57, no. 1, p. 102141, 2020.

[77] C. Cortes and V. N. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 09 1995.

[78] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[79] A. Elnagar, R. Al-Debsi, and O. Einea, "Arabic text classification using deep learning models," *Information Processing & Management*, vol. 57, no. 1, p. 102121, 2020.

[80] D. Ji, P. Tao, H. Fei, and Y. Ren, "An end-to-end joint model for evidence information extraction from court record document," *Information Processing & Management*, vol. 57, no. 6, p. 102305, 2020.

[81] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1724–1734, Association for Computational Linguistics, Oct. 2014.

[82] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, (ICLR 2015), San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[83] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), (Berlin, Germany), pp. 207–212, Association for Computational Linguistics, Aug. 2016.