MARMARA UNIVERSITY

INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

Natural Language Processings in Legal Domain: Classification of Turkish Legal Texts

ONUR AKÇA

MASTER THESIS

Department of Computer Science and Engineering

Thesis Supervisor

Assoc. Prof. Dr. Murat Can Ganiz

ISTANBUL, 2023

MARMARA UNIVERSITY

INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

Natural Language Processings in Legal Domain: Classification of Turkish Legal Texts

ONUR AKÇA
(524120009)

MASTER THESIS

Department of Computer Science and Engineering

Thesis Supervisor

Assoc. Prof. Dr. Murat Can Ganiz

ISTANBUL, 2023

MARMARA UNIVERSITY

INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

Onur Akça, a Master of Science student of Marmara University Institute for Graduate Studies in Pure and Applied Sciences, defended his thesis entitled, "**Natural Language Processings in Legal Domain: Classification of Turkish Legal Texts**", on 15.08.2023 and has been found to be satisfactory by the jury members.

Jury Members

Assoc. Prof. Dr. Murat Can Ganiz (Advisor)
Marmara University, Department of Computer Engineering .....

Prof. Dr. Ali Fuat Alkaya (Jury Member)
Marmara University .....

Prof. Dr. Selim Akyokuş (Jury Member)
Medipol University .....

APPROVAL

Marmara University Institute for Graduate Studies in Pure and Applied Sciences Executive Committee approves that Onur Akça be granted the degree of Master of Science in Department of Computer Engineering, Computer Engineering Program on ..... (Resolution no: .....).

Director of the Institute
Prof. Dr. Bülent Ekici

ACKNOWLEDGEMENT

I would like to express my profound gratitude to my advisor, Assoc. Prof. Murat Can Ganiz, for his patience, support, and guidance throughout my thesis work. I am also deeply thankful for his mentorship as he led me into the domain of Natural Language Processing, an area that has witnessed significant growth throughout the course of my academic journey.

I'd like to express my gratitude to the entire team of the Big Data and Text Analytics (BIGDaTA) Research Laboratory for their significant inputs. A special mention to our domain expert Cihan Erdoğanyılmaz for his assistance with label merging and his valuable insight in AI studies on Turkish legal domain. My thanks also go out to Mehmet Selman Baysan and Fatih Satı for their work and help on collecting Turkish legal data from public resources for this study. Furthermore, Abdul Majeed Issifu and Giyaseddin Bayrak have been instrumental with their expertise in the realm of large language models, and I am grateful for their guidance.

I also owe a deep debt of gratitude to my beloved wife, Ödül, who has patiently supported me and helped maintain my motivation throughout this process.

I wish to thank everyone who contributed to the completion of this study. My sincere appreciation extends to all the organizations and individuals who have made my work come to life.

07.2023 **ONUR AKÇA**

TABLE OF CONTENTS

ABSTRACT

NATURAL LANGUAGE PROCESSINGS IN LEGAL DOMAIN: CLASSIFICATION OF TURKISH LEGAL TEXTS

Key words: Natural Language Processing, Large Language Models, Domain Adaptation, Text Classification, Legal Document Classification

Legal documents such as higher court decisions are complicated due to the intensive use of technical vocabulary. They are usually composed of very long and complex sentences. This is especially visible in Turkish legal documents due to the highly morphological and agglutinative nature of the language. Due to these difficulties and the lack of large benchmark datasets, there have been only a few Natural Language Processing (NLP) studies on artificial intelligence use in Turkish legal texts. In this research, we utilize a large unsupervised dataset of about 10 GBs of legal texts and compile a supervised dataset of about 90 thousand higher court decisions having unique 56 crime labels. Our main aim is the see how domain adaptation, i.e. continued pre-training of BERT, a large language model, by employing a domain-specific corpus affects the classification performance. We conduct extensive multi-class and multi-label classification experiments with a range of classifiers. As expected, BERT models outperform other classifiers by a wide margin. More importantly, we show that domain adaptation leads to about a 2% increase in F1 score. Our study contributes to the expanding corpus of studies on NLP in the legal domain and highlights the potential of domain-specific language models.

ÖZET

NATURAL LANGUAGE PROCESSINGS IN LEGAL DOMAIN: CLASSIFICATION OF TURKISH LEGAL TEXTS

Anahtar Kelimeler: Doğal Dil İşleme, Büyük Dil Modelleri, Alan Uyarlaması, Metin Sınıflandırma, Hukuki Belge Sınıflandırma

Yüksek mahkeme kararları gibi hukuki belgeler, teknik kelimelerin yoğun kullanımı nedeniyle karmaşıktır. Genellikle çok uzun ve karmaşık cümlelerden oluşurlar. Bu durum, dilin yüksek morfolojik ve eklemeli yapısı nedeniyle özellikle Türkçe hukuki belgelerde daha belirgindir. Bu zorluklar ve büyük kıyaslama veri kümelerinin eksikliği nedeniyle, Türkçe hukuk metinlerinde yapay zeka kullanımı üzerine sadece birkaç Doğal Dil İşleme (DDİ) çalışması yapılmıştır. Bu çalışmada, yaklaşık 10 GB'lık hukuk metinlerinden oluşan büyük bir denetimsiz veri kümesi kullandık ve benzersiz 56 suç etiketine sahip yaklaşık 90 bin yüksek mahkeme kararından oluşan bir denetimli veri kümesi derledik. Temel amacımız, etki alanı uyarlamasının, yani büyük bir dil modeli olan BERT'in etki alanına özgü bir derlem kullanılarak ince ayarının sınıflandırma performansını nasıl etkilediğini görmektir. Çeşitli sınıflandırıcılarla kapsamlı tek etiketli ve çok etiketli sınıflandırma deneyleri gerçekleştiriyoruz. Beklendiği gibi, BERT modelleri diğer sınıflandırıcılardan büyük bir farkla daha iyi performans gösteriyor. Daha da önemlisi, etki alanı uyarlamasının F1 puanında yaklaşık %2 artışa yol açtığını gösteriyoruz. Çalışmamız, hukuk alanında DDİ üzerine giderek artan araştırmalara katkıda bulunmakta ve alana özgü dil modellerinin potansiyelini vurgulamaktadır.

LIST OF ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
BERT	Bidirectional Encoder Representations From Transformers
Bi-LSTM	Bi-directional Long Short Term Memory
BoW	Bag-of-Words
CCC	Civil Court Corpus
CNN	Convolutional Neural Network
CoC	Court Of Cassation
GPT-3	Generative Pre-trained Transformer 3
GRU	Gated Recurrent Unit
LLM	Large Language Model
LR	Logistic Regression
LSTM	Long Short-Term Memory
MLM	Masked Language Model
MNB	Multinomial Naïve Bayes
NLP	Natural Language Processing
RBF	Radial Basis Function
RNN	Recurrent Neural Network
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TLC	Turkish Legal Corpus

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Natural Language Processing (NLP) represents a crucial subset of artificial intelligence which bridges human communication with machine understanding. The primary objective of NLP is to equip machines with the capacity to comprehend, analyze, and even replicate human language in a coherent manner. This profound ability holds the potential to revolutionize the way we interact with technology, redefining the boundaries of what is currently perceived as possible. In certain disciplines, such as the medical and legal domains, the potential applications of NLP are particularly pronounced. These fields are characterized by their requirement for specific domain knowledge, which can often act as a barrier for laypeople trying to understand complex documents or data. Every day, these domains generate an abundance of data in the form of court rulings, medical records, legal petitions, research articles, and more. This data, while rich in insights, can be difficult to navigate and understand due to its sheer volume and complexity.

This is where the power of NLP comes to the fore. NLP algorithms can effectively parse through these massive amounts of data, reducing the burden on human agents and enabling the extraction of relevant information with enhanced speed and precision. These algorithms can understand the context and semantic relationships within the language, allowing them to identify patterns and draw insights that could easily be overlooked by a human.

For instance, NLP algorithms can be employed to categorize large volumes of legal documents in a fraction of the time it would take a human to do the same task. By sorting these documents into relevant categories, stakeholders in the justice system, such as lawyers and judges, can access and utilize the information more efficiently. This enhanced accessibility to pertinent information allows for more informed and accurate decision-making processes, potentially improving the overall effectiveness of the justice system.

Furthermore, the potential applications of NLP extend beyond simply sorting and organizing data. They can provide real-time translation services, sentiment analysis, speech recognition, and much more. As our understanding and development of NLP continue to grow, so too does the potential for machines to interact with humans in increasingly nuanced and meaningful ways.

Over the past few years, the field of NLP has seen a surge in research, leading to

faster advancements in this particular area. Transformer-based language models like Bidirectional Encoder Representations from Transformers (BERT) [1] and Generative Pre-trained Transformer 3 (GPT-3) [2] changed the NLP domain. They also affect legal fieldwork by improving the accuracy and efficiency of legal document analysis and answer retrieval [3]. A language model uses machine learning algorithms or statistical methods to learn the language usage patterns from usually a large corpus of text to predict the subsequent word based on the given context. NLP models can now analyze contracts to identify key clauses, such as termination or indemnification clauses, or can be used to build chat-bots that can assist clients with legal questions or issues [4]. These practical applications recently became feasible due to the evolution of transformer-based Large Language Models (LLMs) and their incredible efficiency. LLMs are usually pre-trained using a very very large general text corpus from the internet and can be further trained using a much smaller dataset for a specific downstream NLP task or a specific technical domain. Continued pre-training can be used to adapt a language model to a specific task, a domain [5] or another language [6]. Continuing the pre-training of a general pre-trained LLM to a specific language domain such as legal domain or medical domain is called domain adaptation. Domain adaptation is a machine learning technique that seeks to improve the pre-trained model's performance on a target domain by continuing the pre-training of the model using domain-relevant data and the pre-training task. Domain adaptation allows researchers to create domain-specific language models such as LegalBERT [7] for the legal domain.

LLMs are available in many languages but the research and development are mainly focused on the English language for reasons such as availability of data and English being the primary language of many academic institutions and businesses. Languages with limited resources, such as Turkish, examine and evaluate the applicability of these studies in their language [5]. There are many pre-trained general Turkish LLMs available. There are also studies for adapting these LLMs for specific domains. Bayrak et al. [8], analyzed the domain adaptation of general Turkish LLMs for the medical domain in order to classify radiology reports. To the best of our knowledge, no such study on domain adaptation of general language models in the legal domain for Turkish texts exists. Only lately have been been a modest number of research in the legal domain for the classification of Turkish legal writings [9, 10].

Our research objectives are listed below:

- Investigate the challenges posed by extensive use of technical vocabulary and

highly complex sentence structures in legal documents, particularly in Turkish legal documents with their highly morphological and agglutinative nature in text classification.

- Compile a large labeled dataset to address the lack of large benchmark datasets in Turkish legal NLP research.
- Explore classification performance of traditional and deep learning-based classifiers on Turkish legal documents.
- Analyze the impact of domain adaptation by continued pre-training and specific task fine-tuning of BERT, on classification performance in the legal domain.
- Conduct extensive experiments in multi-class and multi-label settings to compare the performance of general domain and domain-adapted BERT models.

This research aims to investigate the efficiency of models specifically designed for the categorization of Turkish legal texts, with special emphasis laid on the effect of domain adaptation of LLMs, particularly the BERT, and their influence on classification performance. We delve deeper into the outcomes of domain adaptation and fine-tuning, by conducting a large number of experiments to help us draw conclusions. In conducting these tests, we utilize a compilation of publicly available decisions from higher-court instances. This collection is supplemented with an unlabeled dataset, comprising court decision texts and a variety of other legal texts sourced from various public resources. We anticipate that these datasets will serve as valuable benchmarks for upcoming studies in the field of natural language processing within the Turkish legal realm. Our experiments show that despite the complexity of legal texts, with their technical vocabulary, complex and lengthy sentence structures, and the intricacies of Turkish as a highly morphological and agglutinative language, crime labels can be predicted with reasonable accuracy, above 70% for both multi-class and multi-label classification settings. Furthermore, we establish that adapting a large language model, such as BERT, to the specific nuances of the Turkish legal domain employing an unsupervised domain-specific corpus enhances the F1 performance in terms of crime classification. This adapted model consistently outperforms all other classifiers experimented with within our study. This research contributes to the ever-expanding field of study on Natural Language Processing NLP within the legal domain, underscoring the potential and practicality of domain-specific language models. In addition, by making available extensive benchmark datasets and detailed results of numerous classifiers trialed on these datasets, we aim to aid in setting

a foundation for future research endeavors within this area.

The structure of this document is as follows; in Section 2, we delve into a comprehensive exploration of the most significant technologies that underpin our study, from the base algorithms to the transformer-based LLMs. We dissect the key elements and inner workings of these technologies, focusing on how they contribute to the domain adaptation and fine-tuning processes. We introduce the unique dataset used for our research. This data, collected from various public resources, including higher-court decisions and other legal texts, provides a rich resource for our experimentation. We detail our approach to the problem, outlining the methodologies we employed for domain adaptation, fine-tuning the language model, and classifying the legal texts. We also explain our experiment setup, including the configuration of our machine learning models and our evaluation metrics in section 3. Section 4 provides the findings of our careful evaluation. We present a detailed examination of the performance of different classifiers on our Turkish legal text dataset. We demonstrate the effect of domain adaptation on the performance of text classification, highlighting the improvements achieved through continued pre-training of the BERT model with domain specific data. We also engage in a rigorous discussion on the implications of our findings, drawing connections between our results and the field's existing collection of knowledge. Finally, section 5 reflects on our study, considering the significance of our findings in the broader context of NLP research. We examine the study's limitations and make recommendations for further research in the topic, particularly in the domain of Turkish legal text classification. We hope that our research, and the benchmark datasets we have provided, will inspire further research and innovation in the field of NLP, especially in domain adaptation and fine-tuning techniques for highly morphological and agglutinative languages such as Turkish.

2 RELATED WORK

Text classification is a cornerstone in the field of Natural Language Processing (NLP) [11]. It represents a process whereby textual data is assigned to specific classes or categories based on its content. In its simplest form, text classification tasks involve labeling an entire document, or specific segments within it such as sentences or paragraphs, according to the predefined classes. These classifications, while appearing straightforward, lay the foundation for an array of practical applications that we encounter in our daily digital lives and professional sectors.

One of the most well-known uses of text classification is spam detection. Email clients leverage text classification algorithms to distinguish between legitimate emails and unwanted spam. Similarly, sentiment analysis, another essential application of text classification, plays a critical role in understanding and gauging public opinion. By classifying text based on the sentiment expressed, companies can discern customer feedback, online reviews, or social media comments as positive, negative, or neutral. In the customer service sector, text classification aids in efficient query routing. When a customer submits a query or a complaint, text classification algorithms can analyze and categorize the request based on its content. Furthermore, text classification is crucial in managing and organizing large volumes of documents, especially in legal and medical domains. It can assist in categorizing legal documents or medical records into meaningful categories based on their content, making it significantly easier for professionals to locate and utilize this information.

In recent times, the application of text classification has extended to legal text categorization. Given the large volume of legal documents generated daily, NLP algorithms can categorize these documents based on their content, helping legal professionals to manage their caseloads more efficiently. By identifying the key themes and topics within each document, this technology enables more streamlined organization and retrieval of case files, facilitating more informed legal proceedings.

The most traditional approaches to text classification included rule-based systems, where hand-crafted rules were created to categorize the text. However, these methods often require extensive domain knowledge and are difficult to scale up. The text classification process has substantially changed with the introduction of machine learning techniques. Methods such as Naive Bayes, Support Vector Machine (SVM), and Random Forests have been widely used for this task.

In these methods, text documents are represented as feature vectors using techniques like Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), or Binary weighting. BoW turns a document into a vector, with each element representing the count or frequency of a single unique word in the document, regardless of word order whereas TF-IDF considers the frequency of a word in a given text (Term Frequency) as well as its rarity across all documents (Inverse Document Frequency). The TF-IDF value grows with the total number of times a word shows up in the text however is reduced by the number of instances of the word itself in the corpus as a whole, which helps to account for the fact that some words appear more frequently in general. A simplified version of BoW is the binary weighting. Instead of calculating the count of each term in a document, binary weighting employs a binary schema in which if a term appears in a document, its element in the vector is set to 1, otherwise it is set to 0. Although these approaches have proven to be effective, they fail to grasp the semantic context and relationships of words in a document. This limitation has been overcome by the emergence of neural network-based models, which have brought about a significant shift in how text classification tasks are approached.

With the advent of Deep Learning, techniques such as Word Embeddings (like Word2Vec and GloVe), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) units, and Convolutional Neural Networks (CNNs) have become popular in the field of text classification. These models have proven effective at capturing complex patterns, semantic relationships, and context in textual data.

These methods are used for text classification in the legal domain. Orosz et al. [12] conducted a study comparing human experts' versus machine learning algorithms' performance on legal classification challenges using SVM Classifiers to find a cost-effective alternative to human experts. Sulea et al. [13] explored the use of SVM ensembles to categorize legal texts, such as court decisions or contracts. The aforementioned methodologies serve not only as foundational techniques in this realm but also provide a benchmark for gauging the performance of more sophisticated models we are going to explore in further sections. For example, Soh et al. [14] conducted a comparison of several text classifiers on Singapore Supreme Court decisions, including SVM, word embeddings, and neural network models.

More recently, transformer-based models which we will further discuss in section 2.1, including the Bidirectional Encoder Representations from Transformers (BERT), Generative Pre-trained Transformer 3 (GPT-3), and their variants, have become the state-of-

the-art in many NLP tasks including text classification [15]. These models, based on the transformer architecture, leverage self-attention mechanisms to capture dependencies between words, irrespective of their positions in the sentence. While language models are much more capable than previous methods, studies show that in special cases similar results can be achieved using simpler methods. Chen et al. [16] compared the performance of random forests and deep learning models, including CNN, LSTM, and BERT, on automated legal text classification. They found that information extraction coupled with random forests can achieve similar scores to deep neural networks. Clavié et al. [17] compare BERT based models to LSTM based models. They show that LSTM based models can capture enough information to achieve excellent performance on short legal text classification tasks.

2.1 Transformers

Artificial Neural Networks (ANNs), based on the linked nature of the human brain, revolutionized the field of text classification. Specifically, RNNs and CNNs have shown considerable success.

RNNs are designed to capture sequential information in data, making them suitable for text classification where context and order of words are vital. However, RNNs face challenges in capturing long-distance dependencies in text, leading to the development of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, which incorporate memory units to remember and utilize contextual information from distant parts of the text.

On the other hand, CNNs, initially used primarily for image processing tasks, have been adapted for text classification. CNNs use convolutional layers with multiple filters to automatically learn and extract useful features from the text, thereby capturing local dependencies and semantic concepts effectively.

Although RNNs and CNNs were instrumental in advancing text classification, their sequential nature made it difficult to parallelize computation and thus led to significant computational inefficiency. This limitation paved the way for Transformers, a novel architecture that focuses on the self-attention mechanism, which allows the model to pay multiple levels of attention to distinct words in context.

The Transformer model, introduced in the seminal paper "Attention is All You Need" by Vaswani et al. [18] abandoned the sequential nature of RNNs and instead relied

only on the self-attention mechanism. Self-attention allowed transformers to capture dependencies between words, irrespective of their distance in the text and paved the way for more powerful and computationally efficient models.

The core of the Transformer model is the method of self-attention, which is additionally known as scaled dot-product attention. This mechanism computes the relevance of each word in the context of every other word in the sentence, enabling the model to capture dependencies regardless of their position or distance in the text.

Each word in a sentence is translated into a collection of three vectors known as the Query, Key, and Value vectors. These vectors are generated through separate learnable linear transformations of the original word embedding.

The attention score for a given word is computed by taking the dot product of its Query vector and the Key vectors of all other words in the text, followed by a softmax operation to obtain the weights (probabilities). These weights reflect how much attention should be paid to other words when encoding a given word. The final output is obtained by multiplying these weights with the Value vector of each word and summing the results.

Transformers led to a series of powerful models like Bidirectional Encoder Representations from Transformers (BERT) [1], Generative Pre-trained Transformer 3 (GPT-3) [19], T5 [20], and others, which are generally known as Large Language Models (LLMs) that are setting new standards in various NLP tasks, including text classification.

2.2 Large Language Models

LLMs are designed to generate human-like text and can capture nuances in language, context, and world knowledge to an impressive degree, owing to their size and the diverse datasets on which they are trained. Unlike smaller models, they possess an inherent capability to comprehend complex semantics and long-range dependencies in textual data, making them potentially well-suited for the intricate language found in legal texts. They have become the standard technique for sequential data where previously RNNs were common [21].

They can generate human-like language, perform translations, and even respond to complex questions [19]. This enables one of the significant advantages of LLMs is their capacity for 'zero-shot' or 'few-shot' learning. This means that they can often perform tasks like text classification without the need for explicit fine-tuning, simply by being

prompted appropriately. This can be particularly valuable in scenarios where labeled training data is scarce or costly to obtain, a common situation in the legal domain.

There are studies related to the use of pre-trained Turkish LLMs but the multilingual models that support Turkish are gaining popularity as well. BERTurk is a BERT model trained on Turkish corpus by the MDZ Digital Library team. Training corpus "has a size of 35GB and 44,04,976,662 tokens" [22]. The same team also pre-trained DistilBERT and ELECTRA models. There are also multilingual language models that support Turkish. Multilingual BERT or MBERT [1], XLM-RoBERTa [23], and Google's multilingual model mT5 [24] supports Turkish. These language models have demonstrated significant performance in a variety of Turkish NLP tasks [25–27].

In spite of the impressive capabilities of LLMs, their application to specific tasks like legal text classification is not straightforward. They demand significant computing power for training and inference, which can be a limitation. Also, their "black box" nature makes it difficult to understand and explain their decision-making process, a critical factor in the legal domain.

Despite these challenges, preliminary research has shown promise in applying LLMs to legal text classification and other legal NLP tasks. While the legal domain has its own challenges, the ability to fine-tune and domain-adapt language models allows them to be used effectively in the legal domain.

2.3 Domain Adaptation

Transfer learning in NLP is a technique that uses a pre-trained model which is trained on a large corpus as the starting point for new NLP tasks. To adapt to the new task or domain, the pre-trained model is refined with a smaller task or domain-specific dataset. Pre-training a large language model from scratch is usually costly due to the immense number of model parameters which is learned from a very large textual dataset. The exact cost can vary depending on factors such as the size of the training dataset, the number of GPUs used for training, and the duration of the training process. For example, training the GPT-2 model is estimated to cost around $1,6m to Open AI [28]. That is why transfer learning is an important concept to adapt the models to face new challenges. One of these challenges is to adapt a model to a new domain such as medical, legal, financial, or social media domains.

Domain adaptation refers to the method of modifying a model that has been trained

on one domain (source domain) to work effectively on a different, but related domain (target domain). In NLP, a "domain" generally refers to a body of text with certain shared characteristics. These characteristics could be related to the topic of the text (e.g., medical literature vs. news articles), the style of the text (e.g., formal writing vs. social media posts), or other factors. The need for domain adaptation arises because a model trained on one type of text may not perform well when applied to a different type of text. This is due to the potential variations in language use, style, context, and specific terminologies between the two domains. Domain adaptation might involve continued pre-training of a model in its original task [29] or employing domain-specific data to fine-tune a pre-trained model [30].

Continued pre-training of LLMs to the legal domain is a hot topic in recent years [7, 31–33]. Chalkidis et al. train the "LegalBERT" using public legal data from EU, UK, and USA sources. Legal documents are typically long and can extend the input size limit of BERT-based models. To respond to this problem, Beltagy et al. [34] propose a transformer model (Longformer) with an attention mechanism that scales linearly with token size instead of quadratically. The quality of the data utilized during the domain adaptation procedure is another significant aspect to consider. Limsopatham [35], discusses different representation techniques for effective usage of BERT in the legal domain and states "Pre-training on the documents that are similar to the target task would result in more effective performance". Mamakas et al. [36] propose to combine LegalBERT with Longformer to get a legal domain-specific BERT model that can handle longer texts.

The main approach to domain adaptation is continuing the pre-training. During this process, the pre-trained model is further trained on a smaller dataset that is specific to the desired domain. Alternatively, there are other suggested approaches that could potentially be more effective in addressing this matter. There are several studies on how to gain new knowledge about the domain while not losing the general language knowledge obtained from initial training. Studies range from adding adapter modules on networks layers [37] to training only certain parts of the network like freezing a number of initial layers and using the training the rest [38], training only bias terms [39] and training only batch normalization parameters [40].

2.3.1 Adapters

Adapters are a technique employed in transfer learning for transformer-based models such as BERT or GPT. Rather than training the entire pre-trained model, adapters only train a small additional component that is inserted into the model, making it much more parameter-efficient.

In more technical terms, an adapter is a small neural network that is added to the model after each transformer layer. It is commonly a feed-forward network with one hidden layer. During fine-tuning for a specific task, the original transformer model's parameters aren't changed; just the parameters of these adapters are.

This has several advantages:

- Efficiency: Adapters require significantly fewer parameters to be updated during fine-tuning, making it much faster and less resource-intensive than fine-tuning the complete model.
- Flexibility: The same pre-trained model can be used with different adapters for different tasks because the original transformer settings are not modified. This makes it easy to switch between tasks or even to perform multiple tasks simultaneously with a single model.
- Avoiding Catastrophic Forgetting: A model may "forget" the information it gained during pre-training when it is fine-tuned for a new task, a condition known as catastrophic forgetting. This danger is reduced by maintaining the original parameters unchanged and only upgrading the adapters.

A well-known implementation of this approach is the AdapterHub [41] framework which we'll also use for this study.

2.3.2 Frozen Layers

This method, mentioned in the study of Chalkidis et al. [42] builds upon the study of Rosenfeld and Tsotsos [38]. The frozen layers approach is often used in transfer learning, a machine learning technique where a pre-trained model is fine-tuned for a new task. When fine-tuning such a model on a specific task, it's common to freeze the lower layers of the model that capture more general features of the language, and only train the upper layers that are expected to adapt to the specific task. This is done under the assumption that the lower layers have already learned useful representations

of the language during pre-training, and last few layers would be enough to distinguish the domain text.

Alternatively, all the layers of the pre-trained model can be frozen, and a new task-specific layer (or layers) can be added on top. In this case, only the weights of the new layers are trained, while the rest of the model simply acts as a feature extractor. This is a common approach when the new task's dataset is relatively small, as it reduces the risk of overfitting.

In both cases, the goal is to leverage the knowledge that the model has already learned during pre-training while minimizing the resources required for fine-tuning and reducing the risk of overfitting on the new task.

3 APPROACH

This study involves detecting the crime labels in higher court decisions. We approach the problem of detecting the crimes involved in a court decision as a text classification problem. Court of Cassation (CoC), the higher court of Turkey, provides a large number of decision texts via its public search engine. This data is in a semi-structured format in which the crime labels for criminal court decisions are specifically indicated in a different field. In order to build a strong classification model we compile a large dataset from this corpus. Compiling a large supervised dataset allows us to test deep learning-based classifiers along with the traditional machine learning-based classifier that are using Bag-of-Words (BoW) representation. In addition to this supervised dataset, we compiled large unsupervised datasets for domain adaptation of deep learning models, more specifically transformer based Large Language Models (LLMs) such as BERT [1]. For instance, the civil court decisions of CoC does not include explicit crime labels therefore it is an unsupervised dataset from the same source. We also compile a larger unsupervised dataset from master's thesis and Ph.D. dissertations, journal papers, and lawyer blogs in the justice domain in Turkish. All these datasets are important as they are first in terms of their size and coverage in the Turkish legal domain. These Turkish legal datasets will have the potential to increase the quantity and quality of NLP studies in this domain.

Large language models are highly complex, artificial neural networks or more appropriately deep learning algorithms using transformer architectures. The complexity of these models is usually measured by their parameter sizes which can range between hundreds of millions to hundreds of billions. They are pre-trained using an immense amount of text, the largest ones are trained using almost all publicly available text on the internet in either a specific language or multiple languages. Their exceptional performance has transformed the field of Natural Language Processing (NLP) and captured public attention. These pre-trained models have the advantage of being able to be tailored to a certain domain or task practically transferring the general knowledge of the language and therefore eliminating the need for very large training sets.

This study's aim is to investigate a practical approach for adapting general domain pre-trained LLMs, more specifically the BERT [1], to the Turkish legal domain for legal text classification. We have chosen to utilize BERT as our preferred LLM over larger models with more parameters due to its suitability in terms of computation power. BERT

is widely employed in numerous NLP downstream tasks, including text classification and its computational requirements align better with the capabilities of our laboratory. There are Turkish pre-trained BERT models available on the internet and they are used in various NLP tasks in the Turkish language [43–45].

The primary approach taken in this research project involves conducting two key experiments: "Multi-class Classification for Crime Labels" and its successor, "Corpus Influence on Domain Adaptation". The first experiment provides a foundational understanding of the phenomenon under study, while the second experiment is an evolved version of its predecessor, developed to investigate deeper dimensions of the subject matter and to address any limitations or gaps identified in the initial experiment. In the following section 3.1, we explain the dataset that is used in both experiments and their differences. Then in section 3.2 we explain the experiment setups. The outcome of the first experiment has an influence on the design and execution of the second one, we mention these influences in the experiment setup section but the outcomes of both experiments are discussed in section 4.

3.1 Dataset

The application of Artificial Intelligence (AI) techniques, particularly machine learning-based models in the field of Natural Language Processing (NLP), is relatively recent in the context of processing Turkish legal documents, such as court decisions. Consequently, there are only a few studies available on this topic. One of the primary factors contributing to this scarcity is the absence of benchmark datasets specific to this domain. Hence, the Turkish legal domain can be regarded as an under-resourced language domain. For this reason, this study starts with assembling datasets including benchmark sets and legal corpora. The sources we use for this data are as follows:

- **Turkish Court of Cassation Decisions**: The Turkish Court of Cassation, also known as the Supreme Court of Appeals (Yargıtay in Turkish), is the highest court of appeals in Turkey. Its jurisdiction covers both criminal and civil cases. The court reviews decisions and judgments of lower courts such as criminal and civil courts, and it has the authority to uphold or overturn these rulings. In the Turkish judicial system, the Court of Cassation's decisions play an important role. When it reviews a case, the rulings it issues often set legal precedents that lower courts are expected to follow. This helps to maintain consistency and uniformity

in the interpretation and application of the law throughout the country. Decisions of the Turkish Court of Cassation are public data and available via their public search engine [1].

- **Turkish Legislation**: Turkish legislation refers to the body of laws and regulations that are enacted and enforced in the Republic of Turkey. It includes the Constitution of the Republic of Turkey, codes and laws enacted by the Grand National Assembly of Turkey, regulations issued by various government bodies, and other legal instruments. These legislations are officially published in The Official Gazette of the Republic of Turkey (Resmi Gazete).
- **Masters and Ph.D. Thesis**: Yüksek Öğretim Kurumu (YÖK) is the Council of Higher Education in Turkey. It is a government-affiliated institution responsible for overseeing and regulating higher education in Turkey. YÖK publishes all Masters and Doctorate thesis that is produced in Turkey on its website. From this website, we have gathered theses on the legal domain.
- **Turkish Constitutional Court Decisions**: Since 23.09.2012 Turkish Constitutional Court is opened to individual applications. This constitutional mechanism allows individuals to directly petition the Constitutional Court for the protection of their fundamental rights and freedoms, as guaranteed by the Turkish Constitution. The decisions on individual petitions are available on the website of the Constitutional Court.
- **Academic Papers & Journals**: Academic journals and papers in the legal domain in Turkey are vital sources of scholarly research and analysis within the country's legal landscape. These publications encompass a wide range of legal subjects, including, but not limited to, constitutional law, civil law, criminal law, administrative law, and international law. Turkish legal journals and papers typically adhere to rigorous academic standards and are authored by legal scholars, practitioners, and experts in the field. They often employ a combination of theoretical frameworks, empirical research, case studies, and legal analysis to provide deep insights and critical perspectives on various legal issues in Turkey. These publications are typically written in Turkish and are widely circulated among legal professionals, researchers, and academics in the country. They serve as a platform for legal discourse, debate, and the exchange of ideas, contributing

[1] https://karararama.yargitay.gov.tr

to the advancement of legal scholarship and practice in Turkey.

The study requires unsupervised Turkish legal corpus for the domain adaptation of the language models and labeled legal texts to classify using the trained model. By using the sources mentioned above we assembled 3 different unsupervised corpora. We discuss the details of our Turkish legal corpora for both experiments in sections 3.1.1.2 and 3.1.2.2. We also need supervised datasets to test the performances of various machine-learning methods and large language models (LLMs). We use the Court of Cassation decisions on criminal courts for this purpose. Sections 3.1.1.1 and 3.1.2.1 go into detail about the preparation process of our supervised datasets for both experiments.

3.1.1 Multi-class Classification for Crime Labels

3.1.1.1 Supervised Dataset

We use classification to compare the performances of our models. For this classification, we need a benchmark dataset. The creation of this dataset involved downloading court decisions from the first half of 2021 from the search engine of the Turkish Court of Cassation. The Court of Cassation is the highest appellate court in Turkey that reviews judgments from the nation's criminal and civil courts and makes the final rulings. The judgments made by the Court of Cassation are ideally taken as precedent by lower courts to ensure uniformity of application throughout the country.

While the court decisions from the criminal courts can include multiple crime labels, decisions from the civil courts don't include any crime labels, making it a multi-label dataset. To simplify our first classification experiment, we transformed this dataset into a multi-class dataset by only retaining the first crime label in cases where a document contained multiple labels. All civil court decisions are categorized as 'Suçsuz' meaning they do not include a crime. Another method we employed for this simplification is to use only the most frequent 9 labels. The most frequent 9 class each has at least 2000 instances. Instances with other labels are used in the unsupervised corpus which we give more details in its own section.

While most crime labels are self-explanatory, two may need clarification: violation of laws no. 5607 and 6136, which correspond to anti-smuggling legislation and the law governing the possession of firearms, respectively. 220,488 court decisions were downloaded, from which we selected those falling under the nine most frequent labels to ensure sufficient training data for each class.

Table 1 details the class distribution in our supervised dataset. In table 2 translations of the labels can be found. It is important to understand that this distribution might not accurately represent the actual crime distribution, as the data used was only from the first six months of 2021. The data were divided into training, evaluation, and testing subsets at a ratio of 80%, 10%, and 10% respectively, maintaining a stratified distribution of classes. An example text with label "tehdit" (threat) from this dataset can be seen in example 1.

Example 1. *(TR) "KARAR: Yerel Mahkemece verilen hüküm temyiz edilmekle, başvurunun süresi ve kararın niteliği ile suç tarihine göre dosya görüşüldü; Temyiz isteğinin reddi nedenleri bulunmadığından işin esasına geçildi. Vicdani kanının oluştuğu duruşma sürecini yansıtan tutanaklar, belgeler ve gerekçe içeriğine göre yapılan incelemede: Eyleme ve yükletilen suça yönelik katılan [DEIDENTIFIED] vekilinin temyiz nedenleri yerinde görülmediğinden tebliğnameye uygun olarak, TEMYİZ DAVASININ ESASTAN REDDİYLE HÜKMÜN ONANMASINA, 13/04/2021 tarihinde oy birliğiyle karar verildi."*

(EN) "DECISION: The judgment given by the Local Court was appealed, and the file was discussed according to the duration of the appeal and the nature of the decision and the date of the crime; Since there were no reasons for the rejection of the appeal request, the merits of the matter were passed. In the examination made according to the content of the minutes, documents and justification reflecting the hearing process in which the conscientious opinion was formed: Since the reasons for the appeal of the representative of the participant [DEIDENTIFIED] regarding the action and the imputed crime were not deemed appropriate, in accordance with the notification, it was unanimously decided on 13/04/2021 that THE APPEAL WAS REJECTED ON THE MERITS AND THE VERDICT WAS APPROVED."

3.1.1.2 Unsupervised Dataset

For the continued pre-training of the transformer models, a corpus of Turkish legal texts is essential. We compile this corpus starting with 94,117 court decisions from the Court of Cassation, which are excluded from the supervised dataset. In addition, we incorporate 14,207 documents from Turkish Legislation and 2,245 Doctoral Theses in the field of Law from Turkish scholars. The average length of the documents, particularly those of doctoral theses, is quite considerable, averaging 102,062 words per document.

Table 1: Distribution of classes within the supervised dataset comprising 9 labels.

Label	Training Set	Validation Set	Test Set
Suçsuz	67960	8495	8496
Hırsızlık	7894	987	987
Tehdit	5846	730	731
Hükümlü veya Tutuklunun Kaçması	5012	626	627
Kasten Yaralama	4914	615	614
Hakaret	3097	387	387
Nitelikli Hırsızlık	2606	326	326
5607 Sayılı Kanuna Aykırılık	1914	239	239
6136 Sayılı Kanuna Aykırılık	1853	232	231

These legal documents are merged into a single text file without any prior processing to create the unsupervised corpus. In this corpus, every sentence is placed on a new line. The corpus has a size of 2.6 GB and contains approximately 107 million tokens. We call this initial legal corpus the Alpha Legal Corpus (ALC).

3.1.2 Corpus Influence on Domain Adaptation

3.1.2.1 Supervised Dataset

In the second experiment, we expand the dataset we use for the classification in order to create a harder problem for the model. We use the same data that we obtain from the court decisions. Out of the 220,488 court decisions we have gathered, 126,139 are civil court decisions that are used as an unsupervised dataset as explained in section 3.1.2.2. The remaining 94,349 are from criminal courts and they include one or more crime labels associated with them. We analyzed the dataset and extracted basic statistics first. The average word count for the court decision texts is 164 with a standard deviation of 136 terms. The shortest court decision has only 24 terms and the longest has 9,714 terms. The length distribution of court decisions is shown in Fig. 3.1. In our study, we attempt to predict these crimes by using them as class labels. This data is inherently multi-label and multi-class since there can be multiple crimes involved in a case. The overall dataset is our multi-label dataset. We create a second dataset by eliminating the documents that have multiple labels to form a multi-class dataset because we want to test our models for both multi-class and multi-label classification tasks. Court decision no. 2021/3155 of the Sixth Criminal Chamber that has the labels of "Theft" and "Violation of Housing Immunity" and its English translation is shown in example 2.

Example 2. *(TR) "Mahalli mahkemece verilen hükümler temyiz edilmekle dosya incelenerek, gereği düşünüldü: Dosya ve duruşma tutanakları içeriğine, toplanıp karar yerinde incelenerek tartışılan hukuken geçerli ve elverişli kanıtlara, gerekçeye ve hakimin takdirine göre; suçların suça sürüklenen çocuk tarafından işlendiğini kabulde usul ve yasaya aykırılık bulunmadığı anlaşılmış, diğer temyiz itirazları da yerinde görülmemiştir. Ancak; 1- Suça sürüklenen çocuğun, olayın akabinde şüphe üzerine durdurulduğunda hırsızlık suçunu işlediğini kabul ederek suç yerini gösterip çaldıkları malzemeleri iade etmek suretiyle henüz müracaatı bulunmayan müştekiye teslimini sağlaması karşısında; hakkında TCK'nın 168/1. maddesinin uygulanması gerektiğinin düşünülmemesi, 2- İşyeri dokunulmazlığını ihlal suçunun birden çok kişiyle birlikte işlenmesi sebebiyle suça sürüklenen çocuk hakkında TCK'nın 119/1-c maddesiyle de uygulama yapılması gerektiğinin gözetilmemesi, Bozmayı gerektirmiş, suça sürüklenen çocuğun temyiz itirazı bu bakımdan yerinde görülmüş olduğundan, hükmün açıklanan nedenle tebliğnameye aykırı olarak BOZULMASINA, sonuç ceza bakımından 1412 sayılı CMUK'nın 326/son fıkrasının gözetilmesine, 24.02.2021 gününde oybirliğiyle karar verildi."*

(EN) "The judgments given by the local court were appealed, the file was examined and the necessity was considered: According to the contents of the file and the minutes of the hearing, the legally valid and favorable evidence collected and examined and discussed at the decision, the reasoning and the judge's discretion; it has been understood that there is no violation of the procedure and the law in accepting that the crimes were committed by the child dragged into the crime, and other appellate objections have not been deemed appropriate. However; 1- In the face of the fact that the child dragged into the crime, when he was stopped on suspicion after the incident, admitted that he committed the crime, showed the crime scene and returned the stolen materials to the complainant; it was not considered that Article 168/1 of the Turkish Penal Code should be applied to him. 2- Not considering that Article 119/1-c of the TPC should also be applied to the child dragged into crime due to the fact that the crime was committed with more than one person, required to be reversed, and since the appeal of the child dragged into crime was deemed appropriate in this respect, it was decided unanimously on 24.02.2021 that the verdict be DISMISSED for the reason explained, contrary to the notification, and that the 326/last paragraph of the Code of Criminal Procedure No. 1412 be observed in terms of the resulting penalty."

The first experiment showed us some preparatory work is required to utilize the court

decision data. There are three main reasons why such work is necessary. First of all, these court decisions are linked to real crimes and come from real court records, we see a wide variety of crimes and hence a large number of unique labels. Even worse, some of these labels are actually describing the same crime, but they are written in a different style, or using different words and synonyms or they can include spelling errors. For this reason, we identified these problems and merge labels by working with a domain expert. Our domain expert guided us during the standardization of the crime labels process. For example; "threat to collect debt", "threat with more than one person", and "threat to create fear and panic among the public" labels are all collected under the "threat" label. Together, we identified 56 unique classes with minimum support of 100. A total of 7,183 documents were eliminated from the dataset as their crime labels do not belong to any of the 56 classes and were deemed irrelevant to the research question at hand. These 56 classes are listed in table 2, along with their class distributions for multi-class and multi-label sets.

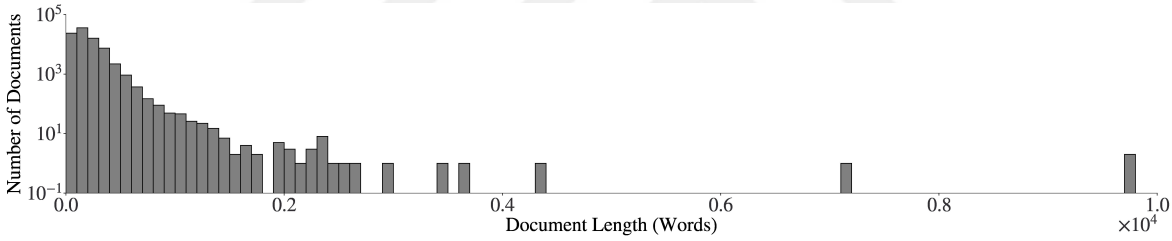

Figure 3.1: Number of terms in each document.

Fig. 3.2 illustrates the distribution of documents across different labels, thereby providing a visual summary of the frequency of occurrence of each label in the dataset. Please note that it is possible for individual documents to contain more than one label, resulting in the same document being counted under multiple labels. It is worth noting here again that our data consist of only the decisions published in the first six months of 2021 hence these labels and their distributions are not necessarily representatives of crime distribution of all CoC decisions. Fig. 3.3 displays the frequency distribution of the number of labels attributed to each document in the dataset.

Another preprocessing operation is done for the court decisions that include crime labels in the text. We identify and remove these labels from the text to avoid data leaks. Turkish is a highly morphological and agglutinative language. This means that some label roots might occur in the text with different suffixes. Label "hırsızlık" which means theft is a good example. Hırsızlık comes from the root word "hırsız" which means thief.

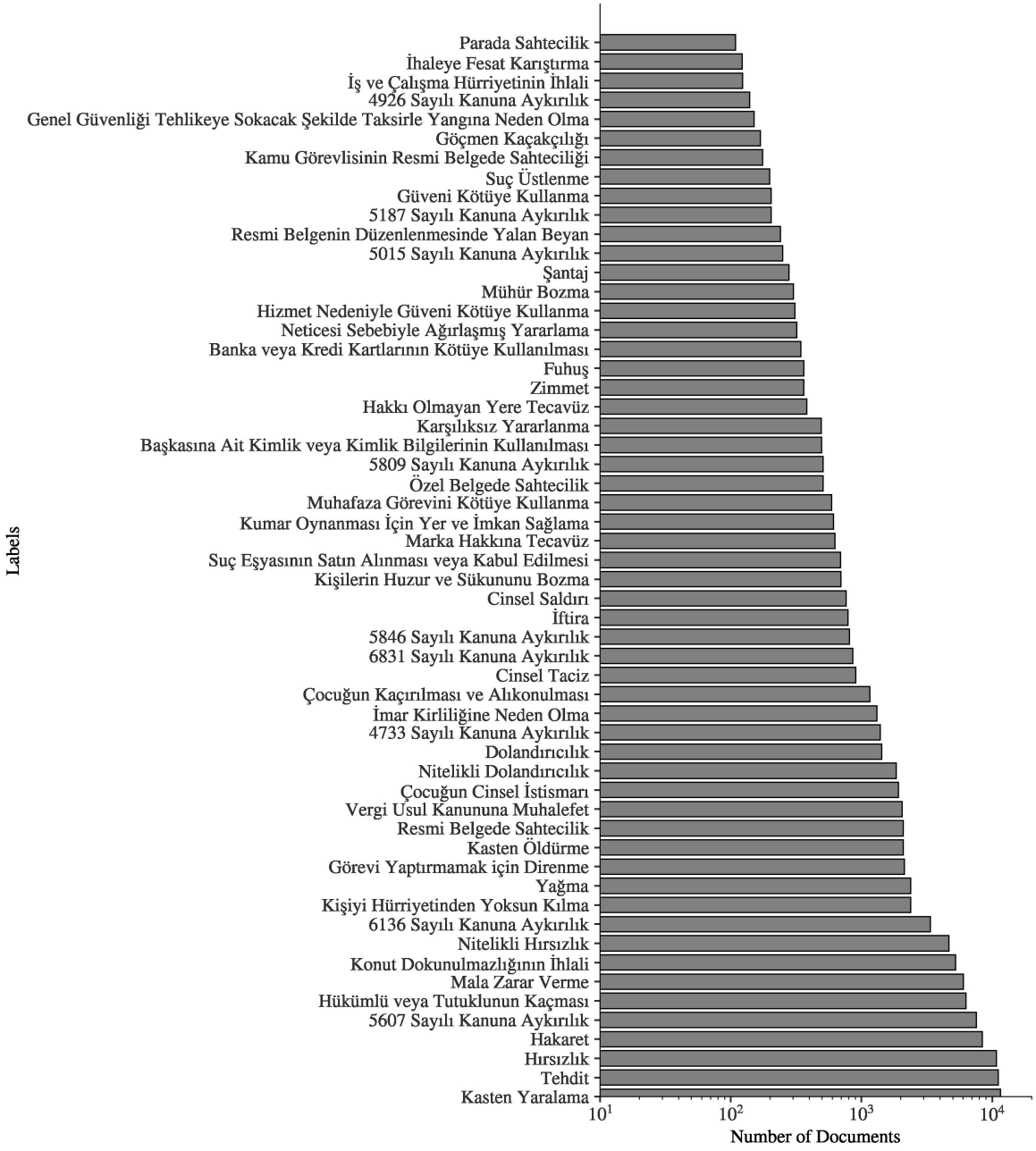

Figure 3.2: Distribution of labels in logarithmic scale.

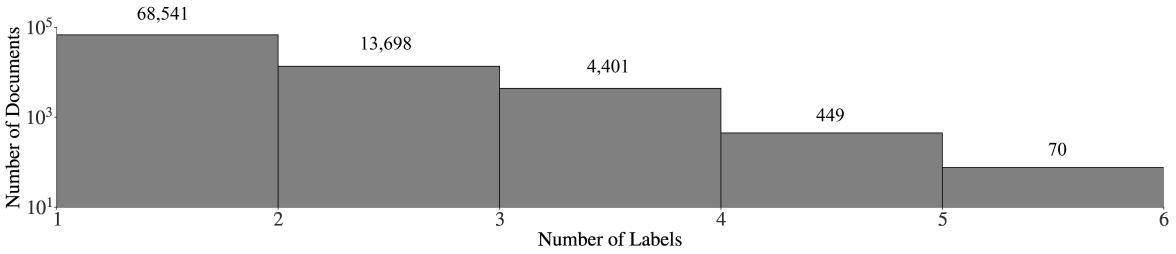

Figure 3.3: Number of labels in each document in logarithmic scale.

Table 2: Full list of labels and their distributions in multi-label and multi-class sets.

No	Label (Turkish)	Label (English)	perc. (Multi-label)	perc. (Multi-class)
1	Kasten Yaralama	Intended Injury	10.38%	10.21%
2	Tehdit	Threat	10.02%	7.78%
3	Hırsızlık	Theft	9.66%	9.55%
4	Hakaret	Insult	7.55%	3.82%
5	5607 Sayılı Kanuna Aykırılık	Violation of the Law No 5607	6.80%	10.80%
6	Hükümlü veya Tutuklunun Kaçması	Escape of the Detainee	5.65%	9.17%
7	Mala Zarar Verme	Damage to Property	5.42%	2.04%
8	Konut Dokunulmazlığının İhlali	Violation of Housing Immunity	4.71%	0.92%
9	Nitelikli Hırsızlık	Qualified Theft	4.20%	3.45%
10	6136 Sayılı Kanuna Aykırılık	Violation of the Law No 6136	3.03%	3.21%
11	Kişiyi Hürriyetinden Yoksun Kılma	Deprivation of Liberty	2.14%	1.47%
12	Yağma	Looting	2.14%	2.75%
13	Görevi Yaptırmamak için Direnme	Resisting to an Officer	1.91%	1.07%
14	Kasten Öldürme	Homicide	1.88%	2.46%
15	Resmi Belgede Sahtecilik	Forgery of Official Document	1.87%	1.40%
16	Vergi Usul Kanununa Muhalefet	Opposition to Tax Procedures	1.84%	2.98%
17	Çocuğun Cinsel İstismarı	Sexual Abuse of a Child	1.73%	1.84%
18	Nitelikli Dolandırıcılık	Qualified Fraud	1.66%	1.24%
19	Dolandırıcılık	Fraud	1.28%	1.71%
20	4733 Sayılı Kanuna Aykırılık	Violation of the Law No 4733	1.25%	1.96%
21	İmar Kirliliğine Neden Olma	Causing Land Use Pollution	1.18%	1.88%
22	Çocuğun Kaçırılması ve Alıkonulması	Abduction and Detention of a Child	1.04%	1.60%
23	Cinsel Taciz	Sexual Harassment	0.81%	0.72%
24	6831 Sayılı Kanuna Aykırılık	Violation of the Law No 6831	0.77%	1.25%
25	5846 Sayılı Kanuna Aykırılık	Violation of the Law No 5846	0.73%	1.18%
26	İftira	Slander	0.71%	0.87%
27	Cinsel Saldırı	Sexual Assault	0.68%	0.64%
28	Kişilerin Huzur ve Sükununu Bozma	Disturbing the Peace and Tranquility of People	0.63%	0.57%
29	Suç Eşyasının Satın Alınması veya Kabul Edilmesi	Purchasing or Accepting Criminal Goods	0.62%	0.78%
30	Marka Hakkına Tecavüz	Trademark Infringement	0.57%	0.92%
31	Kumar Oynanması İçin Yer ve İmkan Sağlama	Providing a Place and Opportunity for Gambling	0.55%	0.86%
32	Muhafaza Görevini Kötüye Kullanma	Misuse of Duty to Protect	0.53%	0.85%
33	Özel Belgede Sahtecilik	Forgery of Document	0.46%	0.36%
34	5809 Sayılı Kanuna Aykırılık	Violation of the Law No 5809	0.46%	0.66%
35	Başkasına Ait Kimlik veya Kimlik Bilgilerinin Kullanılması	Identity Theft	0.44%	0.58%
36	Karşılıksız Yararlanma	Benefiting From A Service Without Compensation	0.44%	0.66%
37	Hakkı Olmayan Yere Tecavüz	Trespassing	0.34%	0.50%
38	Zimmet	Embezzlement	0.33%	0.49%
39	Fuhuş	Prostitution	0.33%	0.49%
40	Banka veya Kredi Kartlarının Kötüye Kullanılması	Misuse of Bank or Credit Cards	0.31%	0.34%
41	Neticesi Sebebiyle Ağırlaşmış Yaralama	Consequential Injury	0.29%	0.38%
42	Hizmet Nedeniyle Güveni Kötüye Kullanma	Abuse of Trust Due to the Service	0.28%	0.39%
43	Mühür Bozma	Unsealing Offence	0.27%	0.32%
44	Şantaj	Blackmail	0.25%	0.19%
45	5015 Sayılı Kanuna Aykırılık	Violation of the Law No 5015	0.22%	0.36%
46	Resmi Belgenin Düzenlenmesinde Yalan Beyan	False Declaration	0.22%	0.28%
47	5187 Sayılı Kanuna Aykırılık	Violation of the Law No 5187	0.18%	0.30%
48	Güveni Kötüye Kullanma	Abuse of Trust	0.18%	0.23%
49	Suç Üstlenme	Taking the Blame	0.18%	0.26%
50	Kamu Görevlisinin Resmi Belgede Sahteciliği	Official Document Fraud by a Public Officer	0.16%	0.20%
51	Göçmen Kaçakçılığı	Human Trafficking	0.15%	0.24%
52	Genel Güvenliği Tehlikeye Sokacak Şekilde Taksirle Yangına Neden Olma	Causing a Fire by Negligence	0.14%	0.21%
53	4926 Sayılı Kanuna Aykırılık	Violation of the Law No 4926	0.12%	0.20%
54	İş ve Çalışma Hürriyetinin İhlali	Violation of Freedom of Work and Labour	0.11%	0.13%
55	İhaleye Fesat Karıştırma	Bid Rigging	0.11%	0.15%
56	Parada Sahtecilik	Counterfeiting	0.10%	0.15%

This word might occur in the body as "hırsızlığı" which means "theft of" something. As a result, to identify if a crime label exists in the text, we use a Turkish stemmer to find the root word of the label and search for the root word in the documents to increase the coverage.

Additionally, there are very similar or even identical documents in the dataset. The main reason for this is the use of ready-made drafts especially for certain kinds of decisions such as the dismissal of the case on procedural grounds or a decision of lack of jurisdiction by the court. Since these templates do not include an investigation of the case, they don't possess any valuable information. They are identical or very similar to each other. Example 3 displays decision no. 2021/7891 of the Fourth Criminal Chamber, an instance of a decision that has been eliminated. This is due to the fact that there exist multiple court decisions which bear resemblance to this decision, making it redundant and unnecessary to keep. Albeit they may have crime labels coming from the court of first instance. They are usually the shortest decisions in our collection. These types of decisions can cause a problem when we split the data into training and test sets as the same or highly similar document the in test set can occur in the training set as well. To prevent this, after the training set, and test set split, we check the training set for the same or "very similar" occurrences in the test set. In order to detect these instances, we use fastText [46] embeddings to represent documents and calculate cosine similarity. 300-dimensional vectors from fastText Turkish common crawl embeddings are used. FastText is a word embedding library created by Facebook AI Research Lab. If a training set document has a cosine similarity of 0.995 and above to a test set document, the training set document is deleted hence reducing the size of the training set instead of the test set.

Example 3. *(TR) "KARAR Yerel Mahkemece verilen hükümler temyiz edilmekle, başvurunun süresi ve kararların niteliği ile suç tarihine göre dosya görüşüldü: Temyizin kapsamına göre sanığın tehdit suçundan beraatine ilişkin karara yönelik temyizle ilgili Yargıtay Cumhuriyet Başsavcılığının 01/10/2019 tarih ve 2015/341254 sayılı tebliğnamesinde görüş bulunmadığı, Anlaşıldığından, müşteki [DEIDENTIFIED]'nın temyiz davası isteği hakkında şimdilik bir KARAR VERİLMESİNE YER OLMADIĞINA ve dosyanın incelenmeksizin tehdit suçlarından sanığın mahkumiyetine ilişkin temyiz ile ilgili EK TEBLİĞNAME düzenlenmesi için Yargıtay Cumhuriyet Başsavcılığına GERİ GÖNDERİLMESİNE, 08/03/2021 tarihinde oy birliğiyle karar verildi."*

(EN) "DECISION The file has been discussed according to the duration of the appeal, the nature of the decisions and the date of the crime: According to the scope of the appeal, it is understood that there is no opinion in the notification of the Chief Public Prosecutor's Office of the Court of Cassation dated 01/10/2019 and numbered 2015/341254 regarding the appeal regarding the decision regarding the acquittal of the defendant from the crime, It was unanimously decided on 08/03/2021 that there is no place to make a decision about the request of the complainant [DEIDENTIFIED] for the appeal case for the time being and that the file be SENT BACK to the Chief Public Prosecutor's Office of the Court of Cassation for the issuance of an ADDITIONAL NOTICE regarding the appeal regarding the conviction of the defendant for the crimes without being examined."

3.1.2.2 Unsupervised Dataset

There are studies that show the importance of corpus selection for domain adaptation of models [47]. In this experiment, we want to study the effect of corpus on the performance of the model. Our first corpus included text from both CoC decisions and general legal texts. In this study, we've established two distinct corpora to examine the influence of the domain adaptation corpus on the efficiency of the classification model. One corpus consists of the court rulings which are closer to the target task data, while the other is an enhanced iteration of the generic corpus from the Turkish legal field (ALC). Both of these corpora consist of text files in this domain but with different specialties.

The first corpus called Turkish Legal Corpus (TLC) and contains about 9.68 GB of text from public resources including 9,943 documents of Turkish Legislation, 1,273 doctoral dissertations, and 12,928 Masters Thesis' from Turkish Universities in the Law domain, 9,407 decisions from the Turkish Constitutional Court, and over 2,000 academic publications from the journals of law faculties of universities and bar associations. Especially the doctoral dissertations and master's thesis are the longest documents, with journal papers coming second. Insights on the data are listed in table 3. Note that in order to represent the data better we have removed the outliers from this representation by taking into consideration only values between quantiles 0.005 and 0.995. This is the extension of the corpus that we used in the first experiment. The main difference between the two is that TLC is almost four times larger and it does not contain any text from CoC decisions.

The data we have gathered from the Court of Cassation (CoC) includes civil court decisions along with criminal courts. Unlike the first experiment, civil court decisions

Table 3: Dataset insights; Document count & term lengths of documents.

		Number of terms in documents					
Dataset Name	**Count**	**Min.**	**Max.**	**Mean**	**Median**	**Std. Dev.**	**Included in**
Masters Thesis	12,928	22,962	346,218	87,198	69,169	52,838	TLC
PhD Dissertations	1,273	17,453	330,691	117,295	107,554	57,453	TLC
Legislation	9,943	21	11,323	2,154	1,346	2,276	TLC
Constitutional Court Decisions	10,352	608	55,515	4,338	3,223	4,255	TLC
Academic Publications	2,435	2,021	406,310	73,842	23,920	81,101	TLC
Court of Cassation Civil Court Decisions	124,877	46	2596	287	166	294	CCC

are not used for classification tasks instead will be used to domain adapt the model. Therefore, these 126,139 civil court decisions, about 334 MB in size, are used as an unsupervised dataset for domain adaptation and will be referred to as Civil Court Corpus (CCC). While this corpus is much smaller than the TLC, it is much more similar to the criminal court decisions which we use in the last task of classification. The use of language, sentence formations, and technical jargon is very close to the criminal court decisions as they came from the same source; CoC which is the highest court of appeal for civil and criminal cases in Turkey similar to France and Italy.

3.2 Experiment Setup

3.2.1 Multi-class Classification for Crime Labels

The task of predicting the crime label in a court decision is formulated as a multi-class classification problem in this experiment. Since this research is among the first in the field of Turkish legal document classification, it aims to establish a benchmark. Both traditional machine learning algorithms and deep learning structures that utilize static and contextual word embeddings are employed throughout the experiments. Algorithms such as Multinomial Naïve Bayes (MNB), Logistic Regression (LR), and Support Vector Machine (SVM) with Radial Basis Function (RBF) Kernel are used, applying Bag-of-Words (BoW) representations with binary and Term Frequency-Inverse Document Frequency (TF-IDF) term weighting schemes [48].

For deep learning techniques, Bi-directional Long Short Term Memory (Bi-LSTM) classifier is used with fastText word representation method [49]. Word embeddings, considered vector representations of words or tokens, can capture semantics by illustrating the relationships between different words. Static word embeddings such as fastText are useful, yet they have limitations as they fail to distinguish different meanings of the same word. FastText, developed by the Facebook AI Research lab, is a static word embedding

method that maps each word into a multi-dimensional space with a relatively short and dense vector. We employ Turkish fastText vectors in the embedding layer in our network, which feeds into a single Bi-LSTM layer [50] followed by a feed-forward classification layer.

Transformers, introduced in 2017 [18], are deep learning architectures designed to handle sequential data like text and can be further trained for domain-specific corpora for better performance. This strategy has been proven effective in English Legal Document classification [7]. We utilize transformer-based models, specifically BERT and its distilled version DistilBERT, which offers faster processing with minimal sacrifice to language learning capacity [51].

Distillation in the context of NLP refers to the process of training a smaller, simpler model (often called the student model) to mimic the behavior of a larger, more complex model (known as the teacher model). This is done by having the student model learn from the outputs or the predictions of the teacher model rather than directly from the raw data. The goal of distillation is to create a model that maintains the performance of the larger model while being more computationally efficient and faster to use, especially in situations where resources are limited, such as on mobile devices or in real-time applications. This method allows for the preservation of essential knowledge from the complex model in a more accessible form, hence the term 'distillation'.

Pre-trained Turkish BERT (dbmdz/bert-base-turkish-cased) and DistilBERT (dbmdz/distilbert-base-turkish-cased) models from the MDZ Digital Library team are used, which are then domain adapted using our corpus of Turkish legal texts via a Masked Language Model (MLM) task. By domain adaptation we mean the continued pre-training of the model in one of its original tasks namely the Masked Language Model (MLM). The other pre-training task, next sentence prediction (NSP) is not used in this study because studies shows that NSP is either inconsistent or ineffective in continued pre-training [52–54]. For domain adaptation, we follow the original BERT paper's guidelines [1] and continue the pre-training for 3 epochs with a learning rate of 5e-5 and batch size of 32. We call our domain adapted BERT models as HukukBERT. Since we train more than one domain adapted models with several corpuses HukukBERT is a family of language models instead of a single model. We indicate the corpus used in the domain adaptation in the name of the model. For this experiment we call our model HukukBERT-ALC as it is trained on ALC corpus.

BERT can be utilized effectively in classification tasks by adding a dense classification layer atop the pre-trained BERT model. The overall process is highly efficient, as it leverages BERT's capability to extract complex language features. The sequences are fed into the BERT model, which generates meaningful contextual representations for each token in the input. Then, these representations are passed to the added classification layer. This layer, often a simple linear layer, makes the final prediction for the classification task. The whole model is then fine-tuned on a specific task using labeled data. The combination of the sophisticated feature extraction capabilities of BERT with a task-specific fine-tuning process makes BERT a powerful tool for text classification. Unlike in fastText + Bi-LSTM where the embedding layer is frozen, the transformer layers are not frozen during fine-tuning for classification. Grid Search technique is used to find optimal hyperparameters, exploring a space of epoch counts and learning rates.

As the evaluation metrics, we report accuracy, precision, recall, and F1 score which is the harmonic mean of precision and recall. Among these, precision, recall, and F1 are calculated for each class.

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

To obtain a single value from these for the classification model, we use simple averaging of all class values which is known as macro averaging. The difference between accuracy and macro average F1 is an important indicator of the performance of the classifier on small classes in datasets with skewed-class distribution as ours. In a skewed dataset, accuracy may not be a reliable performance metric because a model that simply predicts the majority class for all instances may achieve high accuracy but would not be very useful in practice. The macro F1 score takes into account the F1 scores of the small classes equally therefore it is the more informative measure. This is why we choose to use macro averaging.

3.2.2 Corpus Influence on Domain Adaptation

The initial experiment confirms the practicality of domain adaptation. In this subsequent study, we are intensifying our examination of a variety of adaptation strategies. The corpus employed in the first experiment encompassed both general legal prose and text derived from CoC. However, in this investigation, we plan to differentiate the CoC

content from the broader Turkish legal texts, in order to evaluate the impact of the corpus composition on the domain adaptation process.

Like the first experiment, we start with traditional machine learning classifiers to establish our baseline. These are MNB, SVM with RBF Kernel, and good old LR. We use sklearn [55] python library for the implementations. These algorithms are not directly applicable to multi-label classification. In order to adapt them to multi-label multi-class classification, we utilize the MultiOutputClassifier wrapper which uses the One-vs-Rest approach provided by the sklearn library.

For traditional machine learning classifiers, we use the traditional BoW approach and create a document by term matrices. Feature extraction is a useful technique for text analysis tasks such as sentiment analysis [56], and text classification [57]. However, due to the potential loss of context and the inability to capture semantic relationships between words we do not use this approach. Instead, we remove the Turkish stop words from the nltk library and Turkish legal stop words. A Turkish legal stop words list is a collection of commonly used words in legal documents that are deemed insignificant for analysis or search purposes. This list was compiled by calculating the frequency of each word in various court decisions and then removing the words that appeared too frequently to be useful in distinguishing between documents. By removing these words from the analysis, the focus can be placed on more meaningful terms and concepts within the legal documents. The list includes words like; oybirliği (consensus), sonra (later), and gün (day). After the stop word removal, the remaining words then stemmed. For stemming we use the Fixed Prefix Stemming approach which only uses a word's first n characters [48, 58], for this study n is taken 5. These yield a vocabulary size of about 18 thousand terms and 21 thousand terms for multi-class and multi-label datasets, respectively. Finally, we use binary term weighting as our previous experiment shows that it performs better or comparable to term frequency or TF-IDF weighting. These pre-processing methods are only applied to traditional machine learning methods.

We use the same pre-trained fastText [46] embeddings trained on Turkish Common Crawl corpus for 300-dimensional word representations in deep learning based classifiers. FastText library also includes a classifier. It uses a Logistic Regression (LR) layer on the top of the fastText embeddings [49]. To examine the effects of utilizing Bi-LSTM, in this experiment, we additionally share the results for this included classifier. We train this using our supervised datasets for 25 epochs with a learning rate of 0.1. For multi-class classification, we use softmax as the activation function. For multi-label

Table 4: Grid search results for Bi-LSTM: Sigmoid activation function.

Activation Function	Loss Function	Batch	Epoch	F1 Score
Sigmoid	Binary Cross Entropy	128	16	35.16
			32	49.35
			64	66.21
			128	66.64
		256	16	30.66
			32	41.32
			64	55.73
			128	67.05
		512	16	17.19
			32	31.82
			64	36.60
			128	62.21
	Categorical Cross Entropy	128	16	46.30
			32	56.84
			64	64.64
			128	66.88
		256	16	40.37
			32	56.24
			64	63.61
			128	63.92
		512	16	38.04
			32	50.05
			64	58.69
			128	64.58

classification, One-vs-Rest approach is used.

We also use fastText embeddings to initialize Bi-directional Long Short Term Memory (Bi-LSTM) [50]. In training Bi-LSTM models, fastText embeddings are frozen and only Bi-LSTM layers are modified. We find best performing hyperparameters using grid search. These are 128 batch size, 128 epochs, softmax activation, and binary cross-entropy loss function. This search is done using the multi-class set. Our search space is activations = ['sigmoid', 'softmax', 'relu', 'tanh'], loss functions = ['binary crossentropy', 'categorical crossentropy'], batches = [128, 256, 512] and epochs = [16, 32, 64, 128]. Results of this search are given in tables 4 and 5.

As the LLM, we use only BERT as it performed better than the DistilBERT in the first experiment. A general domain Turkish BERT model pre-trained on OSCAR corpus

Table 5: Grid search results for Bi-LSTM: Softmax activation function.

Activation Function	Loss Function	Batch	Epoch	F1 Score
Softmax	Binary Cross Entropy	128	16	30.60
			32	53.91
			64	64.25
			128	67.89
		256	16	26.93
			32	40.19
			64	40.67
			128	65.12
		512	16	18.00
			32	30.16
			64	43.35
			128	61.00
	Categorical Cross Entropy	128	16	44.83
			32	59.64
			64	64.18
			128	65.82
		256	16	43.35
			32	57.39
			64	60.85
			128	64.08
		512	16	31.01
			32	39.52
			64	57.79
			128	63.78

[59] and OPUS corpus [60] is available thanks to the efforts of the MDZ Digital Library team from the Bavarian State Library [22]. This model is available in HuggingFace [61] under the name 'dbmdz/bert-base-turkish-cased'. Here, we adapt the pre-trained BERT model to fit our task and domain

We want to note the important distinction between continued pre-training for domain adaptation and fine-tuning for task adaptation in our study. Domain adaptation [62] is a type of machine learning technique where a pre-trained LLM using a general domain large corpus in a self-supervised manner is adapted to work on a specific domain by continuing the pre-training on relatively small unsupervised data from the target domain. To put it simply, the purpose is to transfer knowledge from the source domain to the target domain. In our study, we continue the self-supervised training of BERT on one of its original training objective of the Masked Language Model (MLM). A domain-adapted model is shown to perform better at downstream tasks in the target domain [7]. On the other hand, the fine-tuning of the model for a specific downstream task using a supervised dataset is referred to as task adaptation. In our study, this is the classification task. We create our Turkish legal domain models by domain adaptation of the Turkish BERT model. Following this, we fine-tune each model for classification using a labeled dataset. These Turkish models that are domain adapted for the legal domain will potentially boost NLP studies in this domain.

For the domain adaptation, we use two unlabeled datasets whose details are given in section 3.1.2.2. We continue the pre-training of the BERT model with these unsupervised datasets. We use a batch size of 32 and fine-tune for 3 epochs with a learning rate of 5e-5 as it was suggested in the original paper [1]. These models are named HukukBERT-TLC and HukukBERT-CCC according to the corpus used in domain adaptation. We also explore the freezing of some of the layers [38, 42]. BERT architecture has 12 layers. We experiment with training only a quarter (last 3 layers) and a half (last 6 layers) of the model and freezing all the other layers. These models are named HukukBERT-FF6-CCC and HukukBERT-FF9-CCC. FF stands for "freeze first" and the number after that indicated the number of initial layers that are frozen.

Another adaptation method we try is the adapter mechanism. An adapter [37] is a type of transfer learning technique that allows for the efficient adaptation of pre-trained models to new downstream tasks. Adapters are small and task-specific neural network modules that can be plugged into pre-trained transformer models. This allows for faster and more efficient training on new tasks as the pre-trained model can be used

Table 6: Details of domain adapted models. FF stands for "freeze first" and the number that follows indicates how many layers are frozen.

Model Name	Corpus	Number of Frozen Layers	Epoch	Batch	Learning Rate
HukukBERT-ALC	Alpha Legal Corpus	0	3	32	5e-5
HukukBERT-TLC	Turkish Legal Corpus	0	3	32	5e-5
HukukBERT-CCC	Civil Court Corpus	0	3	32	5e-5
HukukBERT-FF6-CCC	Civil Court Corpus	6	3	32	5e-5
HukukBERT-FF9-CCC	Civil Court Corpus	9	3	32	5e-5

as a fixed feature extractor while only the task-specific adapter is trained. We use the implementation of adapters which is made available by AdapterHub [41]. To explore the possibility of using adapters for fine-tuning for task adaptation; we use our domain-adapted models and freeze them and train only the adapters for the classification task. We use the adapter architecture proposed by Pfeiffer et al. [63]. We explore both fine-tuning whole model (indicated with abbrevation FT in the model name) and fine-tuning only the adapter layers (indicated with abbrevation AD in the model name) to compare this two approaches.

Lastly, we want to see if the additional parameters gained by using adapters would change the performance of an already fine-tuned model. We use the HukukBERT-CCC, HukukBERT-FF9-CCC, and HukukBERT-FF6-CCC after fine-tuning the models add adapters and train them as well to see if the adapters would increase the performance of an already task adapted model. On adapter training, we employ the same hyperparameter set of learning rate 5.00E-05, batch size 8, and epoch 10. We observed the effect of domain adaptation on classification tasks by comparing BERT classifiers based on the domain-adapted models with different unsupervised datasets and frozen layers that are shown in table 6.

Each instance in a multi-class classification task is assigned only one label from a set of predefined labels. In contrast, each instance in multi-label classification can be assigned numerous labels. Multi-class classification problems are typically easier to handle as they are more straightforward, while multi-label classification problems require more complex approaches. We are interested in the performance of our models in both multi-class and multi-label tasks. We have two supervised sets for both multi-class and multi-label training, explained in section 3.1.1.1. On the other hand, BERT is trained on MLM task. It is the task of predicting a masked word in a sentence. This task allows the model to understand the context and meaning of words within a sentence, which is crucial for many NLP tasks. To use such a model for classification, the main method is to add a

linear classification layer and train the model using a labeled training set. The BERT models have been fine-tuned for the classification task using specific hyperparameters, namely an Epoch value of 10 and a Learning Rate of 5.00E-05. These particular values were determined to be the most effective through a grid search, which explored a range of options for the Epoch and Learning Rate parameters. The search space for this task included values of; Epoch:[3,5,10,20], Learning Rate:[5.00E-05, 1.00E-04].

We employ a 3-fold stratified repeated hold-out method in our experiments. A random 10% of the data is kept as a test set, while the remaining 90% is used as a training set. We do this for both multi-label and multi-class datasets. For multi-label dataset we use the iterative stratified split [64, 65]. After the split, we check for the document similarities as it is described above and remove highly similar instances. From the training set, we again select a 10% to form the validation set. The training set, test set and validation set sizes measured by the number of documents can be seen in table 7.

Table 7: Number of documents in training, validation and test sets of multi-class and multi-label sets.

Set	**Fold**	**Training set**	**Validation Set**	**Test Set**
Multi-label	1	38,465	4,266	8,730
	2	36,276	3,997	8,741
	3	36,474	4,026	8,700
Multi-class	1	26,067	2,897	6,854
	2	26,092	2,900	6,854
	3	26,095	2,900	6,854

4 RESULTS & DISCUSSIONS

4.1 9-class multi-class classification: Binary weighting vs. TF-IDF

To set a foundation, we begin by employing conventional machine learning techniques, specifically MNB, SVM, and LR on BoW representation using TF-IDF and binary weighting. Multinomial NB and RBF kernel for SVM were selected due to their widespread application in text classification. The findings in Table 8 indicate that the performance of binary weighting is comparable to or surpasses TF-IDF. For this reason, we only employ the binary weighting in our second experiment.

Table 8: Outcomes for TF-IDF and binary weighting (Macro avg.).

Method	**Weighting**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Naive Bayes	Binary	0.89	0.83	0.62	0.61
	TF-IDF	0.88	0.83	0.59	0.58
Logistic Regression	Binary	**0.95**	0.86	0.86	0.85
	TF-IDF	**0.95**	0.86	0.86	**0.86**
SVM	Binary	**0.95**	**0.87**	**0.87**	**0.86**
	TF-IDF	**0.95**	**0.87**	**0.87**	**0.86**

4.2 9-class multi-class classification: HukukBERT vs. other methods

Our deep learning architecture consists of three layers: an embedding layer, a single layer of Bi-LSTM, and a dense layer for classification. To obtain word embeddings, we utilize the pre-trained fastText model specifically designed for Turkish. Throughout the training process, we keep the embedding layer fixed while employing softmax as the activation function. The model incorporates the following parameters: loss function:'categorical crossentropy', optimizer:'Adam', and a learning rate of 1e-3. After conducting experiments, we determined that the optimal performance is achieved after 32 epochs with a batch size of 256.

To implement transformer models for classification, we utilize the HuggingFace library [61]. Employing grid search for hyper-parameter optimization, we identify the optimal parameters for BERT, DistilBERT, and our domain adapted BERT model which we call HukukBERT. BERT and DistilBERT models yield the best results after 3 epochs

of training with a learning rate of 5e-5. On the other hand, our HukukBERT model achieves its highest score after a single epoch with the same learning rate. For all our models, we use a batch size of four and a weight decay of 1e-3. The training losses of our models can be observed in Figure 4.1.

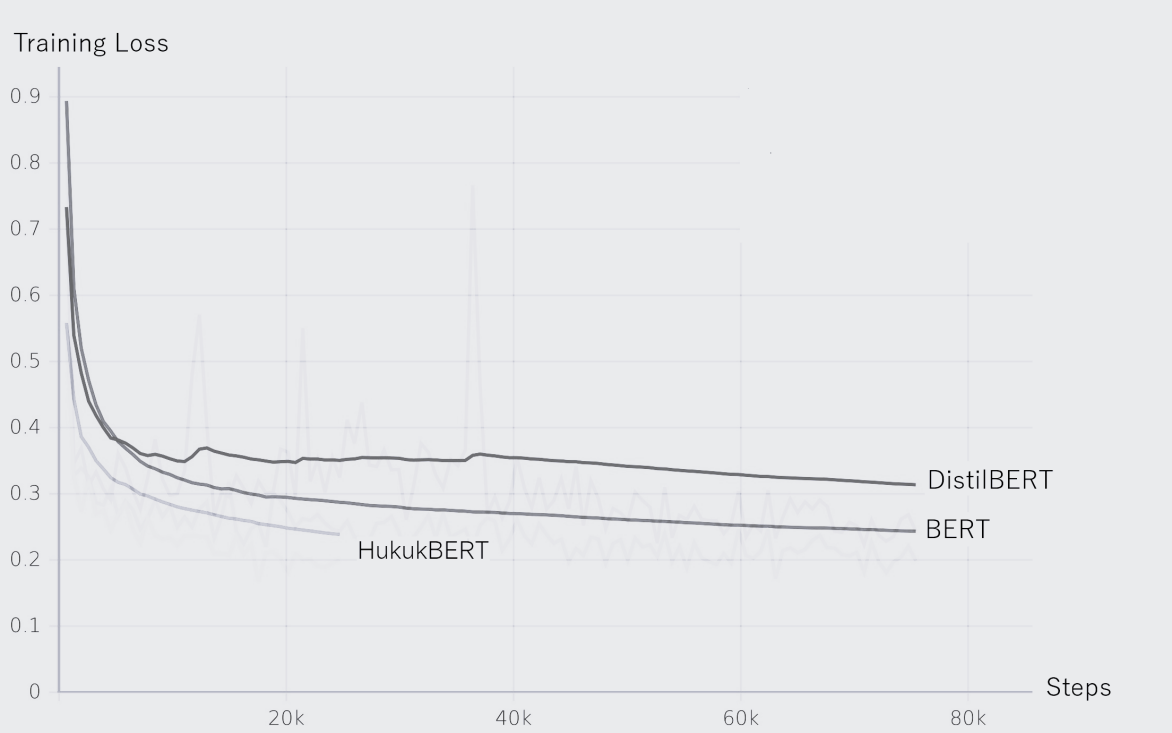

Figure 4.1: Relationship between training loss and training steps for transformers.

The models are evaluated based on their macro average F1 score. Table 9 demonstrates that MNB performs the poorest among the models. However, both SVM and LR outperform the pre-trained BERT. Remarkably, the Bi-LSTM model exhibits the highest recall and achieves a superior F1 score compared to BERT and DistilBERT. When considering resource requirements, SVM, LR, and Bi-LSTM models are comparatively easier to train and yield better results than transformer models designed for general domains. Notably HukukBERT-ALC, our domain adapted BERT model attains the highest scores, indicating that fine-tuning transformer models with domain-specific corpus can significantly enhance performance. Furthermore, it is noteworthy that our domain adapted BERT model achieves this remarkable outcome with a mere 1 epoch of training, while other transformer models undergo 3 epochs of training.

We were surprised to find that Logistic Regression outperformed the more complex

Table 9: Comparative evaluation of crime classification models' performance on Turkish legal text.

Method	Accuracy	Precision	Recall	F1 Score
Naive Bayes	0.89	0.83	0.62	0.61
Logistic Regression	**0.95**	0.86	0.86	0.86
SVM	**0.95**	**0.87**	0.87	0.86
FastText + biLSTM	**0.95**	**0.87**	0.86	0.86
BERT-FT	0.94	0.84	**0.88**	0.85
DistilBERT-FT	0.94	0.85	0.85	0.84
HukukBERT-ALC-FT	**0.95**	0.86	**0.88**	**0.87**

transformer model. We suspect this might be because our 9-class dataset has clear class boundaries, making it a simpler task. Consequently, we decided to tackle a more challenging multi-label classification task with 56 classes in our subsequent experiment.

4.3 56-class multi-class classification: HukukBERT vs. other methods

Table 10 shows the results we have for all the models for the multi-class classification task. Abbreviations after the model name explains the training process of the models. TLC and CCC stands for the corpus that is used in the domain adaptation (continued pre-training) of the model. FF means "freeze first" and the following number indicates the number of layers that is frozen. FT shows that the model itself is fine-tuned for the classification task. Lastly AD means that an additional adapter layer is used with the model and trained for the classification task. So HukukBERT-FF9-CCC-FT-AD: is the model that is domain adapted using CCC corpus while the first 9 layers are frozen. It is then fine-tuned for classification. Lastly an additional adapter layer is used with this model and the adapter is also trained for classification.

Best achieving traditional machine learning is SVM classifier with 50.49% macro F1 score. LR, one of the simplest classifiers is closely follows with 47.81%. Interestingly, these results are better than a deep learning-based classifier, the Bi-LSTM although it is a much more complex algorithm that is even initialized with pre-trained fastText embeddings. Also, using FastText combined with logistic regression underperformed compared to logistic regression with binary weighting. This suggests that FastText embeddings sourced from the Common Crawl might not be ideal for representing legal

documents. Additionally, we compute sentence embeddings by averaging word vectors. In dense documents like ours, this method could dilute the overall meaning due to the abundance of words.

However, when we use a pre-trained transformer based LLM, i.e. BERT, we obtain significantly better than traditional machine learning classifiers and Bi-LSTM with an F1 score of 61.34% averaged over three-fold experiment results. More importantly, all of the domain-adapted BERT models have higher F1 scores compared to the general domain Turkish BERT model. This shows the benefit of domain adaptation in a previously untested domain of Turkish legal domain which is a challenging domain due to the heavy use of legal terminology and quite complex and very long sentence formations. From our domain adapted and fine-tuned models HukukBERT-FF6-CCC-FT achieves the best F1 score of all of the domain adapted BERT models. It improves the score of the Turkish BERT (dbmdz/bert-base-turkish-cased) model by more than 2%.

Table 10: Multi class 56 class, classification performance of our models.

Method	Base Model	Accuracy	Precision	Recall	F1 Score
Naïve Bayes	binary	46.16 ± 0.05	26.77 ± 1.36	14.98 ± 0.07	14.22 ± 0.21
Logistic Reg.	binary	73.38 ± 0.14	71.55 ± 1.66	44.08 ± 0.26	47.81 ± 0.66
SVC	binary	74.17 ± 0.32	75.28 ± 2.06	46.42 ± 0.26	50.49 ± 0.33
Logistic Reg.	Fasttext (cc300)	49.68 ± 0.18	48.76 ± 1.32	24.24 ± 0.18	28.07 ± 0.14
BiLSTM	Fasttext (cc300)	65.55 ± 0.37	52.26 ± 2.26	45.84 ± 1.39	45.64 ± 1.25
BERT Classifier	BERT-FT	75.13 ± 3.10	67.47 ± 1.24	59.43 ± 0.63	61.34 ± 0.32
	HukukBERT-TLC-FT	$\mathbf{78.67} \pm 0.35$	70.54 ± 2.65	59.52 ± 0.96	62.31 ± 1.17
	HukukBERT-CCC-FT	78.37 ± 0.32	68.45 ± 0.55	61.31 ± 0.63	63.33 ± 0.38
	HukukBERT-FF9-CCC-FT	76.11 ± 3.09	68.83 ± 0.77	60.96 ± 0.77	63.19 ± 0.58
	HukukBERT-FF6-CCC-FT	78.49 ± 0.52	69.32 ± 0.82	61.27 ± 0.65	$\mathbf{63.64} \pm 0.68$
Adapter Classifier	BERT-AD	75.32 ± 0.27	68.54 ± 0.90	50.52 ± 0.35	53.68 ± 0.35
	HukukBERT-TLC-AD	73.87 ± 0.43	65.67 ± 2.40	46.85 ± 0.14	49.45 ± 0.13
	HukukBERT-CCC-AD	77.99 ± 0.44	$\mathbf{75.05} \pm 2.78$	55.99 ± 0.48	59.55 ± 0.81
	HukukBERT-CCC-FT-AD	78.54 ± 0.50	70.24 ± 0.90	59.83 ± 1.21	62.50 ± 1.05
	HukukBERT-FF9-CCC-FT-AD	75.37 ± 3.23	66.84 ± 0.31	60.74 ± 0.56	62.44 ± 0.11
	HukukBERT-FF6-CCC-FT-AD	78.15 ± 0.54	67.84 ± 0.84	$\mathbf{61.44} \pm 0.61$	63.39 ± 0.63

Interestingly, HukukBERT-CCC-FT, which was adapted using the Civil Court Corpus, performs better than HukukBERT-TLC-FT, which was adapted using the larger Turkish Legal Corpus. Even though the Civil Court Corpus is smaller in size, it yielded a higher F1 score. The likely reason for this superior performance is that the Civil Court Corpus, which consists of civil court decisions from the CoC, closely mirrors our labeled data that is made up of criminal court decisions from the same CoC. This resemblance in data sources probably enhances the model's accuracy.

Furthermore, it's crucial to note that using adapters instead of directly fine-tuning

the model for classification resulted in notably lower scores. This is evident when comparing BERT-FT with BERT-AD, HukukBERT-TLC-FT with HukukBERT-TLC-AD, and HukukBERT-CCC-FT with HukukBERT-CCC-AD. This trend persists even when adding adapters to an already fine-tuned model. This suggests that increasing model complexity with adapters doesn't necessarily improve performance for the multi-class classification task. The t-test comparison between HukukBERT-CCC-FT and HukukBERT-CCC-FT-AD shows a probability of 35.09%, indicating the difference might not be statistically significant. However, the domain adaptation, as observed in the difference between BERT-FT and HukukBERT-CCC-FT, appears to be significant, with a t-test probability of just 0.4%.

4.4 56-class multi-label classification: HukukBERT vs. other methods

Multi-label classification performances of our models are shown in table 11. The results we have are consistent with the multi-class experiment results. Interestingly, Bi-LSTM now outperforms traditional machine learning classifiers. This was not the case in multi-class results. As the task at hand gets harder we see the benefits of using a neural network.

The SVM classifier is still best among the traditional classifiers by a large margin. This time LR is not a close follower as in the multi-class results. BERT again achieves significantly higher F1 scores. In contrast to the multi-class classification findings, the strategy of adapting the BERT language model using the TLC corpus doesn't offer a performance advantage over the general domain Turkish BERT model. Indeed, employing the more extensive, yet unrelated, legal corpus (HukukBERT-TLC-FT) appears to hinder the performance, consistent with the multi-class results. The TLC corpus constitutes one third of the pretraining corpus used for the Turkish BERT. We believe that using such a large corpus might have led to a "forgetting" effect.

The model domain adapted using the CCC corpus (HukukBERT-CCC-FT) shows a modest performance boost (62.82% compared to 61.25%), though it comes with a greater standard deviation. Table 12 shows the performance of the HukukBERT-CCC-FT model for each class, as expected the model performs much better for classes that have bigger support in the training set. The F1 score goes up to 97.20% for "Kasten Öldürme," representing the 14th most represented class.

Table 11: Multi label 56 class, classification performance of our models.

Method	Base Model	Precision	Recall	F1 Score
Naïve Bayes	binary	26.52 ± 1.71	6.90 ± 0.19	9.01 ± 0.26
Logistic Reg.	binary	65.57 ± 0.91	27.73 ± 0.66	34.86 ± 0.72
SVC	binary	81.30 ± 1.05	35.75 ± 0.50	44.27 ± 0.48
Logistic Reg.	Fasttext (cc300)	51.92 ± 1.89	23.95 ± 0.19	29.90 ± 0.34
BiLSTM	Fasttext (cc300)	52.64 ± 2.93	45.80 ± 2.34	46.50 ± 2.10
BERT Classifier	BERT-FT	73.57 ± 0.77	56.32 ± 0.46	61.25 ± 0.38
	HukukBERT-TLC-FT	71.63 ± 1.51	51.19 ± 0.37	56.25 ± 0.24
	HukukBERT-CCC-FT	73.33 ± 0.15	58.55 ± 0.66	62.82 ± 1.06
	HukukBERT-FF9-CCC-FT	$\mathbf{74.72} \pm 1.00$	56.99 ± 0.96	61.89 ± 1.31
	HukukBERT-FF6-CCC-FT	72.89 ± 1.40	56.51 ± 0.66	60.96 ± 0.53
Adapter Classifier	BERT-AD	70.54 ± 1.62	43.12 ± 1.26	49.31 ± 1.18
	HukukBERT-TLC-AD	57.09 ± 2.66	34.76 ± 0.45	39.04 ± 0.49
	HukukBERT-CCC-AD	70.26 ± 1.03	43.43 ± 1.18	49.22 ± 1.14
	HukukBERT-CCC-FT-AD	71.07 ± 1.32	$\mathbf{59.49} \pm 1.17$	$\mathbf{63.05} \pm 1.49$
	HukukBERT-FF9-CCC-FT-AD	70.22 ± 0.95	58.22 ± 1.11	61.73 ± 1.42
	HukukBERT-FF6-CCC-FT-AD	68.85 ± 0.49	57.88 ± 0.90	61.17 ± 0.72

The results suggest that pre-trained transformer-based Large Language Models (LLMs), such as BERT, is more effective than traditional machine learning or Bi-LSTM models, especially when the domain is highly specialized like the Turkish legal domain. The findings also highlight the importance of domain adaptation in improving the performance of pre-trained models. An important finding of the study is; adapting to a domain using a corpus that is similar to the task at hand is more beneficial than using a larger and more general corpus for domain adaptation. Another noteworthy result from our study is that, in our experiments, the use of adapters did not have a significant effect on the results, and in certain cases, it even resulted in worse performance. This may suggest that adapters may not be well-suited for task adaptation, unlike language adaptation and domain adaptation.

Table 12: HukukBERT-CCC-FT results for each class in multilabel classification.

Label	Precision	Recall	F1-score
4733 Sayılı Kanuna Aykırılık	64.65	75.90	69.82
4926 Sayılı Kanuna Aykırılık	46.82	28.57	35.03
5015 Sayılı Kanuna Aykırılık	73.30	42.67	53.61
5187 Sayılı Kanuna Aykırılık	93.70	65.63	76.79
5607 Sayılı Kanuna Aykırılık	88.35	89.83	89.05
5809 Sayılı Kanuna Aykırılık	92.92	86.27	89.47

5846 Sayılı Kanuna Aykırılık	69.78	72.37	70.61
6136 Sayılı Kanuna Aykırılık	55.45	56.25	50.62
6831 Sayılı Kanuna Aykırılık	84.69	53.75	64.96
Banka veya Kredi Kartlarının Kötüye Kullanılması	70.87	50.53	58.07
Başkasına Ait Kimlik veya Kimlik Bilgilerinin Kullanılması	71.83	58.35	64.18
Cinsel Saldırı	54.80	49.56	51.92
Cinsel Taciz	64.74	33.82	44.34
Dolandırıcılık	85.90	77.62	81.50
Fuhuş	78.11	41.32	53.78
Genel Güvenliği Tehlikeye Sokacak Şekilde Taksirle Yangına Neden Olma	90.11	62.22	72.58
Görevi Yaptırmamak için Direnme	58.05	59.72	58.83
Göçmen Kaçakçılığı	40.60	38.11	38.15
Güveni Kötüye Kullanma	72.39	40.87	52.12
Hakaret	76.86	74.00	75.40
Hakkı Olmayan Yere Tecavüz	73.37	43.86	54.31
Hırsızlık	81.66	78.98	80.23
Hizmet Nedeniyle Güveni Kötüye Kullanma	68.81	56.99	62.31
Hükümlü veya Tutuklunun Kaçması	79.62	63.01	58.24
İftira	49.40	55.11	52.05
İhaleye Fesat Karıştırma	90.74	78.42	83.95
İmar Kirliliğine Neden Olma	98.82	82.23	89.73
İş ve Çalışma Hürriyetinin İhlali	86.67	43.38	56.31
Kamu Görevlisinin Resmi Belgede Sahteciliği	48.48	26.47	34.13
Karşılıksız Yararlanma	48.05	57.41	52.30
Kasten Yaralama	85.63	86.93	86.27
Kasten Öldürme	97.44	96.97	97.20
Kişilerin Huzur ve Sükununu Bozma	50.63	29.64	36.99
Kişiyi Hürriyetinden Yoksun Kılma	75.06	69.14	71.98
Konut Dokunulmazlığının İhlali	77.23	65.38	70.68

Kumar Oynanması İçin Yer ve İmkan Sağlama	79.53	19.03	30.44
Mala Zarar Verme	69.07	63.38	66.05
Marka Hakkına Tecavüz	82.92	72.49	77.22
Muhafaza Görevini Kötüye Kullanma	63.99	58.51	58.23
Mühür Bozma	62.82	26.45	37.21
Neticesi Sebebiyle Ağırlaşmış Yaralama	77.38	52.08	58.39
Nitelikli Dolandırıcılık	83.76	87.73	85.65
Nitelikli Hırsızlık	63.46	63.39	63.30
Parada Sahtecilik	85.71	39.39	53.42
Resmi Belgede Sahtecilik	76.74	76.17	76.38
Resmi Belgenin Düzenlenmesinde Yalan Beyan	71.11	38.89	49.87
Suç Eşyasının Satın Alınması veya Kabul Edilmesi	80.60	64.25	71.42
Suç Üstlenme	50.00	3.42	6.36
Tehdit	85.72	81.33	83.45
Vergi Usul Kanununa Muhalefet	89.57	91.38	90.41
Yağma	80.72	80.50	80.6
Zimmet	87.34	90.74	88.70
Çocuğun Cinsel İstismarı	73.81	83.01	78.12
Çocuğun Kaçırılması ve Alıkonulması	74.62	37.64	50.03
Özel Belgede Sahtecilik	61.42	41.18	49.06
Şantaj	60.51	16.67	26.10
Macro Avg.	73.33	58.55	62.82

This study adds to the growing body of Turkish language model studies and Turkish legal domain studies. It is one of the early studies in the use of language models in Turkish and as far as we know the first study that uses language models for Turkish legal document classification. The research also looks into the performance of various machine learning and deep learning techniques, in addition to the influence of domain adaptation using pre-trained large language models, which has not been thoroughly investigated in

the literature for Turkish legal document classification. The implication of this study is that the success of the language model in the multi-class multi-label design with a larger class size suggests that it can be used effectively in other tasks related to the Turkish language domain, such as named entity recognition and text generation. This is because the study's design closely mimics real-world situations, indicating that the language model can perform well in similar real-world scenarios.

5 CONCLUSION

Use of Artificial Intelligence (AI) technologies as a decision support system, more specifically Natural Language Processing (NLP) and machine learning (ML) in the legal domain have the ability to significantly enhance the throughput of legal systems while also enhancing the lives of individuals and legal professionals. Since tremendous amounts of documents are created and processed mostly manually every day in legal systems, NLP technologies such as recent transformer based Large Language Models (LLMs) has great potential in helping to process this textual data. Regrettably, NLP and text mining in the legal domain has not received much research attention. It is especially the case for Turkish legal documents. There exist only a handful of studies in this domain and they are mostly using small datasets. This is due to the difficulty of the legal language, a large amount of highly technical vocabulary, and the formulation of very long and complex sentences. The difficulty of legal language also creates a barrier to understanding and evaluating the NLP results for computer scientists that can only be overcome by collaborating legal professionals that can understand the meaning of legal texts. Another reason could be the lack of supervised and unsupervised benchmark datasets that are necessary for modern NLP techniques, such as LLMs and other deep learning models. This is especially the case for the Turkish language. Having such benchmark datasets, pre-trained LLMs, and baseline results to build upon is essential for an NLP field to progress. More recently, the trend is to share pre-trained LLMs on a specific language domain. Sharing domain-specific pre-trained or fine-tuned LLMs will also greatly help the researchers.

In this study, we first introduce much-needed large supervised and unsupervised datasets for future NLP studies in the Turkish legal domain that are collected from several different public resources, then carefully cleaned and preprocessed. After this, we conduct extensive experiments for crime classification of higher-court decisions both in multi-class and multi-label classification settings using both traditional machine learning and deep learning-based classifiers, including the LLMs. Following this we focus on the domain adaptation of a general domain LLM as the texts in the legal domain greatly differ, with the hypothesis that fine-tuning a LLM using domain-specific data will improve the classification models based on this. Again, we conduct extensive experiments using different datasets and fine-tuning approaches to validate the hypothesis.

Our analysis of different machine learning models reveals the notable potential of

transformer-based models, specifically domain adapted BERT models, for text classification tasks in specialized domains, such as Turkish legal text. It's been demonstrated that these models, in particular our domain adapted HukukBERT, significantly outperformed traditional machine learning models like MNB, SVM, and LR, and also surpassed the base BERT and DistilBERT models.

Furthermore, we found that the benefit of domain adaptation is quite significant, with our domain adapted HukukBERT model, trained with a domain-specific corpus, delivering superior results. Interestingly, a smaller corpus that was more closely aligned with the specific domain yielded better results than a larger, more general one. This highlights the importance of choosing the right data for fine-tuning in domain adaptation.

In our multi-class classification study, HukukBERT models outperformed other models, showing its efficiency in tasks where the input can only belong to one category. Meanwhile, in multi-label classification tasks, while BERT models achieve significantly higher F1 scores, our HukukBERT models give conflicting results. HukukBERT, when trained on a smaller corpus, performed better, while its performance was worse when trained on a larger corpus compared to the standard BERT. In the multi-label settings, the most optimal results were obtained when HukukBERT-CCC was trained for classification tasks and simultaneously used as the foundation for an adapter. Nevertheless, the enhancements made on HukukBERT-CCC without using adapters didn't significantly impact the results.

On the other hand, our experiments did not find a significant improvement with the use of adapters. Instead, the inclusion of these components occasionally led to a decline in performance. This could suggest that the use of adapters may not always be beneficial and should be carefully evaluated depending on the specific task and dataset.

These findings contribute valuable insights to the field of machine learning for specialized text classification tasks, suggesting that domain adapted transformer models provide promising results. Moreover, the findings underscore the necessity of careful selection and application of methods for domain adaptation and model fine-tuning. In the context of the evolving machine-learning landscape, these findings provide direction for further research and practical applications.

There are various prospective study routes that can build on the findings reported in this paper. One potential direction is to explore the performance of the domain adapted language model on a wider range of legal language tasks and datasets, including tasks

related to legal reasoning and argumentation. Another promising area for future work is to investigate the impact of different training strategies and hyperparameters on the performance of the model, such as the use of additional data sources or alternative pre-training methods.

6 REFERENCES

[1] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[2] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[3] Andrew Vold and Jack G. Conrad. "Using Transformers to Improve Answer Retrieval for Legal Questions". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*. ICAIL '21. São Paulo, Brazil: Association for Computing Machinery, 2021, pp. 245–249. ISBN: 9781450385268. DOI: 10.1145/3462757.3466102.

[4] Marc Queudot, Eric Charton, and Marie Jean Meurs. "Improving Access to Justice with Legal Chatbots". In: *Stats* 3.3 (2020), pp. 356–375. ISSN: 2571-905X. DOI: 10.3390/stats3030023.

[5] Giyaseddin Bayrak and Abdul Majeed Issifu. "Domain-Adapted BERT-based Models for Nuanced Arabic Dialect Identification and Tweet Sentiment Analysis". In: *Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP)*. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 425–430.

[6] Juuso Eronen, Michal Ptaszynski, and Fumito Masui. "Zero-shot cross-lingual transfer language selection using linguistic similarity". In: *Information Processing & Management* 60.3 (2023), p. 103250. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2022.103250.

[7] Ilias Chalkidis et al. "LEGAL-BERT: The Muppets straight out of Law School". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Nov. 2020, pp. 2898–2904. DOI: 10.18653/v1/2020.findings-emnlp.261.

[8] Gıyaseddin Bayrak et al. "Deep learning-based brain hemorrhage detection in ct reports". In: *Challenges of Trustable AI and Added-Value on Health*. IOS Press, 2022, pp. 866–867.

[9] Emre Mumcuoğlu et al. “Natural language processing in law: Prediction of outcomes in the higher courts of Turkey”. In: *Information Processing & Management* 58.5 (2021). ISSN: 0306-4573. DOI: 10.1016/j.ipm.2021.102684.

[10] Cihan Erdoğanyılmaz and Berkay Mengünoğul. “An Original Natural Language Processing Approach to Language Modeling in Turkish Legal Corpus: Improving Model Performance with Domain Classification by Using Recurrent Neural Networks”. In: *2022 Innovations in Intelligent Systems and Applications Conference (ASYU)*. 2022, pp. 1–6. DOI: 10.1109/ASYU56188.2022.9925363.

[11] Berna Altınel and Murat Can Ganiz. “Semantic text classification: A survey of past and recent advances”. In: *Information Processing & Management* 54.6 (2018), pp. 1129–1153. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2018.08.001.

[12] Tamás Orosz et al. “Evaluating Human versus Machine Learning Performance in a LegalTech Problem”. In: *Applied Sciences* 12.1 (2021), p. 297. DOI: 10.3390/app12010297.

[13] Octavia Maria Şulea et al. “Exploring the Use of Text Classification in the Legal Domain”. In: *Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts (ASAIL)*. London, United Kingdom, 2017.

[14] Jerrold Soh, How Khang Lim, and Ian Ernst Chai. “Legal Area Classification: A Comparative Study of Text Classifiers on Singapore Supreme Court Judgments”. In: *Proceedings of the Natural Legal Language Processing Workshop 2019*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 67–77. DOI: 10.18653/v1/W19-2208.

[15] Chi Sun et al. “How to Fine-Tune BERT for Text Classification?” In: *Chinese Computational Linguistics*. Ed. by Maosong Sun et al. Cham: Springer International Publishing, 2019, pp. 194–206. ISBN: 978-3-030-32381-3.

[16] Haihua Chen et al. “A comparative study of automated legal text classification using random forests and deep learning”. In: *Information Processing & Management* 59.2 (2022), p. 102798. DOI: https://doi.org/10.1016/j.ipm.2021.102798.

[17] Benjamin Clavié et al. *LegaLMFiT: Efficient Short Legal Text Classification with LSTM Language Model Pre-Training*. 2021. arXiv: 2109.00993 [cs.CL].

[18] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[19] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[20] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *J. Mach. Learn. Res.* 21.1 (Jan. 2020). ISSN: 1532-4435.

[21] Christopher D. Manning. "Human Language Understanding & Reasoning". In: *Daedalus* 151.2 (May 2022), pp. 127–138. ISSN: 0011-5266. DOI: 10.1162/daed_a_01905.

[22] Stefan Schweter. *BERT - BERT models for Turkish*. Version 1.0.0. Apr. 2020. DOI: 10.5281/zenodo.3770924.

[23] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747.

[24] Linting Xue et al. "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 483–498. DOI: 10.18653/v1/2021.naacl-main.41.

[25] Emrecan Çelik and Tuğba Dalyan. "Unified benchmark for zero-shot Turkish text classification". In: *Information Processing & Management* 60.3 (2023), p. 103298. ISSN: 0306-4573. DOI: 10.1016/j.ipm.2023.103298.

[26] Abdullatif Köksal and Arzucan Özgür. "Twitter Dataset and Evaluation of Transformers for Turkish Sentiment Analysis". In: *2021 29th Signal Processing and Communications Applications Conference (SIU)*. 2021, pp. 1–4. DOI: 10.1109/SIU53274.2021.9477814.

[27] Fatih Çağatay Akyön et al. "Automated question generation and question answering from Turkish texts". In: *Turkish Journal of Electrical Engineering and Computer Sciences* 30.5 (2022), pp. 1931–1940. DOI: 10.55730/1300-0632.3914.

[28] Or Sharir, Barak Peleg, and Yoav Shoham. "The cost of training nlp models: A concise overview". In: *arXiv preprint arXiv:2004.08900* (2020).

[29] Masahiro Suzuki et al. "Constructing and analyzing domain-specific language model for financial text mining". In: *Information Processing & Management* 60.2

(2023), p. 103194. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2022.103194.

[30] Junguang Jiang et al. *Transferability in Deep Learning: A Survey*. 2022. arXiv: 2201.05867 [cs.LG].

[31] Chaojun Xiao et al. "Lawformer: A pre-trained language model for Chinese legal long documents". In: *AI Open* 2 (2021), pp. 79–84. ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2021.06.003.

[32] Lucia Zheng et al. "When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset of 53,000+ Legal Holdings". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*. ICAIL '21. São Paulo, Brazil: Association for Computing Machinery, 2021, pp. 159–168. ISBN: 9781450385268. DOI: 10.1145/3462757.3466088.

[33] Jieh-Sheng Lee and Jieh Hsiang. "Patent classification by fine-tuning BERT language model". In: *World Patent Information* 61 (2020), p. 101965. ISSN: 0172-2190. DOI: https://doi.org/10.1016/j.wpi.2020.101965.

[34] Iz Beltagy, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer". In: *arXiv:2004.05150* (2020).

[35] Nut Limsopatham. "Effectively Leveraging BERT for Legal Document Classification". In: *Proceedings of the Natural Legal Language Processing Workshop 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 210–216. DOI: 10.18653/v1/2021.nllp-1.22.

[36] Dimitris Mamakas et al. "Processing Long Legal Documents with Pre-trained Transformers: Modding LegalBERT and Longformer". In: *Proceedings of the Natural Legal Language Processing Workshop 2022*. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 130–142.

[37] Neil Houlsby et al. "Parameter-Efficient Transfer Learning for NLP". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 2790–2799.

[38] Amir Rosenfeld and John K. Tsotsos. "Intriguing Properties of Randomly Weighted Networks: Generalizing While Learning Next to Nothing". In: *2019 16th Conference on Computer and Robot Vision (CRV)*. 2019, pp. 9–16. DOI: 10.1109/CRV.2019.00010.

[39] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. "BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1–9. DOI: 10.18653/v1/2022.acl-short.1.

[40] Jonathan Frankle, David J. Schwab, and Ari S. Morcos. "Training BatchNorm and Only BatchNorm: On the Expressive Power of Random Features in {CNN}s". In: *International Conference on Learning Representations*. 2021.

[41] Jonas Pfeiffer et al. "AdapterHub: A Framework for Adapting Transformers". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 46–54. DOI: 10.18653/v1/2020.emnlp-demos.7.

[42] Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. "MultiEURLEX - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6974–6996. DOI: 10.18653/v1/2021.emnlp-main.559.

[43] Utku Umur Acikalin, Benan Bardak, and Mucahid Kutlu. "Turkish sentiment analysis using BERT". In: *2020 28th Signal Processing and Communications Applications Conference (SIU)*. IEEE. 2020, pp. 1–4. DOI: 10.1109/SIU49456.2020.9302492.

[44] Azer Çelikten and Hasan Bulut. "Turkish Medical Text Classification Using BERT". In: *2021 29th Signal Processing and Communications Applications Conference (SIU)*. 2021, pp. 1–4. DOI: 10.1109/SIU53274.2021.9477847.

[45] Zekeriya Anil Guven. "Comparison of BERT Models and Machine Learning Methods for Sentiment Analysis on Turkish Tweets". In: *2021 6th International Conference on Computer Science and Engineering (UBMK)*. 2021, pp. 98–101. DOI: 10.1109/UBMK52708.2021.9559014.

[46] Piotr Bojanowski et al. "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5 (June 2017), pp. 135–146. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00051.

[47] Alan Ramponi and Barbara Plank. "Neural Unsupervised Domain Adaptation in NLP—A Survey". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee

on Computational Linguistics, Dec. 2020, pp. 6838–6855. DOI: 10.18653/v1/2020.coling-main.603.

[48] Dilara Torunoğlu et al. "Analysis of preprocessing methods on classification of Turkish texts". In: *2011 International Symposium on Innovations in Intelligent Systems and Applications*. 2011, pp. 112–117. DOI: 10.1109/INISTA.2011.5946084.

[49] Armand Joulin et al. "Bag of Tricks for Efficient Text Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, May 2017, pp. 427–431.

[50] Peng Zhou et al. "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: Association for Computational Linguistics, Aug. 2016, pp. 207–212. DOI: 10.18653/v1/P16-2034.

[51] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108* (2019).

[52] Liu Zhuang et al. "A Robustly Optimized BERT Pre-training Approach with Post-training". English. In: *Proceedings of the 20th Chinese National Conference on Computational Linguistics*. Huhhot, China: Chinese Information Processing Society of China, Aug. 2021, pp. 1218–1227. URL: https://aclanthology.org/2021.ccl-1.108.

[53] Zhilin Yang et al. "Xlnet: Generalized autoregressive pretraining for language understanding". In: *Advances in neural information processing systems* 32 (2019).

[54] Zhenzhong Lan et al. "Albert: A lite bert for self-supervised learning of language representations". In: *arXiv preprint arXiv:1909.11942* (2019).

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

[56] Yathrib Alqahtani, Nora Al-Twairesh, and Ahmed Alsanad. "Improving sentiment domain adaptation for Arabic using an unsupervised self-labeling framework". In: *Information Processing & Management* 60.3 (2023), p. 103338. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2023.103338.

[57] Melike Tutkan, Murat Can Ganiz, and Selim Akyokuş. "Helmholtz principle based supervised and unsupervised feature selection methods for text mining". In:

Information Processing & Management 52.5 (2016), pp. 885–910. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2016.03.007.

[58] Fazli Can et al. "Information retrieval on Turkish texts". In: *Journal of the American Society for Information Science and Technology* 59.3 (2008), pp. 407–421.

[59] Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. "Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures". en. In: ed. by Piotr Bański et al. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019. Mannheim: Leibniz-Institut für Deutsche Sprache, 2019, pp. 9–16. DOI: 10.14618/ids-pub-9021.

[60] Mikko Aulamo and Jörg Tiedemann. "The OPUS Resource Repository: An Open Package for Creating Parallel Corpora and Machine Translation Services". In: *Proceedings of the 22nd Nordic Conference on Computational Linguistics*. Turku, Finland: Linköping University Electronic Press, Sept. 2019, pp. 389–394.

[61] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6.

[62] Abolfazl Farahani et al. "A Brief Review of Domain Adaptation". In: *Advances in Data Science and Information Engineering*. Ed. by Robert Stahlbock et al. Cham: Springer International Publishing, 2021, pp. 877–894. ISBN: 978-3-030-71704-9.

[63] Jonas Pfeiffer et al. "AdapterFusion: Non-Destructive Task Composition for Transfer Learning". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 487–503. DOI: 10.18653/v1/2021.eacl-main.39.

[64] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. "On the Stratification of Multi-label Data". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Dimitrios Gunopulos et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 145–158. ISBN: 978-3-642-23808-6.

[65] Piotr Szymanski and Tomasz Kajdanowicz. "A scikit-based Python environment for performing multi-label classification". In: *CoRR* abs/1702.01460 (2017). arXiv: 1702.01460.

Onur Akça

Experience

- **Accenture IX** — Istanbul, Turkey
  Digital Tech Developer Specialist — *February 2023 - Continue*
- **FIT Global** — Istanbul, Turkey
  ABAP Team Lead — *June 2022 - Oct 2022*
  Senior ABAP Developer — *Oct 2018 - June 2022*
- **ARETE Consulting** — Istanbul, Turkey
  ABAP Developer — *Nov. 2014 - Oct. 2018*

Publications

- O. Akça, G. Bayrak, A. M. Issifu and M. C. Ganiz, "Traditional Machine Learning and Deep Learning-based Text Classification for Turkish Law Documents using Transformers and Domain Adaptation," 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), 2022, pp. 1-6, doi: 10.1109/INISTA55318.2022.9894051.

Education

- **Marmara University** — Istanbul, Turkey
  Master of Science in Computer Engineering — *Oct. 2020 – Cont.*
- **Sabanci University** — Istanbul, Turkey
  Bachelor of Computer Science and Engineering — *Sep. 2008 – July. 2014*