T.R.
İSTANBUL UNIVERSITY
INSTITUTE OF SCIENCES

Ph.D. THESIS

IMPROVING THE PERFORMANCE OF NLP TASKS IN LEGAL TECH

Farnaz ZEIDI

Department of Informatics

Programme of Informatics

ADVISOR
Assoc. Prof.Dr. Çiğdem EROL

Prof. Dr. Mehmet Fatih AMASYALI

Ocak, 2024

İSTANBUL

This work has been accepted as a Ph.D. Thesis in **Department of Informatics**, **Programme of Informatics** by the following jury on 16.01.2024.

Thesis Jury

Assoc.Prof. Çiğdem EROL (Advisor)
İstanbul University
Department of Informatics

Prof. Dr. Sevinç GÜLSEÇEN
İstanbul University
Department of Informatics

Prof. Dr. Gökhan SİLAHTAROĞLU
Istanbul Medipol University
Faculty of Business and Management Sciences

Prof. Dr. Küçük Hüseyin KOÇ
İstanbul University - Cerrahpaşa
Faculty Of Forest

Assoc.Prof. Murat GEZER
İstanbul University
Department of Informatics

Ethic Declaration

As required by the 9/2 and 22/2 articles of the Graduate Education Regulation which was published in the Official Gazette on 20.04.2016, this graduate thesis is reported as in accordance with criteria determined by the Institute of Sciences by using the plagiarism software to which Istanbul University is a subscriber.

Research Fund

This thesis supported by the project numbered …….of İstanbul University Scientific Research Projects Executive Secreteriat.

This thesis supported by the project numbered …….of ……….

Publications Produced From Thesis

PREFACE

I proudly present my doctoral thesis, which I finished after lots of hard work. This thesis is a very important part of my academic journey, and I am really thankful to everyone who helped me.

I am thankful to my great supervisors, **Assoc. Prof. Dr. Çiğdem EROL** and **Prof. Dr. Mehmet Fatih AMASYALI**. They gave me fantastic advice, helpful feedback, and shared their important knowledge. Their support helped me a lot with this thesis. I also deeply respect and appreciate **Prof. Dr. Sevinç GÜLSEÇEN**, who leads the Informatics Department at Istanbul University. She has been like a caring mother to me and other students from different countries, supporting us so much.

My research began with collaborating alongside **Dr. Ali GOKSU**, who heads the NewMind Company. I am grateful to him for providing me with abundant information and consistent support throughout the project. I also extend my thanks to the entire NewMind team, especially **Rayene BECH**. Their contributions were invaluable, and without their support, I would not have completed this thesis. During this collaboration, I had the privilege of meeting **Assoc. Prof. Dr. Ali POLAT**. I want to express special gratitude to him. He was a crucial mentor, not only for my studies but also for significant moments in my life.

I would also like to extend special thanks to the "**Yurtdışı Türkler ve Akraba Topluluklar Başkanlığı**" (YTB). Without their support, the beginning of this journey would have been impossible. Their trust in me and the scholarship they provided have not only changed my life but also the lives of thousands of other students around the world. Throughout this academic journey, the support from this organization has played a significant role in my success.

Finally, I want to thank my dear parents. Even though my father, **Farhad ZEIDI**, is not here with us now, his constant support and belief in my dreams keep inspiring me. My mother, **Zarin Taj ZEIDI**, with her never-ending love and encouragement, has given me strong support. Her strength helped me face and overcome all the difficulties I faced.

This thesis is a reflection of the collective efforts, mentorship, and encouragement of these exceptional individuals, as well as the persistent support of my family. I am thankful to these amazing people and my family. I am excited about what is coming next in my academic and professional journey. I have learned a lot from those who helped me, and I am ready for the future with their wisdom in my mind.

Ocak 2024 Farnaz ZEIDI

TABLE OF CONTENTS

Page Number

PREFACE .... **iv**
TABLE OF CONTENTS .... **v**
LIST OF FIGURES .... **viii**
LIST OF TABLES .... **ix**
ABSTRACT .... **xii**
SUMMARY .... **xiv**
1. INTRODUCTION **1**
1.1. OBJECTIVE OF THE STUDY 3
1.2. SCOPE AND IMPORTANCE OF THE STUDY 3
1.3. LIMITATIONS OF THE STUDY 4
1.4. THESIS STRUCTURE 4
2. GENERAL PARTS **6**
2.1. NLP IN THE TURKISH LANGUAGE 6
2.2. A BRIEF HISTORY OF LANGUAGE MODELS 7
2.2.1. Statistical Language Models 7
2.2.2. Neural Language Models 8
2.2.3. Pretrained Language Models 8
2.2.4. Large Language Models 8
2.3. THE BIRTH OF TRANSFORMERS 9
2.3.1. Architecture Layout of the Transformer 9
2.3.2. Positional Encoding 12
2.3.3. Self Attention 13
2.4. BERT MODEL 15
2.4.1. Architecture Layout of the BERT 15
2.4.2. Various BERT Configurations 16
2.4.3. BERT's Input Representation Layers 17
2.4.4. BERT's Tokenizer 19
2.5. PRE-TRAINING TASKS 20
2.5.1. MLM 21
2.5.2. NSP 23

2.5.3. SOP 24
2.6. FINE-TUNING TASKS 25
2.6.1. NER 26
2.6.2. Multi-label Classification 26
2.7. TF-IDF 27
2.8. LITERATURE REVIEW 28
2.8.1. Enhancements to the BERT Model 28
2.8.1.1. Non-Legal Domain 28
2.8.1.2. Legal Domain 33
2.8.2. Turkish Language-Based Enhancements to the BERT Model 34
2.8.2.1. Non-Legal Domain 35
2.8.2.2. Legal Domain 36
3. MATERIAL AND METHODS 39
3.1. DATASET 39
3.1.1. Corpus in Pre-Training Stage 39
3.1.2. Labeled Data in Fine_Tuning Stage 42
3.1.2.1. Labeled Data in NER Task 43
3.1.2.2. Labeled Data in Multi-label Text Classification Task 45
3.2. METHODOLOGY 47
3.2.1. Original BERT 47
3.2.2. Modified BERT Models 48
3.2.2.1. Replacing NSP with SOP 49
3.2.2.2. Retaining MLM and Removing NSP 49
3.2.2.3. Combining MLM with TF-IDF 50
3.2.2.4. Combined Proposed Models: 51
3.3. RESEARCH INQUIRIES 52
3.3.1. Pre-Training Strategies: Weights Transfer vs. From Scratch 53
3.3.2. Impact of Corpus Size in Pre-Training 53
3.3.3. Comparative Analysis: Proposed Models vs. BERTurk Model 53
3.4. MODELS PERFORMANCE EVALUATION 53
3.4.1. Evaluation Methods 53
3.4.2. Evaluation Criteria 54
3.5. TECHNICAL SPECIFICATIONS OF THE MODELS 56
3.5.1. Technical Specifications in Pre-training Stage 56

3.5.2. Technical Specifications in Fine-tuning Stage 57
3.5.3. Additional Technical Information 57
4. RESULTS 58
4.1. DATASET SPECIFICATIONS 58
4.1.1. Original BERT Dataset 58
4.1.2. Modified BERT Models Dataset 61
4.1.2.1. Dataset after Replacing NSP with SOP 61
4.1.2.2. Dataset after Retaining MLM and Removing NSP 63
4.1.2.3. Dataset after Combining MLM with TF-IDF 64
4.2. MODELS RESULTS 68
4.2.1. Results of Models Trained on Weights from BERTurk 69
4.2.2. Results of Models Trained from Scratch 70
4.2.2.1. Results of Replacing NSP with SOP 70
4.2.2.2. Results of Retaining MLM and Removing NSP 70
4.2.2.3. Results of Combining MLM with TF-IDF 71
4.2.3. Results of Models Trained on a Larger Corpus 74
4.2.4. Results of Proposed Model's Comparison with BERTurk Model 75
5. DISCUSSION AND CONCLUSION 77
REFERENCES 85
CURRICULUM VITAE 92

LIST OF FIGURES

Page Number

Figure 2.1: Transformer encoder-decoder architecture in language translation....................10

Figure 2.2: Architecture layout of the transformer (Vaswani et al., 2017). 11

Figure 2.3: Self-attention connections in sentence analysis. 14

Figure 2.4: Illustration of BERT's pretraining and fine-tuning processes. 15

Figure 2.5: Variations in BERT configurations. 17

Figure 2.6: BERT's input representation layers (Devlin et al., 2018). 18

Figure 3.1: Example of raw text in one thesis. 40

Figure 3.2: Example of thesis including two languages: Turkish and German.....................40

Figure 3.3: List of symbols and unknown non-Turkish characters in the dataset. 41

Figure 3.4: Two-step training process for original BERT model and each proposed model. 48

Figure 4.1: Token selection and transformation in MLM dataset creation. 64

Figure 4.2: Top 200 tokens with highest TF-IDF scores..66

LIST OF TABLES

Page Number

Table 2.1: Sample of NSP dataset. 24

Table 2.2: Sample of SOP dataset. 25

Table 3.1: Corpus details for the pre-training stage. 42

Table 3.2: An example of a sentence in BIO format used for NER tasks. 43

Table 3.3: Distribution of label frequencies and descriptions in NER dataset. 44

Table 3.4: An example of multi-label text classification. 45

Table 3.5: Distribution of label frequencies in multi-label text classification dataset. 46

Table 3.6: An illustrative example demonstrating contrasts between the original MLM and our proposed approach. 51

Table 3.7: Proposed approaches for enhancing BERT model performance. 52

Table 3.8: Data distribution for NER and multi-label text classification. 54

Table 3.9: Confusion matrix structure. 55

Table 3.10: Key BERTConfig parameters for pre-training the models. 56

Table 3.11: Fine-tuning configuration for NER and Multi-label text classification. 57

Table 4.1: NSP dataset examples. 59

Table 4.2: MLM dataset examples. 60

Table 4.3: Combined MLM and NSP dataset examples. 60

Table 4.4: SOP dataset examples. 62

Table 4.5: Combined MLM and SOP dataset examples. 62

Table 4.6: Example of MLM dataset (MLM_80_10_10, NSP removed). 63

Table 4.7: Top 30 Tokens with highest TF-IDF scores. 65

Table 4.8: Comparison of randomly chosen tokens in “MLM_80_10_10” and “MLM_80_(10_TF_IDF)_10”. 67

Table 4.9: Comparison of masked sentences in “MLM_80_10_10” and “MLM_80_(10_TF_IDF)_10”. 68

Table 4.10: Fine-tuning results of models pre-trained on BERTurk weights. 69

Table 4.11: Impact of replacing NSP with SOP on model performance...............................70

Table 4.12: Impact of retaining MLM and removing NSP on model performance. 71

Table 4.13: Impact of combining MLM with TF-IDF on model performance. 71

Table 4.14: Summary of fine-tuning results for models pre-trained from scratch. 73

Table 4.15: Impact of corpus size used in pre-training stage on model performance. 74

Table 4.16: Performance comparison between our modified models and BERTurk. 75

Abbreviations	**Explanations**
AI	: Artificial Intelligence
ALBERT	: A Lite BERT
BERT	: Bidirectional Encoder Representations from Transformers
GPT	: Generative Pretrained Transformer
GPU	: Graphics Processing Unit
LM	: Language Modeling
LSTM	: Long Short-Term Memory
MLM	: Masked Language Model
NER	: Named Entity Recognition
NLM	: Neural Language Model
NLP	: Natural Language Processing
NSP	: Next Sentence Prediction
OOV	: Out Of Vocabulary
PLM	: Pretrained Language Model
RNN	: Recurrent Neural Network
RoBERTa	: A Robustly Optimized BERT Pretraining Approach
SL	: Supervised Learning
SLM	: Statistical Language Model
SOP	: Sentence Order Prediction
SSL	: Self-Supervised Learning
TF_IDF	: Term Frequency - Inverse Document Frequency
UL	: Unsupervised Learning
WWM	: Whole Word Masking
YÖKTEZ	: Yükseköğretim Kurulu Tez Merkezi (*Higher Education Board National Thesis Center*)

ABSTRACT

THESIS Ph.D.

IMPROVING THE PERFORMANCE OF NLP TASKS IN LEGAL TECH

Farnaz ZEIDI

İstanbul University

Institute of Science

Department of Informatics

ADVISOR
Assoc. Prof.Dr. Çiğdem EROL

II. ADVISOR
Prof. Dr. Mehmet Fatih AMASYALI

Transformer sinir ağı'nın tanıtımı, kendiliğinden denetimli ön eğitim ve transfer öğrenme gibi tekniklerle birleştirilerek BERT gibi gelişmiş modellere yol açmıştır. BERT'ün etkileyici performansına rağmen, daha fazla geliştirme fırsatları bulunmaktadır. Çalışmamız, öncelikle Türkçe hukuki alanında BERT modelini ön eğitim aşamasında yapılan değişikliklerle geliştirmeye odaklanmaktadır. BERT'in performansını artırmak için üç temel fikri takip ettik: NSP'nin SOP ile değiştirilmesi, NSP'nin kaldırılması ve MLM'nin TF-IDF ile birleştirilmesi. Yenilikçi yaklaşımımızda, %10'unu, Tokenizer'ın kelime dağarcığından rastgele belirteçler yerine yüksek TF-IDF değerlerine sahip olanlarla değiştirmeyi öneriyoruz. Ayrıca, MLM'nin orijinal kuralları (80_10_10) içinde çeşitli maskeleme stratejileri uygulandı. Önerilen modellerimizi ön eğitmek için, 'Yüksek Öğrenim Tez Merkezi'nde bulunan hukukla ilgili tez belgelerinden elde edilen 50 MB'lik etiketlenmemiş Türkçe metinden faydalandık. Ardından, bu özelleştirilmiş modellerimizi, Türk hukukuyla ilgili etiketli verileri kullanarak NER ve çoklu etiketli metin sınıflandırma için ayarladık. Bu özelleştirilmiş modellerimizi orijinal BERT ile karşılaştıran kapsamlı bir performans analizi gerçekleştirdik. Kesinlik, duyarlılık ve F-ölçütü performans metrikleri olarak kullanıldı. Bulgular, NSP'nin SOP ile değiştirilmesinin genellikle BERT modelini geliştirdiğini gösterdi. Ancak, NSP/SOP'nin dışarıda bırakılmasının sonuçları önemli ölçüde etkilediğini, daha belirgin bir etki gösterdiğini belirtti. Önerilen MLM yaklaşımımız ise, orijinal BERT modeline göre hem NER'de (%71.85 kesinlik, %83.47 duyarlılık, %77.23 F-ölçütü) hem de çoklu etiketli metin sınıflandırma görevlerinde (%86.61 kesinlik, %61.83 duyarlılık, %72.15 F-ölçütü) önemli iyileştirmeler gösterdi. Ayrıca, farklı maskeleme stratejileri uygulandıktan sonra, çoklu etiketli metin sınıflandırma görevinde, MLM stratejisine daha fazla rastgele belirteç eklemenin olumlu bir etki gösterdiği görüldü. Özellikle, bu belirteçlerin yüksek TF-IDF puanlarına sahip bir listeden seçildiğinde daha iyi performans gösterilecektir. Ancak, NER görevinde, rastgele belirteçlerin eklenmesi sonuçları olumsuz yönde etkiledi ve en iyi model performansı, bu tür belirteçlerin MLM stratejisinden çıkarıldığında görüldü.

Ocak 2024, 108 pages.

Keywords: BERT modeli iyileştirmeleri, maskeli dil model, sonraki cümle tahmini, cümle sırası tahmini, hukuki Türk alanı.

SUMMARY

Ph.D. THESIS

IMPROVING THE PERFORMANCE OF NLP TASKS IN LEGAL TECH

Farnaz ZEIDI

İstanbul University

Institute of Sciences

Department of Informatics

Supervisor : Assoc. Prof. Dr. Çiğdem EROL

Co-Supervisor : Prof. Dr. Mehmet Fatih AMASYALI

The introduction of the Transformer neural network, coupled with techniques like self-supervised pre-training and transfer learning, has paved the way for advanced models like BERT. Despite BERT's impressive performance, opportunities for further enhancement exist. Our study is primarily dedicated to enhancing the BERT model within the legal Turkish domain through modifications in the pre-training phase. We pursued three primary ideas to boost BERT's performance: replacing NSP with SOP, eliminating NSP, and merging MLM with TF-IDF. In our innovative approach, we propose replacing 10% of the MLM selected tokens with those having high TF-IDF values, rather than using random tokens from the Tokenizer's vocabulary. In addition, diverse masking strategies were applied within the MLM's original rules (80_10_10). To pre-train our proposed models, we utilized a 50 MB legal Turkish corpus (unlabeled text) sourced from legal-related thesis documents available in the Higher Education Board National Thesis Center. Subsequently, these customized models were fine-tuned for NER and multi-label text classification using Turkish legal related labeled data. Following over 60 rounds of training in both pre-training and fine-tuning, we conducted a comprehensive performance analysis, comparing these customized models to the original BERT. Precision, Recall, and F-measure were employed as performance metrics. The findings revealed that replacing NSP with SOP generally enhances the BERT model. However, the exclusion of NSP/SOP significantly impacts the outcomes, indicating a more substantial effect. Our modified MLM approach demonstrated significant improvements in both NER (Precision: 71.85%, Recall: 83.47%, F-measure: 77.23%) and multi-label text classification tasks (Precision: 86.61%, Recall: 61.83%, F-measure: 72.15%) compared to the original BERT model. Moreover, applying different masking strategies showed that presenting more random tokens in the MLM strategy positively influenced multi-label text classification, particularly when selected from a list with high TF-IDF scores. However, in the NER task, adding random tokens negatively impacted the results, indicating optimal model performance when such tokens were excluded from the MLM strategy.

January 2024, 108 pages.

Keywords: BERT model improvements, masked language model, next sentence prediction, sentence order prediction, legal Turkish domain.

1. INTRODUCTION

In today's world, we are surrounded by science and technology. Almost everyone has a computer or smart device. Many companies and organizations have special department for information technology and AI. Technology keeps improving and is used in many areas like industry, education, medicine, and law. These smart systems make things work better, save money, give accurate results, and solve problems faster by handling lots of information. For example, in law, AI helps with tasks like checking contracts and doing legal research, making the work quicker and more accurate. In education, online classes and virtual learning help students learn in interactive ways. In healthcare, AI helps with correct diagnoses and planning treatments, especially in reading medical images. Factories use machines and robots to make production faster and with the same good quality. Self-driving cars and smart traffic control change how we move around, making it easier and safer. AI also helps in suggesting things to watch or read for entertainment, and smart systems help in using energy more efficiently.

Meanwhile, NLP is emerging as a pivotal domain within AI, drawing significant research attention. NLP seeks to optimize language and human-computer interaction, boosting communication between individuals and bridging linguistic divides. NLP applications encompass information retrieval, extraction, question answering, summarization, machine translation, and dialogue systems (Nadkarni et al., 2011). Additionally, NLP occupies a central role in legal technology, exemplified by lawyer robots extracting court decisions, automating legal procedures, and providing counsel (Khan, 2021). In the legal scope, data largely exists in unstructured text format, demanding meticulous analysis and time. The prospect of human error looms large. NLP's integration into the legal sector has the potential to diminish errors, expedite processing, yield precision, and elevate decision-making efficiency.

In about ten years, NLP has changed technology a lot. Even though it is relatively new, NLP has quickly developed different language models, from simple N-grams to more complex neural language structures. These changes are really important for NLP researchers and companies, leading them to focus their research in new directions. At the core of NLP are language models (Jozefowicz et al., 2016), wherein probability distributions for linguistic units like words and sentences are estimated. LMs have changed notably, with prior iterations relying on statistical or rule-based frameworks. However, simpler models such as N-gram struggle with longer texts (Khan, 2021). Neural language models have effectively overcame these problems, surpassing their statistical counterparts in LM tasks. RNNs faced challenges like vanishing gradients, hindering their grasp of intricate dependencies in sequences. The advent of LSTM networks (Graves, 2013) resolved these issues, excelling at comprehending complex sequences. The development of language modeling continued with the introduction of the transformer architecture, which was a significant milestone for advanced language models (Vaswani et al., 2017a). In the realm of natural language processing, a significant breakthrough occurred in 2018 with Google's BERT model, sparking a paradigm shift. This revolutionary model showcased its prowess by achieving remarkable results across 11 NLP tasks, encompassing challenges such as sentiment analysis, text classification, NER, and more (Devlin et al., 2018). BERT introduced a new way of learning context from both directions, which helped overcome the earlier problems in understanding detailed context. BERT operates through two stages: pre-training, where it learns language representations from extensive unlabeled datasets (corpus) in an self-supervised manner, followed by fine-tuning, allowing the pre-trained model to adapt to supervised tasks using limited labeled data (Devlin et al., 2018). Pre-training stage involves two pre-training tasks including the MLM and the NSP. MLM, inspired by Cloze task (Taylor, 1953), entails randomly masking certain tokens in the input and predicting the masked words based on context. This allows for bidirectional learning from text, enabling the model to comprehend words within their preceding and succeeding contexts. NSP focuses on discerning sentence relationships. Given two sentences, A and B, if sentence A precedes sentence B, the task involves predicting whether sentence B genuinely follows A or is a randomly selected sentence from the corpus (Devlin et al., 2018). In fine-tuning stage, the pre-trained BERT model can be readily fine-tuned for various downstream NLP tasks. Fine-tuning is relatively efficient and can range from minutes to hours, depending on available resources. Tasks in the fine-tuning stage include "Single sentence classification", "Sentence pair classification", "Token classification" and "Question answering" (Devlin et al., 2018).

Since BERT's inception in 2018, researchers have struggled to enhance its performance. Some have restructured the pre-training stage. For instance, modified BERT's hyperparameters, achieving superior results. Others have introduced variations like XLNET, combining the capabilities of transformer-xl and BERT, and focusing on improving pre-training via permutation language modeling (Yang et al., 2019). Additionally, specialized adaptations such as BioBERT for biomedical content (Lee et al., 2020), SciBERT for scientific data (Beltagy et al., 2019), and Legal-BERT for legal documents were also developed (Chalkidis et al., 2020) have been developed.

Despite remarkable advances in English-language NLP research, the adaptation of methodologies to other languages presents challenges stemming from language-specific characteristics. Turkish, characterized by its agglutinative nature where morphemes attach to a root word, boasts a complex morphological structure (Oflazer, 2014; Mengusoglu and Deroo, 2001) and, due to its flexible sentence order, poses intricate semantic analysis issues (Acikalin et al., 2020). Moreover, there is a lack of resources for natural language processing in Turkish, which limits the efforts of researchers in the field (Oflazer and Saraçlar, 2018).

1.1. OBJECTIVE OF THE STUDY

The main goal of this thesis is to improve BERT's performance by modifying pre-training tasks within the Turkish legal domain. This involves innovating pre-training tasks and implementing modifications to create an enhanced BERT model. The proposed models aim to support legal professionals by helping identify statements, extract crucial information, and streamline decision-making processes. Ultimately, these enhancements aim to lighten the heavy workloads of legal professionals.

1.2. SCOPE AND IMPORTANCE OF THE STUDY

In comparison to English-based models, Turkish-based models are significantly limited, with most of these models being trained on general text rather than specific domains. The utilization of NLP techniques in the field of Legal Tech is gaining prominence, particularly due to the challenges posed by lengthy legal texts and time-consuming search processes. Applications such as named entity recognition and text classification have the potential to greatly enhance legal professionals' performance by enabling quicker decision-making. Additionally, the rapid growth of transformer-based models like BERT has underscored the importance of refining and optimizing such models—a priority that many contemporary NLP experts are diligently pursuing.

Given these circumstances, focusing on identifying weaknesses in the BERT model and striving to enhance its performance, specifically tailored to legal Turkish text, becomes imperative. This study's scope, with a focus on BERT's structure, revolves around the implementation and evaluation of modified BERT models to extract information from legal Turkish text.

1.3. LIMITATIONS OF THE STUDY

The limitations of this study are summarized in the following points:

1. The proposed model is fine-tuned solely for NER and multi-label text classification NLP downstream tasks. No generalization is performed for other NLP downstream tasks such as question answering and binary label text classification.

2. The proposed model is trained exclusively on Turkish legal text and not on texts in other languages or specific domains. It is recommended to train the developed model on general domains and other languages as well.

3. From a technical standpoint, due to GPU memory limitations, we were constrained to using 50 MB of legal text in the pre_training task. It is advisable to use larger datasets to achieve more robust generalization of results.

4. Additionally, due to the number of experimental models and lack of time, along with GPU memory limitations, we trained all models, both in the pre-training stage and the fine-tuning stage for two downstream tasks (NER and multi-label classification), for just one epoch. It is recommended to train for more epochs to potentially achieve better results.

5. Due to the sensitive nature of the data in this field, the text used for pre-training is extracted from Turkish-language theses, focusing on legal topics from the Higher Education Board National Thesis Center (YÖKTEZ). However, it is recommended to utilize texts from legal professionals' files for a more representative corpus.

1.4. THESIS STRUCTURE

This thesis is organized into four chapters: General Parts, Materials and Methods, Results, Discussion, and Conclusion. In the General Parts chapter, the primary focus is on NLP and transformed base models, which include two learning stages: pre-training and fine-tuning stages. Following this, an overview of efforts aimed at enhancing NLP models will be presented. Subsequently, definitions of specific NLP methods and a variety of approaches will be summarized.

The Materials and Methods chapter introduces the datasets used in both the pre-training and fine-tuning stages. To improve the performance of the BERT model, recommended methods are outlined. Additionally, a deeper understanding of the sub-tasks within the fine-tuning and pre-training stages, along with the integration of new ideas into the pre-training stage, will be provided.

In the Results chapter, the outcomes of the models developed within the proposed approaches are presented. Furthermore, the Discussion section involves evaluating the suggested approaches and conducting a comparative analysis of the original BERT model.

Lastly, the Conclusion chapter synthesizes the key findings and insights drawn from the study.

2. GENERAL PARTS

In this section of our study, we aim to delve into several significant topics, including the NLP in Turkish, a brief history of language models, and the innovation of transformers, particularly BERT models. We will explore the two primary steps in training, encompassing pre-training and fine-tuning. Since TF-IDF plays a pivotal role in our innovative approach, we will extensively discuss it. Toward the end, we will offer a comprehensive overview of the literature, examining studies dedicated to enhancing the BERT model in languages other than Turkish. This includes a focus on both non-legal and legal domains. Following this, we will deeply explore investigations centered on augmenting BERT models specifically for the Turkish language, covering domains in both non-legal and legal contexts.

2.1. NLP IN THE TURKISH LANGUAGE

Turkish is an agglutinative language, akin to Finnish, German, and Hungarian, where morphemes adhere to a root word, somewhat resembling 'beads-on-a-string'. It differs from languages using prefixes and productive compounding (Oflazer, 2014). Turkish has the flexibility to incorporate various inflectional and derivational suffixes. It's typical to construct words that encapsulate an entire English sentence. For example, “oku+y+abil+ecek+se+m” translates to 'if I will be able to read' (Oflazer, 2014). In agglutinative languages like Turkish, word meaning arises from the combination of a word root and morphemes, where the word's meaning alters with the addition of suffixes to the root (Aytan and Şakar, 2023).

The agglutinative structure of Turkish facilitates an extensive vocabulary due to its highly productive derivational processes. Furthermore, morphological ambiguity can result for various reasons even without considering the context (Özer and Korkmaz, 2022):

- Ambiguity in the part of speech of the word root.
- Presence of homograph morphemes with multiple interpretations.
- Multiple segmentation possibilities for word forms arising from a range of root forms and morphographemic processes that generate similar surface structures.

In the realm of NLP tasks, challenges posed by a large vocabulary and linguistic ambiguity are prevalent. To address this issue and mitigate data sparsity, a technique used in language modeling involves the utilization of sublexical units as opposed to complete words (Özer and Korkmaz, 2022). Research in Turkish language studies within the field of NLP is abundant (Oflazer and Saraçlar, 2018; Budur et al., 2020; Çöltekin et al., 2023; Oztürk et al., 2023), with various studies showcasing the emergence of Transformed-based language models as a potential solution to overcome the challenges of large vocabulary and ambiguity (Özer and Korkmaz, 2022; Aytan and Şakar, 2023). These models have shown superior performance compared to previous models like LSTM, mainly due to their attention mechanisms and the ability to represent contextual information. Additionally, these models leverage different Tokenizers, such as syllable Tokenizer, byte-pair encoder Tokenizer, Character Tokenizer, and Wordpiece Tokenizer, tailored for various tasks in NLP.

Therefore, in our study, we are focusing on the BERT model, which serves as a precursor to transformed models that will be elaborated on in the forthcoming sections.

2.2. A BRIEF HISTORY OF LANGUAGE MODELS

From a technical standpoint, language modeling represents a significant approach in enhancing the linguistic capabilities of machines. In essence, LM seeks to create models that can estimate the likelihood of word sequences, enabling them to predict the probabilities of forthcoming or absent tokens (Jozefowicz et al., 2016). Scholarly works have given considerable emphasis to the study of LM, and this research may be divided into four main developmental stages (Zhao et al., 2023):

2.2.1. Statistical Language Models

Statistical language models or SLMs are computational models that employ statistical methods to estimate and predict the probability of word sequences, relying on the Markov assumption (Jelinek, 1998). This assumption means that the prediction of the next word is influenced by the most recent context. One example of an SLM is the N-gram model, where "n" represents the context length. However, statistical or rule-based models like N-grams struggle when dealing with longer texts, as they confront the challenge of the "curse of dimensionality" due to the substantial number of transition probabilities that need to be estimated (Gao and Lin, 2004).

2.2.2. Neural Language Models

Neural language models or NLMs have notably outperformed their statistical counterparts in language modeling tasks. NLMs (Kombrink et al., 2011) are models that evaluate the probability of word sequences using neural networks, encompassing RNNs and the introduction of word2vec (Mikolov et al., 2013), which aimed to create a simplified neural network for obtaining distributed word representations. However, NLMs also encountered challenges, including the problem of vanishing gradients, which hindered their ability to grasp intricate dependencies within sequences (Khan, 2021).

2.2.3. Pretrained Language Models

In an initial endeavor, ELMo (Peters et al., 2018) was presented to achieve word representations that are context-aware. It did so by initially pre-training a bidirectional LSTM (BiLSTM) network. Afterward, in the field of natural language processing, a significant breakthrough occurred in 2018 with Google's BERT model (Devlin et al., 2018), marking a revolutionary turning point. This groundbreaking model demonstrated its excellence by delivering exceptional performance across 11 NLP tasks, including challenges like sentiment analysis, text classification, and named entity recognition. BERT ushered in a new era of contextual representation learning through its bidirectional approach, effectively overcoming previous limitations in capturing nuanced context. The BERT model undergoes two distinct stages for training: first, pretraining on a vast corpus (unlabeled data) to create the pretrained model as output, and then in the second stage, utilizing labeled data to fine-tune the weights of the pretrained model. Models following this approach, similar to BERT, are classified as Pretrained Language Models or PLMs.

After BERT, a multitude of PLMs have been created, including encoder-decoder models like GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2019), along with enhanced pre-training techniques found in modified models such as ALBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019), and ELECTRA (Clark et al., 2020).

2.2.4. Large Language Models

When researchers increase the size of PLMs by either enlarging the model itself or incorporating more training data, it often leads to enhanced performance in tasks downstream

(Kaplan et al., 2020). Numerous studies have examined the performance of increasingly larger PLMs, often referred to as large language models or LLMs, including the extensive 175 billion-parameter GPT-3. Although the idea of scaling primarily pertains to the model's size, using similar architectures and pre-training methods, these LLMs exhibit unique behaviors in contrast to smaller PLMs like 1.5 billion-parameter GPT-2 (Zhao et al., 2023). Moreover, these larger models display unexpected capabilities, often described as "emergent abilities" (Wei et al., 2022), especially in their effectiveness at handling complex tasks.

2.3. FURTHER PRE TRAINING THE BIRTH OF TRANSFORMERS

The capacity of recurrent models, like RNN and LSTM, to capture long-term dependencies is one of their main challenges. In response to this limitation, Vaswani et al. (2017) introduced an innovative architecture named the Transformer, as detailed in their study titled "Attention Is All You Need". The Transformer has emerged as the cutting-edge model for numerous natural language processing tasks, representing a significant advancement in the field and serving as the basis for revolutionary architectures like BERT, RoBERTa, BART, GPT-3, T5, and more (Ravichandiran, 2021).

The fundamental foundation of the Transformer model is constructed entirely upon the attention mechanism, eliminating the necessity for recurrence (Ravichandiran, 2021). It employs a distinctive variant of attention known as self-attention, a concept we will delve into extensively in the forthcoming sections.

2.3.1. Architecture Layout of the Transformer

Some transformed models, like BERT, include an encoder, while others, such as BART and GPT-3, include both an encoder and a decoder. To explain the roles of both the encoder and decoder, the 'language translation task' is the best example in which both the decoder and encoder are used.

To grasp how the Transformer operates in a language translation task, it employs an encoder-decoder architecture. The input sentence, often referred to as the source sentence, is initially input into the encoder. The encoder's role is to learn the representation of the input sentence and subsequently transmit this representation to the decoder. The decoder, equipped with the encoder's learned representation, then produces the output sentence, known as the target sentence (Ravichandiran, 2021). For example, when we aim to translate a sentence from English to Turkish, we feed the English sentence into the encoder. The encoder acquires the representation of the English sentence and conveys it to the decoder. Subsequently, the decoder utilizes the encoder's representation to create the Turkish sentence as the final output (see Figure 2.1).

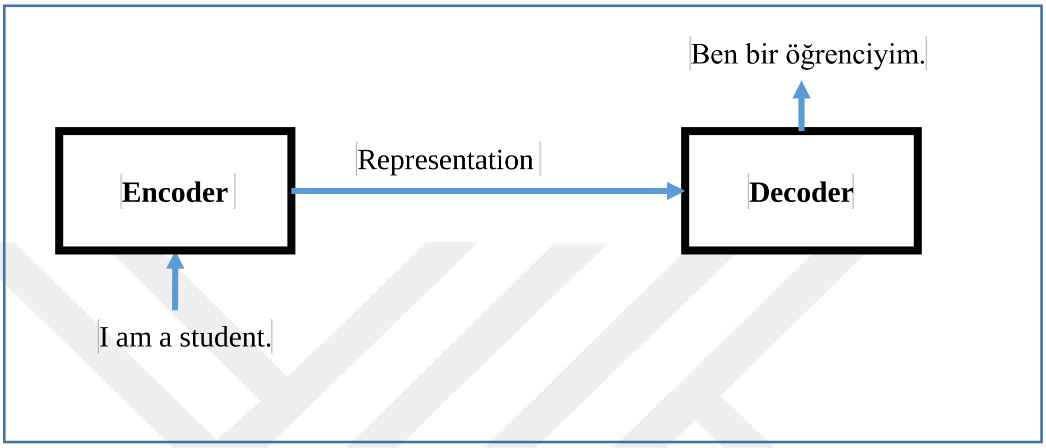

Figure 2.1: Transformer encoder-decoder architecture in language translation.

The architecture of Transformers involves two primary components, the Encoder and Decoder, as shown in Figure 2.2 (Vaswani et al., 2017). It is important to note that the architecture can encompass multiple instances of both Encoders and Decoders.

The Encoder handles the processing and encoding of the input sequence. It takes data, such as a natural language sentence, and transforms it into encoded representations. This segment of the Transformer system analyzes input sequences by performing tasks like tokenization and positional encoding (explained further in the following section) to create embeddings that represent the information contained within the input. It employs self-attention mechanisms to evaluate relationships and dependencies among elements in the input sequence. The self-attention mechanism allows the Encoder to understand the context and importance of each segment in relation to others (further discussed in the next section).

On the other hand, the Decoder is responsible for generating the output sequence based on the information processed by the Encoder. It utilizes the encoded representations from the Encoder, along with previously generated outputs (during training or inference), to predict or generate the subsequent segment of the sequence. Similar to the Encoder, the Decoder also utilizes self-attention mechanisms. Additionally, it employs encoder-decoder attention, enabling it to reference the encoded input sequence to understand and align with the pertinent parts necessary for generating the output sequence. This feature enables the Decoder to consider the context from the input while producing the output.

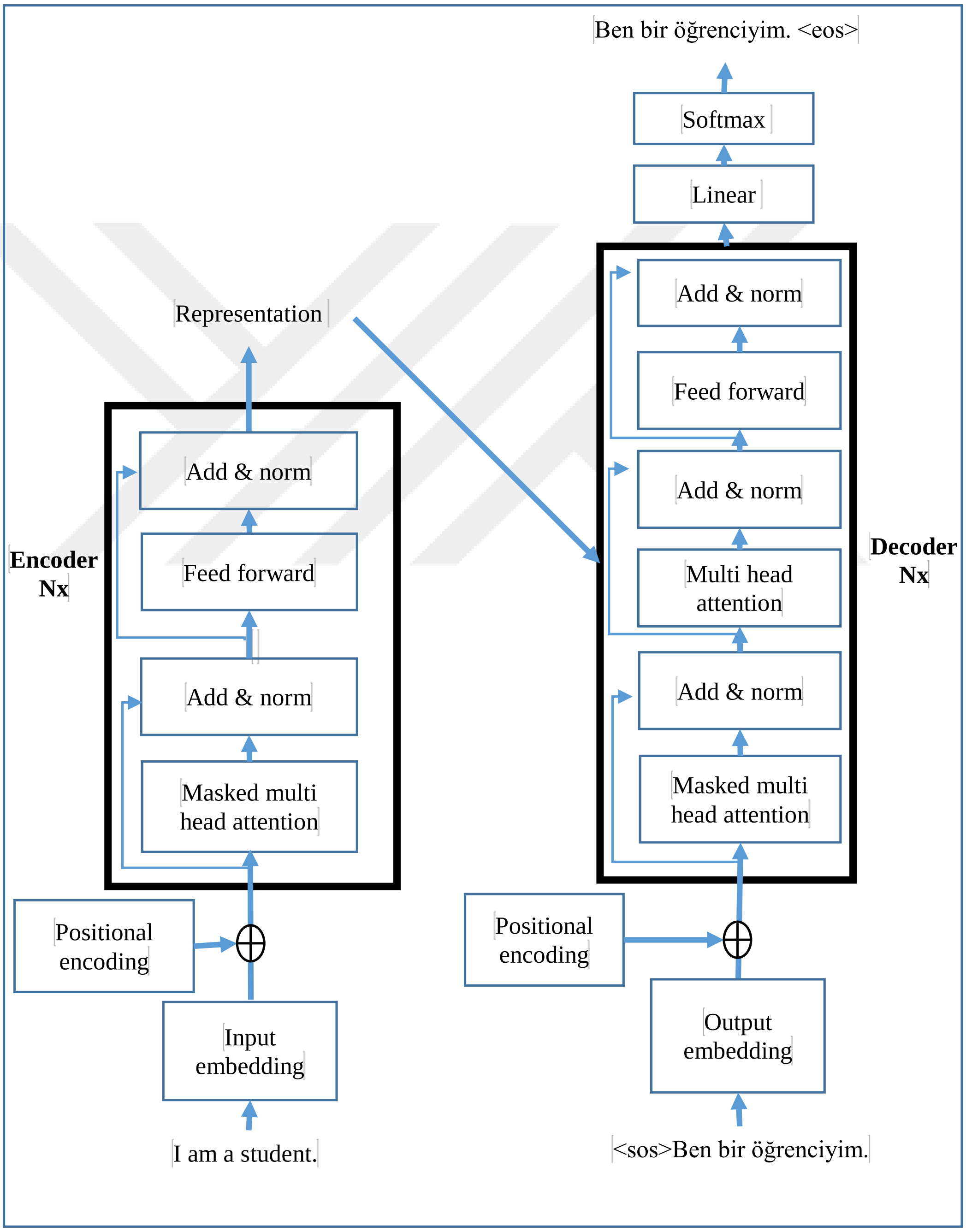

Figure 2.2: Architecture layout of the transformer (Vaswani et al., 2017).

The collaboration between these two components—Encoder and Decoder—is essential for various sequence-to-sequence tasks, such as machine translation, text summarization, and more. The Encoder processes the input sequence, and the resultant encoded information is utilized by the Decoder to generate the output sequence based on the context derived from the input. This architectural design empowers the Transformer model to effectively capture dependencies across sequences and produce outputs while retaining context and relevance between different elements of the input and output sequences.

Moreover, the architecture of the Transformer includes essential components within the Encoder and Decoder. Both the Encoder and Decoder consist of feed-forward neural networks with linear layers and activation functions. These feed-forward networks contribute significantly to the internal computation and transformation of information across tokens in the input and output sequences. The linear layers, with their weighted transformations and biases, play a crucial role in processing and transforming the embedded data. Additionally, the final layer of the Decoder commonly integrates a “Softmax” function, responsible for generating probability distributions for the prediction of subsequent tokens in the output sequence. Simultaneously, the "add and norm" layers, involving residual connections and normalization techniques, aid in maintaining and enhancing the flow of information through the network, facilitating more effective and stable training and inference within the Transformer model. These integrated components collectively contribute to the powerful and complex operations performed within the Encoder and Decoder units of the Transformer architecture (Vaswani et al., 2017).

2.3.2. Positional Encoding

In the context of Transformer models, positional encoding is a method used to inject information about the positions or sequence order of the tokens within a sequence of input data. Traditional neural networks do not inherently account for the order of elements in a sequence. To address this, positional encoding introduces information that helps the model understand the position of tokens within the sequence.

The positional encoding is typically added to the input embeddings. It can take various forms, but a common approach is to create positional embeddings based on sine and cosine functions. These embeddings provide the model with information about the positions of the tokens within the sequence, allowing it to differentiate between tokens based on their position along with their inherent content (Vaswani et al., 2017).

2.3.3. Self Attention

The self-attention mechanism is a fundamental aspect of the Transformer model, enabling the understanding of contextual relationships within input sequences. In self-attention, the model assesses the importance and interdependence of words by evaluating attention scores between different tokens in a sequence. Each word is represented through three vectors: Query, Key, and Value. These vectors, derived from input embeddings, compute attention scores by measuring similarity (using dot product) between Query and Key vectors. The resultant scores guide the weighting of the Value vectors, generating the attention output for each token. This process allows the model to concentrate on relevant parts of the sequence during word processing, facilitating the capture of dependencies and context within the input data (Vaswani et al., 2017).

For a more tangible understanding, consider the sentence: “A cat drank the milk because it was thirsty”. Here, the word 'it' might reference either the “cat” or the “milk”. However, through the model's analysis of context, it's evident that 'it' pertains to the “cat” rather than the “milk”. As the model processes each word in the sentence, it examines relationships among all words to comprehend their meanings. Focusing on the interpretation of 'it,' the model establishes connections with all other words, especially 'cat.' Illustrated in Figure 2.3, these connections (emphasized through a thicker line) highlight 'it' aligning more with “cat” than with “milk”. This showcases how the self-attention mechanism facilitates the model's understanding of contextual inferences within a sentence, determining word references based on their relationships and contextual context.

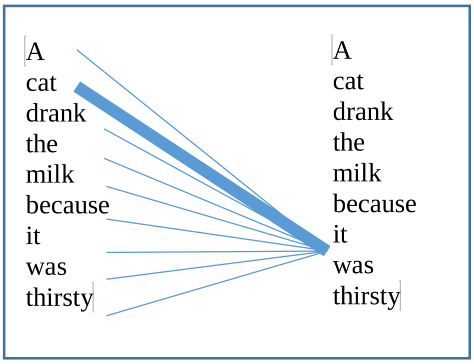

Figure 2.3: Self-attention connections in sentence analysis.

Analyzing self-attention connections among all terms in a sequence is time-consuming, and due to the limited computing memory in BERT, the self-attention mechanism is restricted to handling a maximum of 512 tokens in a single sequence. To address this limitation, some transformer-based models have focused on adapting and enhancing the self-attention mechanism to handle longer sequences, such as BigBird (Zaheer et al., 2020) and Longformer (Beltagy et al., 2020).

BigBird employs block sparse attention instead of the conventional attention used in BERT. By combining global and random attention, it effectively processes sequences up to a length of 4096 while demanding fewer computational resources compared to BERT. Similar to traditional transformers, it utilizes global attention to capture dependencies across a broader context. However, rather than attending to all tokens, it utilizes random attention by segmenting tokens into blocks and attending to specific tokens within these blocks. This approach significantly reduces computational complexity while retaining the capability to capture long-range dependencies (Zaheer et al., 2020).

Longformer introduces an innovative attention mechanism for managing longer sequences. Instead of attending to all tokens in a sequence, Longformer employs a combination of a sliding window mechanism and dilated convolutions to extend the attention span. This strategy enables the model to concentrate on specific tokens within a sequence while excluding others, accommodating longer context sizes (sequences up to length 4,096) without a substantial increase in computational demands (Beltagy et al., 2020).

2.4. BERT MODEL

Google’s BERT (Devlin et al., 2018), or Bidirectional Encoder Representations from Transformers, emerged in 2018 as a breakthrough in natural language processing. Unlike unidirectional learning methods employed by models like GPT, BERT adopted bidirectional transformers. This allowed BERT to understand the context of words by considering both preceding and succeeding words, resulting in a more profound contextual comprehension of language.

2.4.1. Architecture Layout of the BERT

BERT, utilizing the Transformer architecture with an incorporated encoder, harnesses the Transformer's capabilities and attention mechanism for efficient information processing. It employs a dual-phase learning approach: an self-supervised pre-training stage followed by a supervised fine-tuning stage. During the pretraining stage, BERT learns language representations by training on extensive text corpora. It utilizes Transfer Learning to adapt a pretrained model through fine-tuning for specific tasks with labeled data (see Figure 2.4).

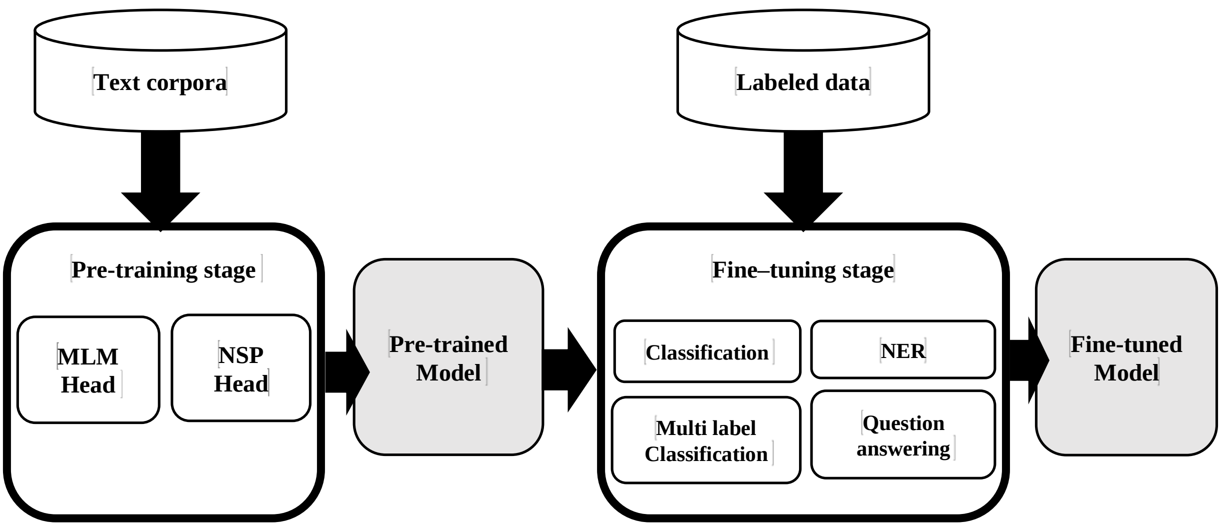

Figure 2.4: Illustration of BERT's pretraining and fine-tuning processes.

In the pretraining stage, BERT focuses on two main objectives: MLM and NSP. MLM involves concealing words within sentences, training the model to predict the missing words based on their contextual surroundings. The NSP objective deals with predicting whether two sentences in the original text are consecutive (Devlin et al., 2018). Further insights into these pretraining objectives will be detailed in the subsequent section for a more comprehensive understanding of BERT's learning process.

In the fine-tuning phase, BERT employs its prelearned representations to address a variety of downstream tasks in natural language processing. These downstream tasks cover specific language applications such as sentiment analysis, named entity recognition, and question-answering systems, which will be further discussed in the following section. Leveraging labeled data customized for each task, the pretrained model adapts through fine-tuning, allowing it to perform proficiently across a diverse array of tasks (Devlin et al., 2018).

2.4.2. Various BERT Configurations

BERT, structured around multiple layers of Transformer Encoders, offers different configurations suitable for various task needs and computational resources (Ravichandiran, 2021):

As the foundation of the BERT series, BERT-base (L=12, H=768, A=12) includes 12 encoder layers (L) with 768 hidden units (H). It contains 12 attention heads (A) and is trained on substantial text corpora, boasting 110 million parameters. This model offers a balanced performance in terms of computational efficiency and accuracy.

The BERT-large (L=24, H=1024, A=16) configuration comprises 24 encoder layers, each containing 1024 hidden units and 16 attention heads. This model demands higher computational resources, housing 340 million parameters. BERT-large is known for its nuanced representations and comprehensive contextual understanding.

Beyond BERT-base and BERT-large, the BERT series includes variants like BERT-tiny (L=2, H=128), BERT-mini (L=4, H=256), BERT-small (L=4, H=512), and BERT-medium (L=8, H=512). These models are adjustments of the base BERT design, varying in their number of encoder layers and hidden units. For instance, BERT-tiny is a compact version designed for low-resource applications, while BERT-small and BERT-medium offer intermediate capabilities between base and large models, balancing performance with computational efficiency.

Figure 2.5 showcases the diverse BERT configurations available:

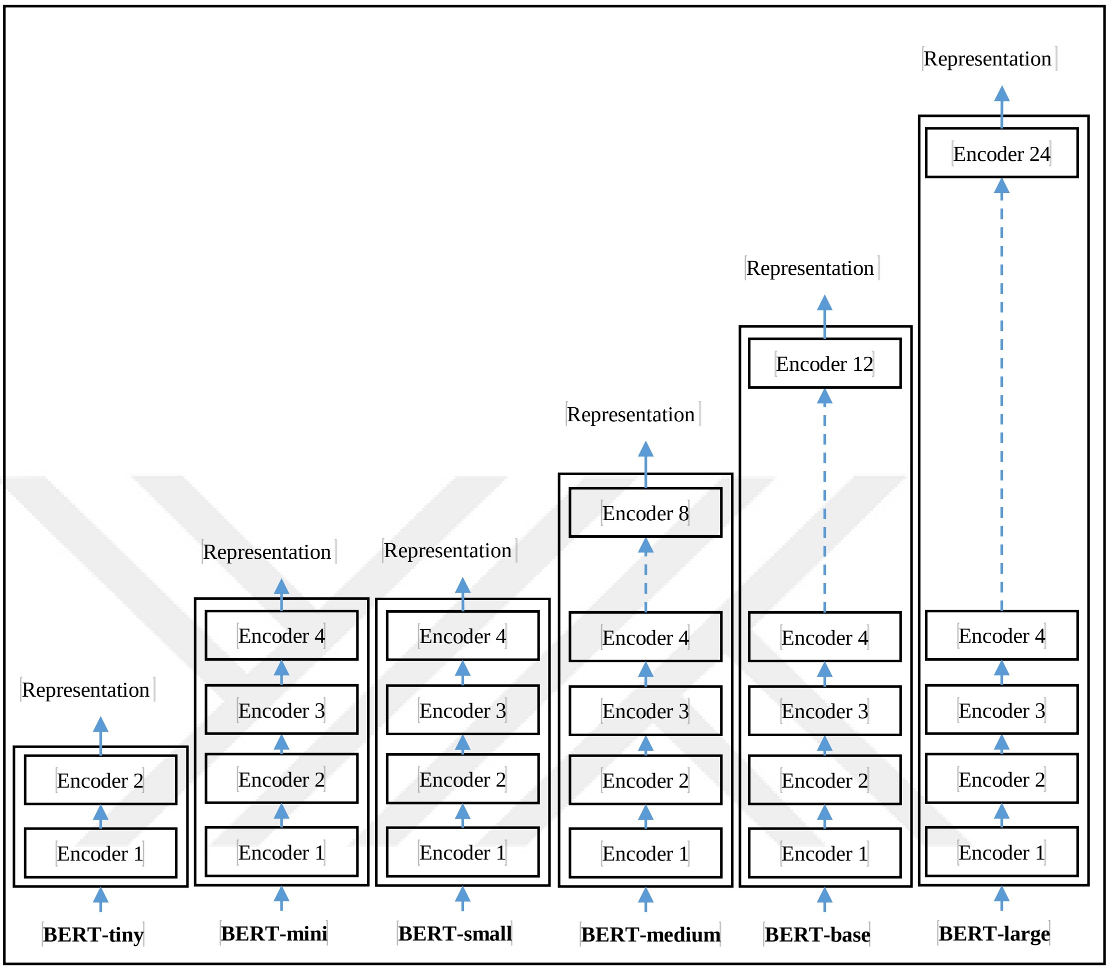

Figure 2.5: Variations in BERT configurations.

Absolutely, the choice of a BERT configuration primarily hinges on the specific needs of a given task and the computational resources accessible. However, BERT-base and BERT-large, as the standard configurations, are commonly preferred due to their balanced accuracy and robust performance across a wide array of natural language processing applications (Ravichandiran, 2021).

2.4.3. BERT's Input Representation Layers

The input representation in BERT consists of three primary embedding layers (Devlin et al., 2018): Token embedding, Segment embedding, and Position embedding. These layers combine to form the final BERT input representation (refer to Figure 2.6).

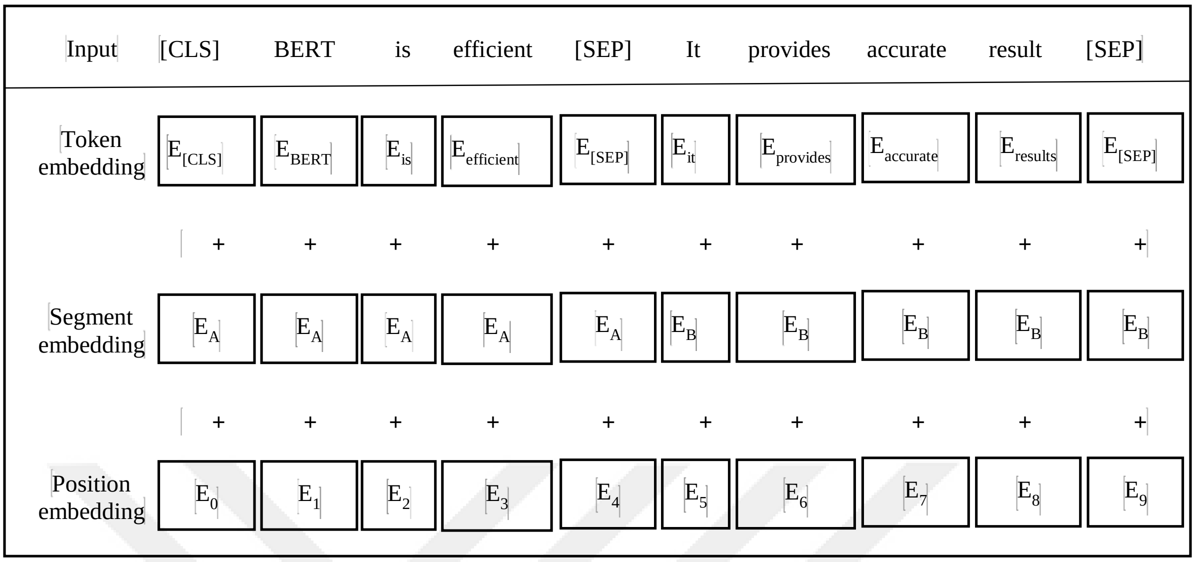

Figure 2.6: BERT's input representation layers (Devlin et al., 2018).

The Token embedding layer processes individual tokens in the input text, creating vector representations for each token. For example, in the sentence “BERT is an amazing model”, each word (token) like “BERT”, “is”, “an”, “amazing”, and “model” is converted into a vector representation.

The Segment embedding layer distinguishes between sentences within the input text. For multiple sentences or segments, this layer assigns distinct embeddings to each sentence. For instance, in sentences such as “BERT is efficient” and “It provides accurate results”, the segment embeddings differentiate these sentences, ensuring BERT understands their separation.

Position embedding aids BERT in understanding the sequential order of tokens. It assigns unique positions to each token, capturing the positional relationships among words in the text. For instance, in the sentence “I enjoy using BERT”, position embeddings allow BERT to identify the specific position of each word.

Furthermore, as depicted in Figure 2.6, the “[CLS]” token, placed at the start of the input sequence, summarizes the entire input, while the “[SEP]” token separates two sentences or segments.

The output from these embedding layers is merged to create the final representation. The combined token embeddings, segment embeddings, and positional embeddings generate a comprehensive representation of the input text, incorporating semantic meaning, sentence separations, and positional information. This final representation is used by BERT for both pretraining and fine-tuning tasks.

2.4.4. BERT's Tokenizer

A Tokenizer is a fundamental component in natural language processing that divides text into smaller units called tokens. Tokens serve as the atomic units or building blocks derived from the text, encompassing words, subwords, or characters, contingent on the design and context of the Tokenizer in use.

In BERT, the WordPiece Tokenizer is employed, segmenting the text into subword units. BERT's vocabulary typically includes 30,000 subword tokens. This vocabulary constitutes a predefined set of subwords that BERT comprehends and processes. When encountering out-of-vocabulary (OOV) words, the Tokenizer dissects them into subword tokens recognizable by the model. The Tokenizer continuously splits and checks the subword with the vocabulary until it reaches individual characters, providing a solution for the OOV problem. In addition to its tokenization function, the Tokenizer can be utilized as an encoder. The encoded output converts these tokens into their respective token IDs, utilized for training or inference (Ravichandiran, 2021).

For instance, consider the Turkish sentence “Bu hüküm bir yanlışlık sonucu sevk edilmiştir.” (translated to English as: “This judgment has been issued as a result of a mistake.”) When tokenized and encoded by the BERT Tokenizer, the output appears as follows:

Original Sentence: “Bu hüküm bir yanlışlık sonucu sevk edilmiştir.”

Tokenized output: ['bu', 'hüküm', 'bir', 'yanlış', '##lık', 'sonucu', 'sevk', 'edilmiştir', '.']

Encoded output: [2, 2123, 3244, 1996, 4190, 2080, 4200, 7306, 6754, 18, 3]

The tokenized output shows that “yanlışlık” is not directly present in the BERT vocabulary. The Tokenizer splits the word into subword units, where “yanlış” and “##lık” are part of the vocabulary. The “##” prefix indicates that the subword follows a previous subword, assisting BERT in comprehending and processing subwords or parts of words absent in its vocabulary.

In the encoded output, the number “2” signifies the [CLS] token at the sentence start, and the number “3” signifies the [SEP] token at the sentence end. The numbers in between correspond to the token IDs for each token/subword within BERT's vocabulary.

2.5. PRE-TRAINING TASKS

In Unsupervised Learning (UL), the primary goal is to derive inherent knowledge from unlabeled data, such as the identification of clusters. Conversely, Supervised Learning (SL) involves training models using labeled data. However, SL encounters challenges attributed to the escalating number of model parameters and the inherent complexity of deep learning models. This expansion necessitates the creation of more extensive labeled datasets, a process that proves to be time-consuming and costly. To confront these challenges, Self-Supervised Learning (SSL) has emerged. SSL combines elements from both supervised and unsupervised learning methodologies. The learning approach of SSL is akin to supervised learning, but the training data labels are automatically generated, known as pseudo labeled data (Qiu et al., 2020).

The innovative concept of SSL and the capability of transfer learning have led to the training of deep learning models in two stages: the self-supervised pretraining step and the supervised finetuning step. During the pretraining step, the model is initially pretrained from scratch, engaging in various "pretext" or "pre-training" tasks aimed at automatically generating pseudo-labeled data from extensive amounts of unlabeled text data (corpora) (Kalyan et al., 2021). This process establishes robust representations. The output of the pre-training step is known as the "pretrained model" with its initial weights.

Subsequently, in the supervised finetuning step, these acquired representations (initial weights) are adjusted (fine-tuned) for other downstream tasks, such as text classification, using labeled data (Ravichandiran, 2021). Recent studies emphasize the significant performance improvements across numerous NLP tasks achieved through the utilization of representations derived from pre-trained Models trained on vast, unannotated corpora. This method has proven to be an effective strategy for leveraging the wealth of information present within extensive, unlabeled text data (Zhou et al., 2023; Qiu et al., 2020).

Various pre-training tasks aim to generate pseudo-labeled data from large, unlabeled corpora. For instance, MLM involves masking a percentage of tokens in each training sample, prompting the model to predict these masked tokens based on contextual cues. Moreover, different pre-training tasks have been employed across various transformed-based models, such as MLM and NSP in BERT (Devlin et al., 2018), dynamic MLM in RoBERTa (Liu et al., 2019), SOP and MLM in ALBERT (Lan et al., 2019).

In addition to the established tasks, different transformer-based models have introduced modified or novel tasks aimed at enhancing model performance. For example, MLM in BERT involves only 15% of tokens in each training sample (Devlin et al., 2018), while tasks like Replaced Token Detection (RTD) in ELECTRA (Clark et al., 2020), Random Token Substitution (RTS) (Di Liello et al., 2021), and Shuffled Token Detection (STD) (Panda et al., 2021) engage all the tokens in the input sample for model learning.

As our project specifically centers around BERT, placing primary emphasis on the MLM and NSP pre-training tasks, a comprehensive discussion of SOP will be presented in Chapter 3. Following that, subsequent sections will delve deeper into MLM, NSP, and SOP, offering in-depth insights into these specific pre-training tasks.

2.5.1. MLM

The MLM serves as a pivotal pre-training task within BERT. Its purpose is to create a comprehensive bidirectional understanding by randomly masking a fraction of input tokens and predicting those that have been masked. The original BERT research (Devlin et al., 2018) dictates that approximately 15% of all WordPiece tokens in each sequence are randomly selected. This approach diverts the model from reconstructing the entire input and encourages its focus on learning specifically from this subset.

The original paper illustrates the MLM strategy using the 80-10-10 Rule, the 15% of selected tokens operate as follows: in 80% of instances, the token is replaced with '[MASK]', in 10% with a randomly chosen word from the Tokenizer's vocabulary, and the remaining 10% retain their original token (Devlin et al., 2018). For example, in the sentence “Bu hüküm bir yanlışlık sonucu sevk edilmiştir", if the word "hüküm" is selected, the model receives the following inputs based on the special part of the MLM strategy:

For 80% of the time masking:

- Model Input: "Bu **[MASK]** bir yanlışlık sonucu sevk edilmiştir."

For 10% of the time replacing with a random token:

- Model Input: "Bu **ağaç** bir yanlışlık sonucu sevk edilmiştir."

For 10% of the time keeping the original selected token:

- Model Input: "Bu **hüküm** bir yanlışlık sonucu sevk edilmiştir."

Moreover, the labeled sentence used for learning by the model remains the original sentence, which is "Bu hüküm bir yanlışlık sonucu sevk edilmiştir.".

Another approach for MLM is Whole Word Masking (WWM) (Ravichandiran, 2021). To understand WWM, let's consider the same sentence, "Bu hüküm bir yanlışlık sonucu sevk edilmiştir", and observe the differences between original MLM and WWM. Since BERT uses the WordPiece Tokenizer, the following tokens are produced:

- Tokens: [[CLS], 'Bu', 'hüküm', 'bir', 'yanlış', '##lık', 'sonucu', 'sevk', 'edilmiştir', '.', [SEP]]

When 15% of tokens are selected, let's suppose 'yanlış' (a part of the word 'yanlışlık') is randomly chosen for masking. In Original MLM, only 'yanlış' is masked, while in WWM, if the selected tokens are part of a word, the entire word, including both 'yanlış' and '##lık', will be masked:

- Original MLM Tokenization: [[CLS], 'Bu', 'hüküm', 'bir', [MASK], '##lık', 'sonucu', 'sevk', 'edilmiştir', '.', [SEP]]
- Whole Word Masking Tokenization: [[CLS], 'Bu', 'hüküm', 'bir', [MASK], [MASK], 'sonucu', 'sevk', 'edilmiştir', '.', [SEP]]

Several transformer-based models have attempted to refine or replace the original MLM, aiming to introduce improved tasks. Models such as RoBERTa (Liu et al., 2019), SpanBERT (Joshi et al., 2020), StructBERT (Wang et al., 2019), and ELECTRA (Clark et al., 2020) have surfaced in this context, and we will delve into them in the literature review section.

2.5.2. NSP

NSP stands apart from the MLM by its emphasis on sentence-level analysis rather than focusing on individual tokens. While MLM plays a pivotal role in tasks such as NER (as elaborated in the following section), NSP directs its attention to learning at the sentence level. This approach is vital for tasks like question answering and text generation, where the objective is to enable a model to grasp the relationships between sentences through a binary classification task (Devlin et al., 2018).

In creating a training dataset for NSP, each example comprises two sentences and their corresponding labels (IsNext or NotNext). The first sentence (A) is randomly chosen from the corpus, such as: “Today the weather is rainy and cold”. Subsequently, for the second sentence (B), 50% of the data are set as positive examples and 50% as negative examples (Ravichandiran, 2021):

In the positive example, the immediate sentence following Sentence A is selected as Sentence B. The combined set of Sentences A and B is then labeled as “IsNext”. For instance:

- Sentence A: “Today the weather is rainy and cold.”
- Sentence B: “I have to take an umbrella.”
- Label: “IsNext”

In the negative example, a different sentence, such as “Water boils at approximately 100 degrees Celsius”, is randomly chosen from a separate document within the corpus as Sentence B. The combination of Sentences A and B is then labeled as “NotNext”. For example:

- Sentence A: “Today the weather is rainy and cold.”
- Sentence B: “Water boils at approximately 100 degrees Celsius.”
- Label: “NotNext”

After sample selection, the model is presented with the following dataset to train the NSP task (Table 2.1):

Table 2.1: Sample of NSP dataset.

Sentences pair	**Label**
['[CLS]', 'Today', 'the', 'weather', 'is', 'rainy', 'and', 'cold', '.', '[SEP]', 'I', 'have', 'to', 'take', 'an', 'umbrella', '.', '[SEP]']	IsNext
['[CLS]', 'Today', 'the', 'weather', 'is', 'rainy', 'and', 'cold', '.', '[SEP]', 'Water', 'b', '##oil', '##s', 'at', 'approximately', '100', 'degrees', 'Ce', '##ls', '##ius', '.' , '[SEP]']	NotNext

This process generates pseudo-labeled data from a vast volume of text, enabling the model to commence learning using these labels.

2.5.3. SOP

As described by the creators of ALBERT (Lan et al., 2019), NSP focuses more on detecting if two sentences belong to the same topic rather than emphasizing grammatical coherence. To address this, a new pre-training task called "Sentence Order Prediction" (SOP) was introduced. SOP is specifically designed to discern the sequential order of sentences within a text. While NSP's primary concern is to determine whether one sentence follows another, SOP aims to predict the accurate sequence of sentences in a given text. This task becomes crucial for understanding long-range dependencies and the coherent structure of a document.

Both NSP and SOP involve a binary classification task, and their methodologies for selecting positive examples in dataset creation are similar. However, the significant difference lies in generating negative examples. In NSP, the second sentence is extracted from another document, whereas in SOP, the positions of the first and second sentences are interchanged to form a negative example (Qiu et al., 2020).

Referring to the previous example with Sentence A: "Today the weather is rainy and cold", in the SOP, both Positive and Negative examples are selected as follows:

Positive example:

- Sentence A: "Today the weather is rainy and cold."
- Sentence B: "I have to take an umbrella."
- Label: "IsNext"

Negative example:

- Sentence A: “I have to take an umbrella.”
- Sentence B: “Today the weather is rainy and cold.”
- Label: “NotNext”

As observed, in the Negative example, only the positions of the two sentences are exchanged, emphasizing inter-sentence coherence. Consequently, SOP is considered capable of solving the NSP task to a reasonable extent, while NSP struggles to resolve the SOP task at all!

Finally, the model is provided with the following dataset for training (Table 2.2).

Table 2.2: Sample of SOP dataset.

Sentences pair	**Label**
['[CLS]', 'Today', 'the', 'weather', 'is', 'rainy', 'and', 'cold', '.', '[SEP]', 'I', 'have', 'to', 'take', 'an', 'umbrella', '.', '[SEP]']	IsNext
['[CLS]', 'I', 'have', 'to', 'take', 'an', 'umbrella', '.', '[SEP]', 'Today', 'the', 'weather', 'is', 'rainy', 'and', 'cold', '.', '[SEP]']	NotNext

Some other transformer-based models also used the SOP instead of NSP, such as StructBERT (Wang et al., 2019) and BERTje (De Vries et al., 2019).

2.6. FINE-TUNING TASKS

Fine-tuning in BERT refers to the process of adapting a pre-trained BERT model to suit a particular task or domain. After training the BERT model on a massive amount of text data to learn language representations, fine-tuning involves adjusting the model parameters by continuing the training on a smaller, task-specific dataset. This fine-tuning process enables BERT to learn task-specific features, making it more proficient in tackling new tasks (Ravichandiran, 2021). Downstream tasks, in the context of fine-tuning BERT, encompass a variety of specific applications or tasks for which the fine-tuned BERT model is designed. These tasks span a wide range, including but not limited to sentiment analysis, NER, text classification, question answering, language translation, and more. By fine-tuning the pre-trained BERT model on labeled data specific to these tasks, the model can effectively make predictions, classifications, or generate task-relevant outputs based on the requirements of the downstream task.

Since in our project, we focus specifically on multi-label classification and NER. The subsequent sections will delve deeper into these two specific downstream tasks.

2.6.1. NER

NER, also known as entity extraction, or identification, is a critical subfield within artificial intelligence. Its main function involves identifying and semantically classifying crucial information, words, or entities within a given text. An entity typically represents words or phrases referring to specific semantic types like persons, locations, organizations, time and more (Yadav and Bethard, 2019; Li et al., 2020). NER is not only fundamental for information extraction (IE) but also plays a pivotal role in various NLP tasks such as information retrieval, question answering, machine translation and text summarization (Li et al., 2020).

The term "Named Entity" (NE) was first introduced at the sixth Message Understanding Conference (MUC-6) (Grishman and Sundheim, 1996) as a task focusing on identifying names of organizations, people, geographic locations, as well as currency, time, and percentage expressions within texts. Since MUC-6, there has been a growing interest in NER, leading to significant focus on this area in scientific events such as ACE (Doddington et al., 2004) and CoNLL03 (Sang and De Meulder, 2003).

In the NER model, two primary steps are involved: 'named entity detection' and 'entity classification'. For example, consider the sentence 'Istanbul is wonderful in spring', which contains five identified entities (tokens): 'Istanbul', 'is', 'wonderful', 'in' and 'spring'. Among these detected tokens, 'Istanbul' is categorized as a 'location', while 'spring' is classified as 'time'.

2.6.2. Multi-label Classification

Machine learning classification involves approximating the mapping function that links input samples to target classes or labels (De Carvalho and Freitas, 2009). In typical classification problems, input samples are associated with a single target label, referred to as single-label classification. Binary classification categorizes data into two classes or categories, predicting outcomes in binary forms such as 'Yes/No', 'True/False', or '0/1'. On the other hand, multi-class classification involves assigning a single class label to each instance within three or more classes or categories. In contrast, multi-label classification extends the concept of multi-class classification, allowing the assignment of multiple labels to an instance or data point. This means that an instance can belong to more than one class simultaneously (Er et al., 2016). Comparatively, multi-label classification demands a more complex implementation of classifiers and can be more time-consuming (Zhang and Zhou, 2007).

For example, in the field of multi-label text classification, consider an article from a news website covering various topics like sports, politics, health, and technology. In a multi-label classification scenario, a specific article could be labeled with 'Health' and 'Technology', signifying that this article can belong to both 'Health' and 'Technology' categories simultaneously.

All types of classification tasks are essential in machine learning and are selected based on the nature of the data and the problem being addressed. Each classification type possesses specific characteristics and applicability across various domains.

2.7. TF-IDF

In this study, we will be incorporating TF-IDF into our modified approach. Therefore, before delving into the literature review, this section will provide a discussion and explanation of TF-IDF.

TF-IDF is an extension of the IDF (Inverse Document Frequency) concept, which was originally introduced by Spärck Jones (2004). TF-IDF is a metric that evaluates the significance of words in one or multiple documents (Chowdhury, 2010). It is effective at reducing the importance of words that are widespread across documents. The core idea behind TF-IDF is that when a word appears frequently in a document but infrequently in other documents, it holds greater importance in distinguishing that document and expressing its core content, thus receiving a higher weight. TF-IDF is a widely employed algorithm for text feature extraction (Zhuohao et al., 2021) and is commonly used for keyword extraction in many research studies. For example, a blend of TF-IDF and the TextRank algorithm is employed to extract keywords from news texts (Yao et al., 2019). Additionally, there are enhanced variations of TF-IDF available for keyword extraction (Sun et al., 2017). Furthermore, numerous studies have used TF-IDF for keyword extraction in their research (Rachman, 2020; Zhuohao et al., 2021; Koloski et al., 2021).

TF-IDF is determined through a combination of two factors: a local one, known as term frequency (TF), and a global one, called IDF (Zhang, 2016). Term frequency, TF(t, d), quantifies the frequency of term t within a specific document, d:

$$\mathrm{TF}(\mathrm{t}, \mathrm{d}) = \text{Count of term t in document d} \;/\; \text{Total terms in document d} \tag{2.1}$$

Inverse document frequency, IDF(t, D), measures the information value of a term t within a collection of documents, D, indicating whether the term is commonly found or rare across that set:

$$\mathrm{IDF}(\mathrm{t.D}) = \log 2\left(\frac{\text{Total documents in D}}{\text{Number of documents in D containing term t}}\right) \tag{2.2}$$

The ultimate TF-IDF score, TFIDF(t, d, D), is calculated by multiplying the term frequency (TF) component, TF(t, d), and the inverse document frequency (IDF) component, IDF(t, D):

$$\mathrm{TFIDF}(\mathrm{t.d.D}) = \mathrm{TF}(\mathrm{t.d}) * \mathrm{IDF}(\mathrm{t.D}) \tag{2.3}$$

2.8. LITERATURE REVIEW

This section provides a comprehensive overview of research pertaining to the BERT model, its enhancements, and its application in various domains and languages. We begin by examining studies aimed at enhancing the BERT model in languages other than Turkish, covering both non-legal and legal domains. Subsequently, we delve into investigations focused on the enhancement of BERT models for the Turkish language, encompassing both non-legal and legal specific domains.

2.8.1. Enhancements to the BERT Model

In this section, we investigate efforts to enhance the performance of the BERT model. We categorize this exploration into two key subdomains: efforts in the Non-Legal Domain and efforts in the Legal Domain.

2.8.1.1. Non-Legal Domain

GPT (Generative Pretrained Transformer) (Radford et al., 2018) and BERT (Devlin et al., 2018) were the pioneering Transformer-based pretrained language models, with GPT based on the

Transformer decoder and BERT on the Transformer encoder (Zhou et al., 2023; Kalyan et al., 2021). Following these, we have seen the development of models like XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020), ALBERT (Lan et al., 2019), T5 (Raffel et al., 2020), BART (Lewis et al., 2019), and PEGASUS (Zhang et al., 2020). Among these, XLNet, RoBERTa, ELECTRA, and ALBERT represent improvements over the BERT model, while T5, BART, and PEGASUS are encoder-decoder models. Given the focus of our study on BERT, an encoder-based model, we will specifically delve into encoder-based models that have improved BERT's performance by modifying subtasks in pre-training, including tasks like MLM and NSP.

RoBERTa (A robustly optimized BERT pre-training approach) (Liu et al., 2019) omitted the NSP task during BERT's pre-training and introduced dynamic MLM in place of the static MLM. Notably, RoBERTa also made significant adjustments to key hyperparameters, including dropout rates, batch sizes, learning rates, and other factors. It is important to note that RoBERTa is pre-trained with longer sequences and a larger corpus compared to BERT. This improvement is achieved by training the model for an extended duration, using larger batch sizes, and training on longer sequences. As a result, RoBERTa demonstrates a substantial enhancement over the originally reported BERT results. SpanBERT (Joshi et al., 2020) also eliminates the NSP task and seeks to reconfigure the MLM task. The authors utilize a creative random process to select token spans instead of individual tokens and then mask them. In addition, they introduce a novel auxiliary objective called the 'span-boundary objective' to predict the complete masked span using the tokens that represent the span's boundaries. Consequently, SpanBERT outperforms Google's BERT in several downstream tasks.

ALBERT (A Lite BERT) (Lan et al., 2019) places its emphasis on modeling inter-sentence coherence and has introduced modifications to the NSP task. The authors argue that NSP primarily evaluates whether two sentences are related in terms of topic, a task relatively simpler than assessing grammatical coherence. Consequently, they have chosen to replace NSP with SOP. It is worth noting that in the ALBERT model, alongside the replacement of NSP with SOP, two parameter-reduction techniques, namely "factorized embedding parameterization" and "cross-layer parameter sharing", have been implemented. These techniques significantly reduce the number of parameters in the model without compromising performance. ALBERT has achieved a score of 86.5, which is 14.5 points higher than BERT. Furthermore, BERTje (A

Dutch BERT Model) (De Vries et al., 2019) and StructBERT (Wang et al., 2019) also utilize SOP in the pre-training phase. In the case of StructBERT, the authors have introduced an innovative structural pre-training approach that incorporates word and sentence structures into the BERT pre-training process. This approach involves the introduction of two new pre-training tasks: a word structural objective and a sentence structural objective. Additionally, StructBERT introduces the Span Order Recovery task to enhance the MLM task. These additions contribute to a more profound comprehension of natural language across various levels of granularity.

XLNet (Yang et al., 2019) represents a generalized auto-regressive pre-training approach that eliminates the NSP task and replaces MLM with a permutation language modeling (PLM). In PLM, it effectively leverages the strengths of both auto-regressive and auto-encoder methods while mitigating their weaknesses. Instead of predicting tokens sequentially, PLM predicts them in a random order. XLNet also incorporates a two-stream attention mechanism and combines BERT's bidirectional capabilities with Transformer-XL's autoregressive technology, resulting in substantial performance improvements compared to BERT and RoBERTa across a variety of downstream tasks.

ELECTRA (Clark et al., 2020) omits the NSP task and places a stronger emphasis on the MLM task. The authors of ELECTRA criticize the MLM task, which involves corrupting the input by replacing randomly selected tokens with [MASK], and then pre-training a model to reconstruct the original tokens. This process demands a significant amount of computational resources. Additionally, the masked tokens often represent only a subset of the input tokens (approximately 15% of the tokens), limiting the model's learning capacity. To address this issue, ELECTRA introduces a new concept with the "replaced token detection" (RTD) approach. The ELECTRA model undergoes pre-training in two steps: in the first step, it trains the generator model, typically a smaller MLM model. Instead of masking the input, ELECTRA corrupts the input by replacing certain tokens with possible alternatives produced from a generator model. In the second step, rather than training a model to predict the original identities of the corrupted tokens, ELECTRA trains a discriminative model to determine whether each token in the corrupted input was replaced by a sample from the generator or not. Consequently, while ELECTRA is pre-trained on the same dataset as XLNet (Qiu et al., 2020), it notably outperforms BERT, RoBERTa, and XLNet while using less than one-fourth of their computational resources.

ConvBERT (Jiang et al., 2020) places its primary focus on self-attention heads. The authors introduce an innovative approach that involves using span-based dynamic convolution to replace these self-attention heads, allowing for the direct modeling of local dependencies. These convolution heads, combined with the existing self-attention heads, together form a new mixed attention block that excels in efficiently capturing both global and local context information. Empirical investigations have demonstrated that ConvBERT outperforms BERT and its variations across a range of downstream tasks. Impressively, the ConvBERT model achieves a score of 86.4, surpassing ELECTRA by 0.7 points and BERT by 3.5 points. There are some other studies that have focused on modifying the self-attention technique, such as Longformer (Beltagy et al., 2020), ETC (Ainslie et al., 2020), BigBird (Zaheer et al., 2020), and Reformer (Kitaev et al., 2020). However, since our focus in this study is on modifying the NSP and MLM tasks, we will not delve deeply into these other studies.

DeBERTa (Decoding-enhanced BERT with disentangled attention) (He et al., 2020) represents an advancement over the BERT model by introducing a unique disentangled attention mechanism and an upgraded masked decoder. This disentangled attention mechanism is responsible for the representation of words using distinct vectors, which capture both their content and positional information. Subsequently, attention weights are computed based on these content and relative positions. In parallel, the enhanced masked decoder is deployed to predict masked tokens, taking the place of the softmax layer during the pre-training phase. It is worth mentioning that the DeBERTa model, which undergoes pre-training on a 78GB corpus, exhibits better performance in comparison to RoBERTa, which is pretrained on a larger 160GB corpus. Besides DeBERTa, there are DeBERTaV2 and DeBERTaV3 (He et al., 2021), which integrate DeBERTa with ELECTRA. These models enhance DeBERTa by integrating ELECTRA's Replaced Token Detection (RTD) mechanism and implementing a novel weight-sharing approach. The DeBERTaV3 Large model stands out with an impressive average score of 91.37%, surpassing DeBERTa by 1.37% and ELECTRA by 1.91%.

There are several other studies aimed at creating faster and smaller versions of the BERT model. For instance, LaMBERT (Light and Multigranular BERT) (Milosheski, 2020) retains the MLM task and utilizes a modified version of SOP. The focus of this research is on reducing training time and achieving faster results compared to models like ELECTRA. Researchers introduced modifications to BERT's input representation and attention mechanism, resulting in a model with a similar parameter complexity to BERT but roughly three times faster in terms of training speed. This speed improvement is achieved by reducing the maximum number of tokens per sentence and enhancing the attention mechanism. LaMBERT maintains a complexity similar to BERT in terms of the number of parameters but achieves faster training, resulting in shorter training times for similar data volumes. Additionally, it incorporates the idea of SOP, as seen in ALBEERT, to predict whether pairs of sentences have been swapped during training. However, in LaMBERT, the training is performed on sentences from the same paragraph or a similar context, rather than using sentences from random paragraphs. Furthermore, other models such as FastBERT (Liu et al., 2020), DistilBERT (Sanh et al., 2019), MobileBERT (Sun et al., 2020), TinyBERT (Jiao et al., 2019), BERT-PKD (Sun et al., 2019), and MiniLM (Wang et al., 2020) also focus on achieving faster training and developing smaller models. Since our focus is on improving BERT's performance rather than reducing training time, we are not paying attention to these types of models.

In the above examples, we summarize studies that attempted to enhance the original BERT by modifying its core structure or subtask manipulation during the pre-training stage. All the mentioned approaches alter the BERT model's structure, resulting in the creation of new models. In the next part, we will examine studies that maintain the BERT structure while aiming to enhance its performance in specific domains or languages.

There are studies focused on pre-training from the original BERT model from scratch for other languages. Examples include Arabic BERT (AraBERT) (Antoun et al., 2020), Turkish BERT (BERTurk) (Schweter, 2020), French BERT (CamemBERT) (Martin et al., 2019), Romanian BERT (RoBERT) (Masala et al., 2020), Persian BERT (ParsBERT) (Farahani et al., 2021), German BERT (Chan et al., 2020), the Russian-language version of BERT, RuBERT (Kuratov and Arkhipov, 2019), SweedishBERT (Malmsten et al., 2020) , Korean BERT (KLUE-BERT) (Park et al., 2021), Hebrew BERT (DictaBERT) (Shmidman et al., 2023) and a Dutch BERT model called BERTje (De Vries et al., 2019). All of these language-specific BERT models have demonstrated superior performance compared to generic multilingual BERT models.

Furthermore, there are BERT models customized for specific domains. For example, in the English biomedical domain, PubMedBERT (Gu et al., 2021) was pre-trained from scratch using PubMed abstracts, outperforming non-specific English BERT models. Similarly, FS-BERT

(Bressem et al., 2020) was pre-trained from scratch using 3.8 million unstructured radiology reports in German, surpassing non-specific German BERT models.

An alternative approach to improving BERT models involves continuous pre-training to customize a language model for a specific task domain (Bayrak and Issifu, 2022), or another language (Eronen et al., 2023). Bayrak and Issifu (2022), in their study on classifying Arabic dialects, extended the pre-training of Arabic pretrained BERT models, including AraBERT (Antoun et al., 2020) and MARBERT (Abdul-Mageed et al., 2020), by incorporating additional dialectal tweets. Their adapted models outperformed generic Arabic pretrained BERT models. In another example, ABioNER (Boudjellal et al., 2021) was further pretrained with an Arabic biomedical corpus. The model's performance was compared with AraBERT and multilingual BERT, and it outperformed both models with an 85% F1 score. There are more examples like RadBERT (Yan et al., 2022), which further pre-trained the BERT-based model to adapt it for the English Radiology Report Corpus, resulting in better performance compared to the BERT-based model.

2.8.1.2. Legal Domain

In the literature, legal-focused research is relatively scarce when compared to other domains, such as the medical field, which we have previously discussed. For instance, in the case of Legal-BERT (Chalkidis et al., 2020), the authors adapted BERT through additional pre-training on domain-specific corpora, and they also pretrained BERT from scratch using English legal domain corpora. Both of these methods outperformed using generic BERT (Devlin et al., 2018). It is important to note that no modifications or structural manipulations were made to the original BERT in this study.

Additionally, in the case of Romanian jurBERT (Masala et al., 2021), which maintains the same structure as the original BERT model, the authors employed a two-step approach. Firstly, they pretrained a BERT model from scratch on a general-purpose collection of legal cases. Then, in the second step, they underwent further training using a subset of the previous corpus. Finally, they fine-tuned the model on labeled data for binary classification tasks. Ultimately, their approach yielded better results than the generic Romanian BERT (RoBERT) (Masala et al., 2020), with a mean AUC of 86.73.

Additionally, in another study (Khan, 2021), the authors proposed additional pre-training of English BERT-Base-Cased on legal data from US court cases, creating Legal-BERT. They also introduced the concept of combining the addition of legal vocabulary, known as Vocab-BERT, which was integrated into the existing WordPiece vocabulary of the BERT model. After fine-tuning on labeled data, this combined approach, Legal-Vocab-BERT, resulted in an enhancement compared to BERT-Base, achieving an accuracy of 95.3% in legal opinions classification tasks. Likewise, in the context of German Legal BERT (Yeung, 2019), the researchers initially pretrained the German BERT model (Chan et al., 2020) using German court decisions. Subsequently, they fine-tuned the language model and integrated domain-specific legal vocabulary, resulting in the development of a dedicated German Legal BERT model. Their conclusions suggested that additional pre-training of a BERT model in the legal domain, especially when the original pre-trained model already contained legal data, resulted in only minor improvements in performance.

Furthermore, in the existing literature, some studies focus on fine-tuning their task-specific labeled data on non-specific BERT models without additional pre-training. For example, in (Chalkidis et al., 2019), a task-specific layer is added on top of English BERT, followed by fine-tuning on task-specific data for multi-label classification tasks. Additionally, in the study titled "German BERT Model for Legal Named Entity Recognition", the authors developed a German legal NER model (Darji et al., 2023). Their method included fine-tuning a non-specific domain German BERT using the Legal Entity Recognition dataset. It is important to note that they did not make any modifications to the pre-training step or adapt the model for the specific domain-related task.

2.8.2. Turkish Language-Based Enhancements to the BERT Model

In general, compared to English-focused studies, there are significantly fewer studies that focus on the Turkish language, especially in the legal-specific domain. After reviewing studies in the Turkish legal-specific domain aimed at improving the results of the BERT model, it is noticeable that most of them concentrate solely on enhancing the BERT model's results through modifications during the fine-tuning step, rather than the pre-training step. There are very few studies that focus on pre-training the original BERT model without any modifications to the MLM and NSP subtasks. In this section, we present a summary of the methods and findings from studies within both the Turkish legal and non-legal domains.

2.8.2.1. Non-Legal Domain

Schweter's BERTurk, monolingual Turkish BERT model, represents a Original BERT model that was trained using a Turkish corpus by the MDZ Digital Library team. This model underwent training on a substantial dataset, encompassing 35GB of text and a total of 44,04,976,662 tokens. This corpus was compiled from a filtered and sentence-segmented version of the Turkish OSCAR corpus, a recent Wikipedia dump, various OPUS corpora, and a specialized corpus contributed by Kemal Oflazer (Schweter, 2020). In addition, the same team has carried out pre-training for DistilBERTurk, ConvBERTurk and Turkish ELECTRA models. It's worth emphasizing that all of these models are pre-trained on the general Turkish corpus and are not specialized for a particular domain.

In another study, Kesgin et al. (2023) extensively researched and developed smaller Turkish BERT models, spanning from tiny to medium. These models underwent testing across various tasks, including mask prediction, sentiment analysis, and news categorization. Their novel approach involved direct training with fewer parameters, emphasizing computational efficiency while maintaining performance. They used BERTurk as their base model for comparison. The key finding of the study revealed a delicate balance between computational efficiency and performance in these smaller models. While the larger BERTurk outperformed them, it demanded significantly more computational resources and time. The authors' goal was to enhance the accessibility of advanced language models like BERT for tasks requiring swift processing and reduced memory usage, catering to real-world applications.

Coban et al. (2023) focused on investigating the domain effects of BERT models fine-tuned for various downstream tasks in Turkish. They pre-trained (MLM)-based BERT models by removing the NSP task from the pre-training stage and trained them from scratch using Facebook messages. Additionally, they selected four different publicly available pre-trained BERT models that were pre-trained on formal Turkish corpora. Subsequently, they fine-tuned the BERT models on various downstream datasets collected from different domains, such as Twitter and Instagram. The study emphasized that performance significantly improved when both the BERT model and downstream task data were selected from the same or closely related domains.

In the literature, there are studies that aim to adapt BERTurk to specific domains through further pre-training on domain-specific corpora, followed by fine-tuning on labeled data, or through direct fine-tuning on BERTurk. For instance, Bayrak et al. (2022) pretrained BERTurk using a more substantial amount of unsupervised medical-specific data to classify Turkish radiology reports, achieving 74.39% precision, 58.66% recall, and 72.21% F1 score. Türkmen et al. (2023a), in their work on classifying Turkish radiology reports, introduced BioBERTurk, which involved additional pre-training of BERTurk on biomedical and radiology-related corpora. Their best model achieved 90.34% precision, 89.98% recall, and 89.97% F1 score. In another study, the authors introduced the BioBERTurk family (Türkmen et al., 2022), featuring three pre-trained Turkish models for biomedicine. These models were initialized from BERTurk, continued general BERTurk pre-training with radiology theses, and combined radiology and biomedicine corpora with the BERTurk corpus for scratch BERT pre-training. They achieved F-scores of 92.99, 92.75, and 89.49 in radiology report classification. These studies underscore the positive impact of further pre-training on small in-domain corpora while revealing the ineffectiveness of pre-training from scratch when combining a smaller in-domain corpus with a much larger general corpus. There are many different BERT models that focus on the Turkish clinical domain and aim to underscore the significance of domain-specific vocabulary during pre-training for enhancing model performance, such as TurkRadBERT (Türkmen et al., 2023b).

Additionally, some studies have focused on multilingual BERT-like language models that provide support for the Turkish language. Notable examples include Multilingual BERT (mBERT), presented by Google AI (Devlin et al., 2018), and XLM-RoBERTa, introduced by Facebook AI (Lample and Conneau, 2019). These language models are employed in a wide range of natural language processing tasks within the Turkish linguistic domain, as demonstrated by Çelik and Dalyan (2023), Köksal and Özgür (2021), and Akyön et al. (2022), showcasing significant results.

2.8.2.2. Legal Domain

To the best of our knowledge, no study involving modifications to the original BERT structure or alterations to subtasks in pre-training (MLM and NSP) for Turkish legal texts has been conducted. Only recently, there has been a limited number of studies in the legal domain, resulting in two models named BERTurk-Legal (Öztürk, 2023) and HukukBERT (Akça, 2023). These models primarily focus on further pre-training BERTurk using legal domain-specific corpora while removing NSP and retaining MLM. In comparison to our study, they do not pretrain the BERT model from scratch and solely concentrate on additional pre-training with MLM-based BERT. Additionally, there is no direct comparison between the original BERT model, which includes NSP and MLM, and the MLM-based BERT in their work. Furthermore, they do not manipulate the original MLM task. Their approach involves augmenting the general corpus with more legal text data to enhance BERT's performance. In contrast, our study uses a fixed corpus, and we aim to enhance the results by manipulating the MLM and NSP tasks.

In the study titled "retrieving Turkish prior legal cases with deep learning", the authors introduce BERTurk-Legal, a model designed for retrieving previous legal cases in the Court of Cassation in Turkey (Öztürk, 2023). This model is created by further pre-training the BERTurk model (which will be discussed in the next section) on the Turkish Legal Corpus, with a primary focus on masked language modeling (MLM), involving the masking of 20% of words during training. The study utilizes BERTurk-Legal to generate document vectors for decision texts within the retrieval dataset. Notably, despite sharing its model architecture with BERTurk, BERTurk-Legal outperforms BERTurk, with an associated f-measure of 38.14%, precision of 52.89%, and recall of 29.8% . It's important to highlight that the model pre-training in this study doesn't start from scratch. Instead, further pre-training is carried out on the BERTurk model. This process includes the removal of the NSP task. Additionally, there's a modification to the token selection for MLM: 20% of tokens are randomly selected, deviating from the original 15% selection. No further adjustments are made to the 80-10-10 rule, where 80% of tokens are masked, 10% are replaced by random tokens from the Tokenizer, and 10% remain unchanged.

In another study (Akça, 2023), the authors employed domain-adapted HukukBERT, which underwent further pre-training on BERTurk. They introduced extensive supervised and unsupervised datasets tailored to the Turkish legal domain, with a specific focus on multi-class and multi-label crime classification for higher-court decisions. Their pre-training approach exclusively used MLM. The study compared their model's performance with traditional machine learning techniques such as Multinomial Naive Bayes, Support Vector Machine, and Logistic Regression, along with deep learning-based classifiers. Notably, their domain-adapted HukukBERT consistently outperformed traditional machine learning models, as well as the base BERT models and DistilBERT models, with an associated f-measure of 87%, precision of

86%, and recall of 88%. This study highlights the effectiveness of domain-adapted BERT models compared to the original BERT, without manipulating the MLM and NSP tasks.

3. MATERIAL AND METHODS

In earlier chapter, a comprehensive discussion has been presented encompassing the problem statement, the foundational framework of BERT, and several related methodologies. Furthermore, an exploration into existing solutions, which have led to substantial modifications in the BERT architecture, has also been undertaken.

In this chapter, our focus is on elucidating the datasets utilized in this research. We delve into the datasets utilized in both the pre-training and fine-tuning stages. Given that the primary aim of this thesis is to conduct a comprehensive analysis of the BERT model and enhance its performance, this chapter introduces innovative and combined approaches to achieve this goal.

3.1. DATASET

As outlined in Chapter 2, the training process in BERT consists of two main stages: pre-training and fine-tuning. Thus, during the pre-training phase, unlabeled Turkish data (corpus) in raw text form plays a crucial role. Conversely, in the fine-tuning phase, labeled Turkish legal data becomes essential, which is provided by NewMind[1].

In general, this section is focused on providing a comprehensive explanation of the datasets utilized in each of these stages.

3.1.1. Corpus in Pre-Training Stage

During this stage, Turkish legal text is required. As the legal text contains sensitive information such as personal data, it is not feasible for NewMind Company to share such data. This sensitive data is not accessible to the public.

To address this need, the preferred approach involves utilizing texts that have been previously published in the legal domain. Consequently, thesis texts related to legal topics from 'YÖKTEZ', are employed. These texts, which are publicly accessible via the YÖKTEZ website[2], are downloaded and utilized as raw legal text for the pre-training stage.

[1] https://www.newmind.ai/en/
[2] https://tez.yok.gov.tr/UlusalTezMerkezi/tarama.jsp

The corpus extracted from YÖKTEZ comprises a total of 14,815 files, amounting to a collective size of 4.35 GB. To exemplify, Figure 3.1 showcases a raw text snippet extracted from one of the theses.

```
çerçevesinin düzenleyen kuralların sadece ve sadece insanların bir arada yaşamak iradesinden
kaynaklanabileceği, siyasal iktidar ve kuralların varlık statüsünün, temel hak ve
hürriyetlerle donatılmış bireyler olarak insanlara (insanlığa değil), bireysel varlıkları
içinde insanlara hizmet etmeye yönelik bir kültürel ürün olduğu aksiyomu, b- Hayatın akışı
içinde toplum kimin ya da kimlerin ve ne yapmak için yöneteceği konusunun, rekabetçi bir
hürriyetler ortamında, sürekli olarak kişilerin karar, tercih ve eylemlerinin toplumsal
sonucu ile belirleneceği aksiyomunun kabulünü gerektirir. ”* 4.Liberal düşünürlerin üzerinde
hassasiyetle durduğu bir diğer önemli nokta da; bu tür argümanların bireylerin hak ve
özgürlüklerini sınırlandırılmasını, buna karşılık devletin gücünün ve etkinliğinin artmasını
sağlayan bir işlev görmesidir. “Ortak refah”, “kamusal iyi” gibi kavramların hepsinin
tarihteki tiranlıkların moral olarak haklılaştırılmasına hizmet ettiğini belirten Rand, bu
konuda şunları yazmaktadır : “Bir toplumun ortak iyisi, o toplum üyelerinin bireysel
iyiliğinden ayrı ve onlara üstün bir şey olarak mütalaa edildiği vakit, bunun anlamı bazı
insanların iyiliğinin, diğerlerinin iyiliği üzerinde bir öncelik kazanması ve ikincilerin
```

Figure 3.1: Example of raw text in one thesis.

Noise is an inherent aspect of any dataset, and this dataset is no exception. After undergoing pre-processing, several instances of noise were identified and subsequently removed from the corpus:

- Certain theses encompass multiple languages, as demonstrated in Figure 3.2 in a specific thesis, the coexistence of two languages—Turkish and German—is evident within a single document.

```
sadece asıl ürün için bir bedel alındığı ve promosyonun karşılıksız olduğu belirtilmektedir.
Ancak, gerçekte satıcının sattığı asıl ürün ve promosyon konusu ek ürünün ekonomik değerleri
toplamı nazara alınarak satış bedeli belirlenmektedir. Günümüzde çok yaygın olan promosyonlu
satımların doğru bir hukuki çerçeveye oturtularak incelenmesi gerekli olup, Tüketicinin
Korunması Hakkındaki Kanunun 11. maddesinde sadece süreli yayın promosyonları için yapılan
düzenlemelere her türlü promosyonlu satımı kapsayacak şekilde bazı eklemeler yapılması
kanımızca yerinde olacaktır. 349 ZUSAMMENFASSUNG Mein Dissertationsthema lautet "Kaufvertrag
mit Zugabe". Die Arbeit besteht aus eine Einfuhrung und drei Kapitel. In Einfuhrung, das
Thema ist vorgestellt und die Bedeutung der Verkaufsförderung ist behandelt. AuBerdem, in
dieser Paragraph ist geschichtlicher Überblick in der Türkei und im Ausland behandelt. In
das erste Kapitel sowohl die Definition, Eigenschaften und Gegenstand der Zugabe als auch
die Definition, Elemente, Rechtliche Qualifikation und Vergleiçhung mit âhnliche Vertrage
des Kaufvertrags mit Promotion und die Arten von Verkaufsförderung behandelt. In das zweite
Kapitel, Abschluss, Parteien und Rechtsfolgen des Kaufvertrags mit Promotion (Zugabe) sind
behandelt. Beim Kaufvertrag mit Zugabe, gibt es drei selbstandige Rechtsverhâltnisse
zwischen Parteien der Vertrag. Die erste Rechtsverhâltnis ist zwischen Hersteller (die
Promotion Durchfuhrende) und dem Verteiler. Die zweite Rechtsverhâltnis ist zwischen der
```

Figure 3.2: Example of thesis including two languages: Turkish and German.

- Some files, characterized by their size of only 1 KB, were examined. It was determined that these texts solely encompassed the first page of the thesis, containing student and supervisor names, as well as university details. Consequently, these instances were excluded from the dataset.
- Noise in the form of symbols and unknown non-Turkish characters, exemplified in Figure 3.3, was detected within some theses. As a result, any theses containing these unknown tokens were removed.

Figure 3.3: List of symbols and unknown non-Turkish characters in the dataset.

Upon completing the data cleaning process, the corpus was reduced to 3.5 GB, comprising 13,094 files. However, due to the extensive time required for pre-training on such a large corpus, a random subset of 500 MB (equivalent to 1,260 files) was selected. As the project progressed and various experiments were conducted (with a minimum of 60 trained models throughout the project), the demand for resources became more pronounced. Considering time constraints and GPU memory limitations (a single pre-trained model on 500 MB took over 12 hours to train), the decision was made to further reduce the corpus volume to 50 MB. In the end, the selected raw legal Turkish text for the pre-training stage encompasses a size of 50 MB and comprises 124 files, totaling 347,147 sentences. Additional corpus details are provided in Table 3.1:

Table 3.1: Corpus details for the pre-training stage.

Attribute	Value
Size of the corpus	50 MB
Number of files in the corpus	124
Total sentences in the corpus	347,147
Total number of tokens	11,543,970
Total number of unique tokens	27,280
Total number of words	6,465,260
Total number of unique words	399,104

It is worth highlighting that the BERTurk model “dbmdz/bert-base-turkish-cased” with a vocabulary size of 32k, accessible through Hugging Face[1], for token tokenization is utilized. Additionally, the process of word tokenization was carried out using the 'tr_floret_web_lg' model, sourced from the spaCy library.

3.1.2. Labeled Data in Fine_Tuning Stage

n the fine-tuning stage, having labeled data is necessary, and this data is provided by NewMind Company. However, due to its sensitive nature, sharing this data publicly is not feasible.

NewMind is a legal technology company based in Istanbul, Turkey. By integrating artificial intelligence, they have transformed the legal profession. Their main objective is to speed up and enhance services, bringing about a fundamental shift in operational methods, cognitive workflows, and collaborative dynamics within the legal sector. To support this research, NewMind has assembled a team of professionals, including legal experts and law students. This collaborative endeavor has produced labeled data that has been meticulously annotated by this skilled team. As a result, two sets of labeled data are utilized to serve two separate fine-tuning objectives: NER and multi-label text classification.

[1]https://huggingface.co/dbmdz/bert-base-turkish-cased

3.1.2.1. Labeled Data in NER Task

When employing machine learning algorithms for NER tasks, the common approach involves converting annotated data into the BIO format. In this format, words are classified into categories using 'B' to indicate the beginning of an entity, 'I' for words inside an entity, and 'O' for words outside any entity. This transformation redefines the NER challenge as a classification task that needs sequential labeling. Each word receives one of three class labels (refer to Table 3.2). Apart from the BIO format, there exists an alternative method of representing tags known as BIESO (Beginning, Intermediate, End, Single word entity, Outside). This approach introduces two additional tags to differentiate between 'end' tokens and 'intermediate' tokens, as well as to distinguish single-word entities from entities that span multiple words (Tang et al., 2013). However, in this study, the BIO format has been chosen.

Table 3.2: An example of a sentence in BIO format used for NER tasks.

Words	**Labels**
Hukuki	O
Bilgi	O
Notu	O
Tarih	O
:	O
3	B-DAT
Nisan	I-DAT
2018	I-DAT
Konu	O
:	O
Kırklareli	B-CRT
Cumhuriyet	I-CRT
Başsavcılığı	I-CRT
tarafından	O
yürütülen	O
2014	O
/	O
1329	O
soruşturma	O
numaralı	O
dosya	O
hakkında	O
.	O

The BIO format labeled data, supplied by NewMind for the NER stage, consists of 614,541 sentences and occupies a total of 305 MB. Additionally, the cumulative count of labeled words reaches 20,220,789. This dataset includes 11 general labels that encompass diverse categories such as 'Person', 'Law', 'Publication', 'Government', and others. The distribution of frequencies for each label, along with their corresponding descriptions, is presented in Table 3.3.

Table 3.3: Distribution of label frequencies and descriptions in NER dataset.

Label	Frequency	Entity Description
B-COR	116143	Beginning of Corporation
B-CRT	12545	Beginning of Court
B-DAT	27363	Beginning of Date
B-GOV	36287	Beginning of Government
B-LAW	91954	Beginning of Law
B-LOC	60577	Beginning of Location
B-MNY	19050	Beginning of Money
B-OOR	6749	Beginning of Other Organization
B-PER	23269	Beginning of Person
B-PRO	2604	Beginning of Project
B-PUB	61507	Beginning of Publication
I-COR	250546	Intermediate of Corporation
I-CRT	14611	Intermediate of Court
I-DAT	99397	Intermediate of Date
I-GOV	74241	Intermediate of Government
I-LAW	700110	Intermediate of Law
I-LOC	1097	Intermediate of Location
I-MNY	71348	Intermediate of Money
I-OOR	14076	Intermediate of Other Organization
I-PER	33550	Intermediate of Person
I-PRO	8254	Intermediate of Project
I-PUB	751274	Intermediate of Publication
O	17744237	Outside any Entity
Total	20220789	

3.1.2.2. Labeled Data in Multi-label Text Classification Task

As previously discussed in Chapter 2, the multi-label text classification task encompasses scenarios where a single sentence or text can be associated with multiple labels or classes. The labeled dataset provided by NewMind for the this task comprises 175,279 sentences, collectively occupying 63 MB of storage. Table 3.4 contains two examples of sentences along with their multilabel assignments. The columns represent the labels, and if the value is equal to "1", it indicates that the sentence has that label; otherwise, "0" indicates that the sentence is not assigned that label. Additionally, due to data sensitivity, certain information such as organization names is represented as "XXX". It is important to note that while there are 31 labels in this dataset, for clarity, only four labels are demonstrated in the examples within Table 3.4.

Table 3.4: An example of multi-label text classification.

Sentence	**Heading**	**Contract Price**	**Taxes**	**Obligation of Notification**
MADDE 3 - FİYAT SİTE YÖNETİMİ , işbu Sözleşme ile , XXX ’ ın SİTE YÖNETİMİ ’ ne vereceği okuma ve faturalandırma işlemi için XXX’ a aşağıda belirtilen bedelde ödeme yapacaktır . [**English version:** “ARTICLE 3 - SITE MANAGEMENT FEE, pursuant to this Agreement, XXX will make a payment to XXX at the below-mentioned rate for the reading and invoicing process that XXX will provide for SITE MANAGEMENT.”]	1	1	0	0
XXX tarafından bireysel ayrıntılı gider paylaşım bildirgesinin (tüm sakinlerin bireysel bildirgesi) e - posta ile SİTE YÖNETİMİ ’ ne gönderilmesi aylık , daire başına 5 . 00 TL + KDV üzerinden hesaplanacaktır. [**English version:** “Sending individual detailed expense sharing statements (individual statements of all residents) to SITE MANAGEMENT via e-mail will be calculated on a monthly basis at 5.00 TL + VAT per apartment.”]	0	1	1	1

As mentioned before, it should be noted that the dataset includes 31 labels, some of which are 'Heading', 'Penalty', 'Transfer_and_assignment', 'Termination', 'Confidentiality', and more. The details of all labels and their corresponding frequencies are presented in Table 3.5.

Table 3.5: Distribution of label frequencies in multi-label text classification dataset.

Label	Frequency
Heading	11015
Penalty	4212
Transfer And Assignment	8104
Termination	7054
Confidentiality	2753
Non Liability Clause	8461
Indemnity	13210
Force Majeure	1328
Change of Control	247
Protectıon of Personal Data	375
Intellectual Property Rights	2156
Waiver	9174
Expenses and Costs	9237
Taxes	4796
Exclusivity	2253
Adaptation Clause	3081
Prohibition of Competition	515
Insurance	1097
Guarantee Clause	4796
Applicable Law	4275
Dispute Resolutıon	2370
Payment	4814
Restriction	2234
Priority of Documents	490
Obligation of Notification	24701
Permit / Approval	15237
Contract Price	6252
Collateral Value	1102
Survival Clause	1238
Renewal	499
Other	77376

3.2. METHODOLOGY

This section introduces innovative ideas aimed at enhancing the performance of the BERT model. It is essential to highlight that, to bolster the efficacy of the BERT model, this project exclusively concentrates on the pre-training stage. Once a pre-trained model is successfully obtained, the subsequent step involves fine-tuning the model on labeled data for both NER and multi-label text classification.

Throughout the project's development, numerous ideas were explored, involving the addition, modification, or removal of subtasks during the pre-training phase. More than 60 models were trained during this period. However, while some models displayed minimal differences in their outcomes, others demonstrated significant improvements. Therefore, this section focuses on the most crucial solutions that were evaluated.

3.2.1. Original BERT

To compare the innovative approach with the original BERT structure, the process starts with pre-training both the original BERT model and the proposed models, followed by fine-tuning and a comparison of their outcomes.

For the original BERT pre-training, depicted in Figure 3.4, tasks include NSP and MLM. After pre-training the original BERT using legal Turkish text extracted from YÖKTEZ, the subsequent step involves fine-tuning two models for NER and multi-label text classification tasks, utilizing specific labeled data for each task, provided by NewMind.

Additionally, as depicted in Figure 3.4, each of the new proposed models undergoes two stages: pre-training the model, then fine-tuning for both NER and multi-label text classification tasks.

Finally, the results obtained from all models will be employed to compare the original BERT model with all the modified BERT models.

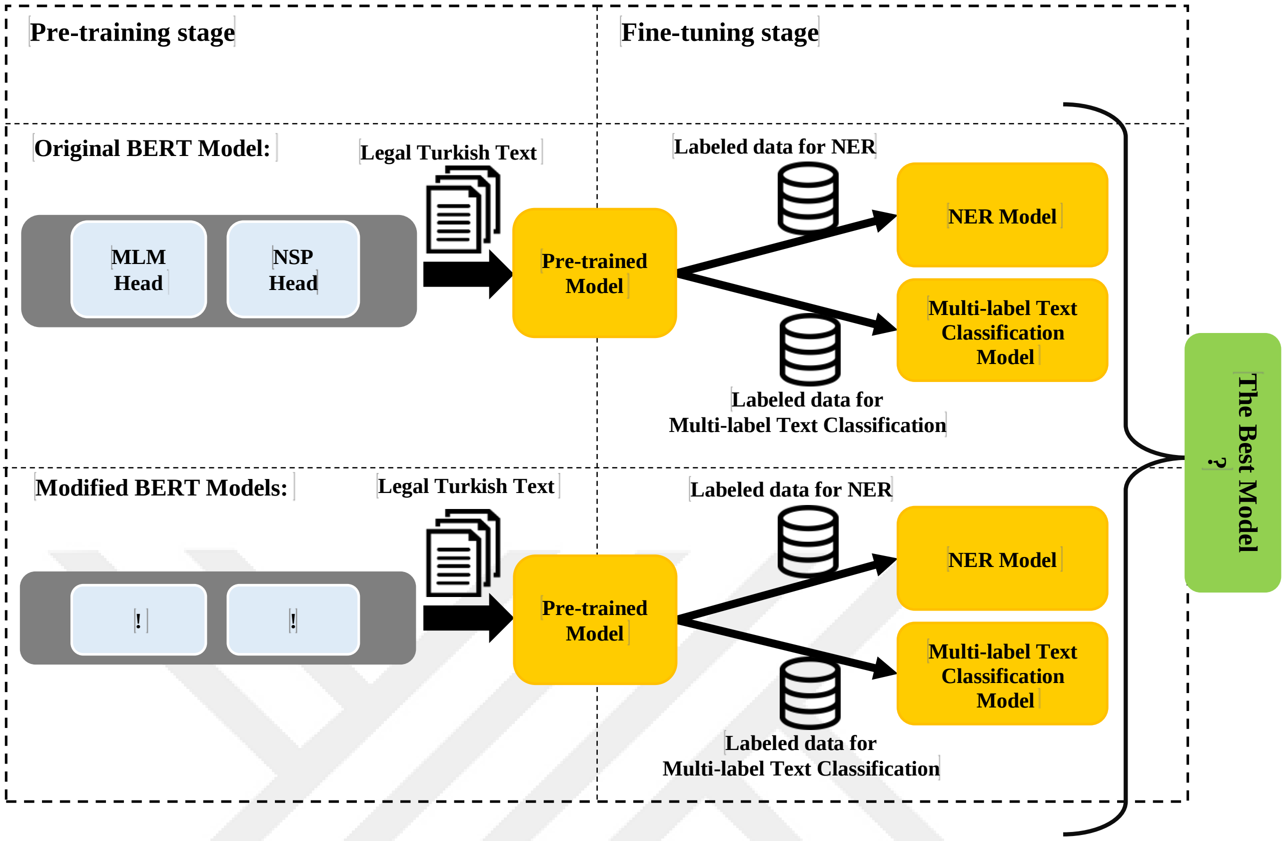

Figure 3.4: Two-step training process for original BERT model and each proposed model.

3.2.2. Modified BERT Models

In Chapter 2, we highlighted the significant efforts undertaken by NLP experts to enhance the BERT model. They diligently addressed specific shortcomings associated with BERT, leading to the emergence of modified models characterized by diverse structures. Within the scope of this project and its time constraints, our primary focus lies in optimizing and manipulating the MLM and NSP components to elevate BERT's performance.

It's essential to highlight that we have preserved the fundamental structure of BERT without any modifications, except for the vocabulary size. Components such as hidden layers, max_position_embeddings, num_attention_heads, hidden_size, and others have been left unaltered.

In this project, the main goal is to execute three essential recommendations ("replacing NSP with SOP," "retaining MLM and removing NSP," and "optimizing MLM with TF-IDF") to improve the performance of the BERT model. Furthermore, various masking strategies were employed within the original rules of MLM (80_10_10), resulting in the formulation of five proposed approaches.

3.2.2.1. Replacing NSP with SOP

As highlighted in Chapter 2, numerous researchers have asserted that the NSP task might not be essential, and altering or removing it could be a preferable approach (Lan et al., 2019; Wang et al., 2019; Liu et al., 2019; Yang et al., 2019; Milosheski, 2020,;Iter et al., 2020; Bai et al., 2020; Clark et al., 2020).

For instance, Google Scholars introduced the ALBERT architecture (Lan et al., 2019), wherein they argue that NSP primarily discerns if two sentences are related in topic—a relatively simpler task compared to assessing grammatical coherence. Consequently, they opted to replace NSP with SOP, a change that effectively addresses the NSP task to a satisfactory extent. In contrast, NSP itself is unable to successfully handle the SOP task in any capacity.

It's noteworthy that in the ALBERT model, alongside the replacement of NSP with SOP, two parameter-reduction techniques, namely "factorized embedding parameterization" and "cross-layer parameter sharing", are employed (Lan et al., 2019). These techniques lead to significant deviations in BERT structures, which have received substantial criticism (Ryu, 2021).

Given this context, this project recommends the replacement of NSP with SOP, while maintaining the unaltered BERT structures.

3.2.2.2. Retaining MLM and Removing NSP

As outlined in Chapter 2, it was emphasized that the exclusion of NSP within the BERT model could adversely affect performance (Devlin et al., 2018). However, recent research has cast doubt on the necessity of the NSP loss (Lample and Conneau, 2019; Joshi et al., 2020).

For instance, RoBERTa excluded the NSP task during BERT's pre-training and introduced dynamic MLM in place of static MLM. Significantly, RoBERTa also made adjustments to key hyperparameters, including dropout rates, batch sizes, learning rates, and other factors (Liu et al., 2019). Additionally, SpanBERT (Joshi et al., 2020) is developed on the foundation of RoBERTa, which adopts the idea of dynamic masking and omits the NSP task.

Considering the potential presence of noise in extracted texts and the prevalence of lengthy sentences in legal texts (often exceeding 512 tokens), it's plausible that NSP or SOP might not yield significant performance improvements in the pre-training task for this database. Therefore, an additional recommendation involves maintaining the unaltered BERT hyperparameters and structure, while excluding NSP and solely utilizing static MLM during pre-training.

3.2.2.3. Combining MLM with TF-IDF

Furthermore, there exist several BERT-like models such as RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), ALBERT (Lan et al., 2019), ELECTRA (Clark et al., 2020), StructBERT (Wang et al., 2019), and DistilBERT (Sanh et al., 2019), all of which aim to enhance the representations derived from BERT by modifying the MLM training approach.

In this project, our objective is to introduce a novel solution for improving the MLM results.

As previously discussed, the MLM method involves randomly selecting 15% of the tokens and then specific strategy, known as "MLM_80_10_10", is applied (Devlin et al., 2018):

1. In the first step, 80% of the selected tokens are masked.

2. In the second step, 10% of the tokens are subject to change and can be replaced with random tokens sourced from the Tokenizer's vocabulary.

3. In the final step, 10% of the tokens remain unchanged.

However, within the context of step 2, we introduce a modification. **Instead of replacing tokens with randomly selected ones from the Tokenizer's vocabulary, we propose substituting them with tokens that demonstrate high TF-IDF values, signifying keywords.** The alteration aims to guide the model to focus on gaining a more profound comprehension of the crucial words, also known as keywords, within the text.

In order to improve understanding, let's consider an example illustrated in Table 3.6. The initial sentence is "Ancak kabul beyanı bu ilişkinin geçerliliğine bağlı olamayan mücerret bir taahhüttür." (English version: ‘However, the acceptance statement is a mere commitment that is not contingent on the validity of this relationship.’) and in the process of random token masking, the second token (corresponding to ‘kabul’(acceptance)) is selected. In accordance with the original approach, in 10% of cases, the token ‘kabul’ might be substituted with a random word from the Tokenizer's vocabulary, such as 'elma' (apple). However, in the proposed approach, this replacement is executed using tokens that have high TF-IDF values within the dataset, like 'talep' (demand).

Table 3.6: An illustrative example demonstrating contrasts between the original MLM and our proposed approach.

Sentence: “Ancak **kabul** beyanı bu ilişkinin geçerliliğine bağlı olamayan mücerret bir taahhüttür.” Selected token: **kabul**		
80% of the time:	**Original method:** Substitute the word with the [MASK] token **MlM output:** Ancak **[MASK]** beyanı bu ilişkinin geçerliliğine bağlı olamayan mücerret bir taahhüttür.	
10% of the time:	**Original method:** substitute the token with a randomly chosen token from the Tokenizer's vocabulary. **MlM output:** Ancak **elma** beyanı bu ilişkinin geçerliliğine bağlı olamayan mücerret bir taahhüttür.	**Our suggested method:** substitute the token with a randomly chosen token from the set of tokens with high TF-IDF scores, indicating their significance as keywords across all documents. **Our modified MlM output:** Ancak **talep** beyanı bu ilişkinin geçerliliğine bağlı olamayan mücerret bir taahhüttür.
10% of the time:	**Original method:** Keep the word unchanged **MlM output:** Original sentence Ancak **kabul** beyanı bu ilişkinin geçerliliğine bağlı olamayan mücerret bir taahhüttür.	

It's essential to note that when calculating TF-IDF scores, we excluded stopwords, punctuation, and numbers from the text.

3.2.2.4. Combined Proposed Models:

Regarding the earlier mentioned three significant recommendations and considering different MLM strategies, Table 3.7 exhibits five combined approaches. This table aims to clarify the differences among these diverse models. For instance, the original BERT model is denoted as 'NSP_MLM_80_10_10', indicating the presence of two subtasks: NSP and MLM_80_10_10.

This designation specifies that 80% of the selected tokens are masked, 10% are replaced with a randomly chosen token from the Tokenizer, and the remaining 10% remain unaltered.

For another example, the designation "MLM_80_(20_TF_IDF)_0" signifies the omission of the NSP component during pre-training. In the context of the MLM task, 80% of the tokens are masked, 20% are replaced with tokens exhibiting high TF-IDF values within the dataset, and 0% remain unaffected.

Table 3.7: Proposed approaches for enhancing BERT model performance.

Approaches	**Pre-training tasks**
1. **Original BERT:** NSP_MLM_80_10_10	NSP Original MLM: • 80% of tokens are masked. • 10% of tokens are replaced with random tokens from the Tokenizer. • 10% of tokens remain unchanged.
2. SOP_MLM_80_10_10	Replacing NSP with SOP Original MLM: • 80% of tokens are masked. • 10% of tokens are replaced with random tokens from the Tokenizer. • 10% of tokens remain unchanged.
3. MLM_80_10_10	Removing NSP Original MLM: • 80% of tokens are masked. • 10% of tokens are replaced with random tokens from the Tokenizer. • 10% of tokens remain unchanged.
4. MLM_80_0_20	Removing NSP Modified MLM: • 80% of tokens are masked. • 20% of tokens remain unchanged.
5. MLM_80_(10_TF_IDF)_10	Removing NSP Modified MLM: • 80% of tokens are masked. • 10% of tokens are replaced with tokens exhibiting high TF-IDF values within the dataset. • 10% of tokens remain unchanged.
6. MLM_80_(20_TF_IDF)_0	Removing NSP Modified MLM: • 80% of tokens are masked. • 20% of tokens are replaced with tokens exhibiting high TF-IDF values within the dataset.

3.3. RESEARCH INQUIRIES

During the course of this project, several central research inquiries have arisen, which will be addressed in Chapter 4:

3.3.1. Pre-Training Strategies: Weights Transfer vs. From Scratch

It is noteworthy that during the pre-training phase, a pivotal consideration emerged: whether to transfer the weights from BERTurk and subsequently train our model on a smaller dataset or to initiate the pre-training process from scratch. This matter will be elaborated upon in greater detail in Chapter 4.

3.3.2. Impact of Corpus Size in Pre-Training

Another significant factor to consider is the size of the corpus. While we have been working with a small 50 MB corpus, we are planning to pre-train our best models using a larger 100 MB corpus. This approach will allow us to assess how the performance of our models is affected by varying corpus sizes.

3.3.3. Comparative Analysis: Proposed Models vs. BERTurk Model

Furthermore, upon obtaining the results of fine-tuning our proposed pre-trained models on a 50 MB legal-related corpus, encompassing 11,543,970 tokens, we will conduct a comprehensive comparative analysis. This analysis will involve a comparison of our results with those obtained from fine-tuning the BERTurk pretrained model, which has been trained on a substantial 35 GB corpus comprising 4,404,976,662 tokens. It is important to note that BERTurk's training corpus is not limited to legal content; it also includes the Turkish OSCAR corpus, a recent Wikipedia dump, various OPUS corpora, and a specialized corpus contributed by Kemal Oflazer (Schweter, 2020).

3.4. MODELS PERFORMANCE EVALUATION

In this section, the evaluation of the proposed pre-trained model's performance is discussed, starting with the evaluation methods and followed by the evaluation criteria.

3.4.1. Evaluation Methods

As depicted in Figure 3.4, each proposed model undergoes pre-training, after which the pre-trained model is subjected to a further training step known as fine-tuning. This fine-tuning process is carried out for two downstream tasks: NER and multi-label text classification.

In this project, NER is selected as a downstream task to fine-tune on a pre-trained BERT model. NER stands for Named Entity Recognition, which is an NLP technique used to identify and classify named entities, such as people's names, organizations, locations, dates, and more, within unstructured text data. In our project, entities are labeled as 'Person', 'Law', 'Publication', 'Government' and others (see Section 3.1.2.1).

Additionally, multi-label text classification serves as another downstream task to fine-tune on a pre-trained model. Multi-label text classification is an NLP technique used to assign multiple labels or categories to a piece of text data, enabling it to be associated with more than one class or category simultaneously. Within our project, the texts undergo categorization into multiple labels, encompassing categories such as 'Heading', 'Penalty', 'Transfer_and_assignment,' 'Termination', 'Confidentiality' and more (refer to Section 3.1.2.2).

To facilitate these tasks, the hold-out method is utilized to partition the datasets into training and test sets. Specifically, 80 percent of the datasets are allocated for fine-tuning purposes, while the remaining 20 percent are reserved for testing. Table 3.8 provides information on the number of sentences in both labeled data (NER and multi-label text classification) during the fine-tuning stage.

Table 3.8: Data distribution for NER and multi-label text classification.

	NER data	**Multi-lable text classification data**
Total sentences	614541	175279
Number of sentences in train set	501661	137955
Number of sentences in test set	112880	37324

3.4.2. Evaluation Criteria

The assessment of fine-tuned model performance relies on the use of a confusion matrix, as presented in Table 3.9. This matrix employs specific abbreviations to communicate essential metrics (Grandini et al., 2020):

- TP (True Positive): Represents correctly predicted positive cases.
- FP (False Positive): Represents incorrectly predicted positive cases.
- TN (True Negative): Represents correctly predicted negative cases.
- FN (False Negative): Represents incorrectly predicted negative cases.

For improved clarity, here are examples for both NER and Multi-label Text Classification:

In the NER task:

- True Positive Example (for 'Person' label): The NER model successfully identifies 'Person' entities like "Farnaz Zeidi" in the sentence: "Farnaz Zeidi avukat olarak tanındı." *(Translates to English as:* "Farnaz Zeidi was recognized as a lawyer."*)*

In the Multi-label Text Classification task:

- True Positive Examples (for 'Heading' and 'Contract Price' labels): The Multi-label Text Classification model correctly identifies 'Heading' and 'Contract Price' in the provided sentence: "MADDE 3 - FİYAT SİTE YÖNETİMİ, işbu Sözleşme ile, XXX' ın SİTE YÖNETİMİ' ne vereceği okuma ve faturalandırma işlemi için XXX' a aşağıda belirtilen bedelde ödeme yapacaktır." *(Translates to English as:* "ARTICLE 3 - SITE MANAGEMENT PRICE: Under this Agreement, XXX will make a payment to XXX in the amount specified below for the reading and invoicing process to be provided to SITE MANAGEMENT by XXX."*)*

Table 3.9: Confusion matrix structure.

		Actual Values	
		Positive	**Negative**
Predicted Values	**Positive**	True positive (TP)	False positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Within the context of the confusion matrix, three fundamental metrics are utilized to assess the effectiveness of the fine-tuned models: Precision, Recall, and F-measure.

Precision: This metric measures the accuracy of positive predictions, quantifying how many of the positive predictions made are correct (true positives).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.1}$$

Recall: Recall assesses how effectively the classifier identifies positive cases among all actual positive cases in the dataset.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.2}$$

F-measure: Calculated as the harmonic mean of precision and recall, the F-measure assigns equal weighting to both metrics, providing a balanced evaluation of model performance.

$$\text{F-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3}$$

3.5. TECHNICAL SPECIFICATIONS OF THE MODELS

There are many details regarding the implementation of the models, but the most important ones are as follows:

3.5.1. Technical Specifications in Pre-training Stage

Given our limited dataset size and the insufficient data available for training the Tokenizer, we made the decision to utilize a BERTurk Tokenizer ('dbmdz/bert-base-turkish-cased'), accessible through Hugging Face[1], with a vocabulary size of 32,000 (Schweter, 2020).

For BertConfig, we selected the bert_base structure with 12 hidden layers, maintaining the default settings of BERTbase except for the vocab_size, which we set to 32,000 to align with the BERTurk Tokenizer. The most important BertConfig parameters are summarized in Table 3.10.

Table 3.10: Key BERTConfig parameters for pre-training the models.

Parameter	**Value**
Vocab_size	32_000
Max_position_embeddings	512
Num_attention_heads	12
Num_hidden_layers	12
Hidden_size	768

Additionally, it's noteworthy that during pre-training, the model's structure differs depending on whether it exclusively employs MLM or combines MLM with SOP or NSP. We selected “BertForMaskedLM” from the Transformers library to train BERT solely on MLM (with all

[1]https://huggingface.co/dbmdz/bert-base-turkish-cased

modifications) and employed “BertForPreTraining” from Transformers to pre-train our models on BERT with MLM and SOP/NSP.

3.5.2. Technical Specifications in Fine-tuning Stage

The fine-tuning process for both NER and multi-label text classification is outlined in Table 3.11. In both of these methods, the models are fine-tuned for a single epoch, utilizing a train_batch_size of 128. This configuration expedites the training process. Additionally, GPU resources are leveraged for both tasks to accelerate the training.

Table 3.11: Fine-tuning configuration for NER and Multi-label text classification.

Parameter	Value/Setting
Train_batch_size	128
Train_epochs	1
Evaluate_during_training	True
Use_cuda	True
Optimizer	AdamW
Learning_rate	4e-5
Scheduler	Linear_schedule_with_warmup

3.5.3. Additional Technical Information

Here are additional details regarding the tools and resources employed:

Python Version: All training and methods were executed using Python version 3.10, utilizing libraries such as NLTK, Pandas, and Torch.

GPU Utilization: The NVIDIA Quadro RTX 8000 model (VCQRTX8000-PB, 48GB GDDR6 ECC, Professional 3D) was used for training language models.

Key Libraries: The primary libraries used in Python include:

- **Transformers**: Utilized for pre-training tasks, including BertTokenizer, BertForPreTraining (for pre-training on NSP/SOP and MLM), and BertForMaskedLM (for pre-training exclusively on MLM).
- **SimpleTransformers**: Employed for fine-tuning tasks, including (MultiLabelClassificationModel, MultiLabelClassificationArgs) for multi-label classification and (NERModel, NERArgs) for NER tasks, respectively.

4. RESULTS

After an extensive discussion of the Turkish corpus and labeled data used in both the pre-training and fine-tuning steps, along with a thorough review of the methodology outlined in the previous chapter, the current chapter provides an overview of the results.

To commence, we will provide detailed specifications regarding the dataset employed for the original BERT model, as well as for the various modified BERT models. Subsequently, we will conduct an in-depth analysis of the performance metrics of all pre-trained models, encompassing not only the original BERT model but also five distinct modified BERT models. Furthermore, we will delve into the outcomes derived from models trained from scratch and those fine-tuned using weights from BERTurk. Additionally, we will showcase the results obtained from models trained on an expanded corpus. Finally, we will engage in a comparative assessment of the modified models in relation to the BERTurk model.

4.1. DATASET SPECIFICATIONS

As mentioned in Chapter 2, during the pre-training stage of the BERT model, a corpus of unlabeled data is utilized. However, for the model to effectively learn from unlabeled data during this stage, this corpus is transformed into a code_base_labeled dataset. In this section, we will present the specifications of the created datasets for the pre-training task for both the original BERT model and other proposed models.

4.1.1. Original BERT Dataset

In the original BERT model, two pre-training subtasks are used: NSP and MLM. In the NSP task, covered in Chapter 2, the dataset comprises sentence pairs. Positive examples demonstrate a logical connection between sentences, while negative examples lack coherence. Both are vital for training the model to recognize meaningful relationships.

Table 4.1 provides NSP dataset examples. In the first line, which represents a negative example, we have Sentence A: 'İkinci ihtimale göre ise, bu hüküm bir yanlışlık sonucu sevkedilmiştir.', and Sentence B: 'Fakat Adalet Komisyonu, tatıl günlerinde bu muamelenin yapılmasını uygun görmediğinden teklifi değiştirmiştir.'. Sentence B not logically following Sentence A is labeled 'NotNext'. For a positive example, consider two sentences: 'İkinci ihtimale göre ise, bu hüküm bir yanlışlık sonucu sevk edilmiştir.', and 'Uygulaması düşünülmeksizin acele ile konulmuş bir hüküm olsa gerektir.'. Here, the second sentence logically follows the first and is labeled as 'IsNext.'"

Table 4.1: NSP dataset examples.

Example Type	Sentence A	Sentence B	Next_sentence_label
Negative example	İkinci ihtimale göre ise, bu hüküm bir yanlışlık sonucu sevkedilmiştir [**English version**: "According to the second possibility, this provision has been dispatched due to an error."]	Fakat Adalet Komisyonu, tatıl günlerinde bu muamelenin yapılmasını uygun görmediğinden teklifi değiştirmiştir. [**English version**: "However, the Justice Commission has changed the proposal as it deemed inappropriate to conduct this transaction on holidays."]	NotNext
Positive example	İkinci ihtimale göre ise, bu hüküm bir yanlışlık sonucu sevkedilmiştir.	Uygulaması düşünülmeksizin acele ile konulmuş bir hüküm olsa gerektir. [**English version**: "It seems to be a provision put hastily without considering its implementation."]	IsNext

In the MLM dataset, as explained in Chapter 2, approximately 15% of tokens are randomly masked and replaced with '[MASK]'. Each sentence is paired with a label indicating the masked tokens. Training pairs comprise the original sentence and its masked version. Negative sampling, as discussed in section 4.1.2.2, involves randomly selecting incorrect tokens from the vocabulary as training distractors. For example, in Table 4.2, the sentence "İkinci ihtimale göre ise, bu hüküm bir yanlışlık sonucu sevkedilmiştir" is masked as follows: "İkinci ihtimale göre ise, bu hüküm bir yanlışlık [MASK] sevkedilmiştir."

Table 4.2: MLM dataset examples.

Sentence	**labels**
İkinci ihtimale göre ise, bu hüküm bir yanlışlık [MASK] sevkedilmiştir	İkinci ihtimale göre ise, bu hüküm bir yanlışlık sonucu sevkedilmiştir
Fakat Adalet [MASK], tatıl günlerinde bu muamelenin yapılmasını uygun görmediğinden [MASK] değiştirmiştir.	Fakat Adalet Komisyonu, tatıl günlerinde bu muamelenin yapılmasını uygun görmediğinden teklifi değiştirmiştir.

For training on both NSP and MLM, the combined dataset is presented in Table 4.3.

Table 4.3: Combined MLM and NSP dataset examples.

Example Type	**MLM Masked Sentence**	**Next sentence label**	**MlM lable (original sentence)**
Negative example	“'[CLS]', 'İkinci', 'ihtimale', 'göre', 'ise', ',', 'bu', 'hüküm', 'bir', 'yanlışlık', **'[MASK]'**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', 'Fakat', 'Adalet', **'[MASK]'**, ',', 'tatil', 'günlerinde', 'bu', 'muamelenin', 'yapılmasını', 'uygun', 'görmediğinden', **'[MASK]'**, 'değiştir', '##miştir', '.', '[SEP]', '[PAD]',… , '[PAD]' ”	NotNext	“'[CLS]', 'İkinci', 'ihtimale', 'göre', 'ise', ',', 'bu', 'hüküm', 'bir', 'yanlışlık', **'sonucu'**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', 'Fakat', 'Adalet', **'Komisyonu'**, ',', 'tatil', 'günlerinde', 'bu', 'muamelenin', 'yapılmasını', 'uygun', 'görmediğinden', **'teklifi'**, 'değiştir', '##miştir', '.', '[SEP]', '[PAD]',… , '[PAD]' ”
Positive example	“'[CLS]', 'İkinci', 'ihtimale', 'göre', 'ise', ',', 'bu', **'[MASK]'**, 'bir', 'yanlışlık', 'sonucu', 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', 'Uygulaması', 'düşünül', '##meksizin', 'acele', 'ile', 'konul', '##muş', 'bir', **'[MASK]'**, 'olsa', 'gerektir', '.', '[SEP]', '[PAD]',… , '[PAD]' ”	IsNext	“'[CLS]', 'İkinci', 'ihtimale', 'göre', 'ise', ',', 'bu', **'hüküm'**, 'bir', 'yanlışlık', 'sonucu', 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', 'Uygulaması', 'düşünül', '##meksizin', 'acele', 'ile', 'konul', '##muş', 'bir', **'hüküm'**, 'olsa', 'gerektir', '.', '[SEP]', '[PAD]',… , '[PAD]' ”

These examples demonstrate how the MLM and NSP tasks can be combined in the dataset for training a model to understand sentence relationships and predict masked tokens. The "MLM Masked Sentence" displays the masked sentence created for the MLM task, where specific tokens are masked with '[MASK]' for prediction. The "NSP Label" shows whether the sentences are considered logically connected (IsNext) or not (NotNext), while the "MLM Label" presents the original sentence without masked tokens, allowing the model to learn from the original tokens.

Additionally, in this representation, '[CLS]' signifies the classification task and is placed at the start of the sentence. '[SEP]' serves as a separator token, indicating the end of the first sentence and the beginning of the second sentence. '[PAD]' is used for padding to ensure sequences have uniform lengths for processing.

4.1.2. Modified BERT Models Dataset

In this section, we will discuss the dataset used for pre-training Modified BERT Models.

4.1.2.1. Dataset after Replacing NSP with SOP

Both NSP and SOP contribute to training models to understand and represent sentence relationships, but they emphasize different aspects of this understanding. In NSP, the model is trained to predict whether the second sentence (Sentence B) logically follows the first sentence (Sentence A). In contrast, SOP focuses on training the model to predict whether the sentences are in the correct order within the pair.

Regarding the Training Data, in NSP, the training dataset consists of sentence pairs where some pairs are logically connected (IsNext), and others are not (NotNext). In SOP, the training data comprises sentence pairs with intentionally shuffled sentence order. In Table 4.4, there are examples of both positive and negative instances. It's evident that in the positive example, Sentence B follows Sentence A in the correct order, labeled as "IsNext." In the negative example, only the order of the sentences is switched, leading to the label "NotNext.".

Table 4.4: SOP dataset examples.

Example Type	Sentence A	Sentence B	Order Label
Positive example	İkinci ihtimale göre ise, bu hüküm bir yanlışlık sonucu sevkedilmiştir	Uygulaması düşünülmeksizin acele ile konulmuş bir hüküm olsa gerektir..	IsNext
Negative example	Uygulaması düşünülmeksizin acele ile konulmuş bir hüküm olsa gerektir.	İkinci ihtimale göre ise, bu hüküm bir yanlışlık sonucu sevkedilmiştir	NotNext

In the end, with the addition of the dataset from the MLM task, Table 4.5 displays the combined dataset for training on both SOP and MLM.

It's important to mention that in MLM, the masked tokens are selected randomly, and there is no requirement for the same masked tokens to be present in both positive and negative examples.

Table 4.5: Combined MLM and SOP dataset examples.

Example Type	MLM Masked Sentence	Order Label	MlM lable (original sentence)
Positive example	“'[CLS]', 'İkinci', 'ihtimale', 'göre', 'ise', ',', 'bu', 'hüküm', 'bir', 'yanlışlık', **'[MASK]'**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', 'Uygulaması', 'düşünül', '##meksizin', 'acele', 'ile', 'konul', '##muş', 'bir', **'[MASK]'**, 'olsa', 'gerektir', '.', '[SEP]', '[PAD]',… , '[PAD]' ”	IsNext	“'[CLS]', 'İkinci', 'ihtimale', 'göre', 'ise', ',', 'bu', 'hüküm', 'bir', 'yanlışlık', **'sonucu'**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', 'Uygulaması', 'düşünül', '##meksizin', 'acele', 'ile', 'konul', '##muş', 'bir', **'hüküm'**, 'olsa', 'gerektir', '.', '[SEP]', '[PAD]',… , '[PAD]' ”
Negative example	“'[CLS]', 'Uygulaması', 'düşünül', '##meksizin', 'acele', 'ile', 'konul', '##muş', **'[MASK]'**, 'hüküm', 'olsa', 'gerektir', '.', '[SEP]', 'İkinci', **'[MASK]'**, 'göre', 'ise', ',', 'bu', 'hüküm', 'bir', 'yanlışlık', **'[MASK] '**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', '[PAD]',… , '[PAD]' ”	NotNext	“'[CLS]', 'Uygulaması', 'düşünül', '##meksizin', 'acele', 'ile', 'konul', '##muş', **'bir'**, 'hüküm', 'olsa', 'gerektir', '.', '[SEP]', 'İkinci', **'ihtimale'**, 'göre', 'ise', ',', 'bu', 'hüküm', 'bir', 'yanlışlık', **'sonucu'**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', '[PAD]',… , '[PAD]' ”

4.1.2.2. Dataset after Retaining MLM and Removing NSP

The MLM dataset is generated by randomly masking approximately 15% of the tokens within a sentence. Out of these 15% selected tokens, 80% are masked, 10% are replaced with random tokens from the Tokenizer to serve as distractors, and the remaining 10% of the tokens remain unchanged (referred to as MLM_80_10_10).

For each sentence, two versions are created:

1. The original sentence, which serves as the label.

2. The masked sentence with '[MASK]' tokens and randomly selected tokens from the Tokenizer (used as negative examples to serve as distractors).

These pairs are employed for training the model. As illustrated in Table 4.6, there is an example that demonstrates the pretrained dataset when only MLM_80_10_10 is used, and NSP is removed. Let us consider the sentence: "İkinci ihtimale göre ise, bu hüküm bir yanlışlık sonucu sevkedilmiştir." Out of this sentence, four tokens are randomly selected: 'İkinci', 'göre', 'hüküm', and 'sonucu'. In this instance:

- 'göre' and 'hüküm' will be masked.

- 'İkinci' remains unchanged.

- 'sonucu' is replaced with a randomly selected token from the Tokenizer **(in this example: "sokak")**.

Table 4.6: Example of MLM dataset (MLM_80_10_10, NSP removed).

MLM Masked Sentence	MlM lable (original sentence)
“'[CLS]', **'İkinci'**, 'ihtimale', **'[MASK]'**, 'ise', ',', 'bu', **'[MASK]'**, 'bir', 'yanlışlık', **'sokak'**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', '[PAD]',… , '[PAD]' ”	“'[CLS]', **'İkinci'**, 'ihtimale', **'göre'**, 'ise', ',', 'bu', **'hüküm'**, 'bir', 'yanlışlık', **'sonucu'**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', '[PAD]',… , '[PAD]' ”

In Table 4.6, it is essential to know that the column for "Next Sentence label" is missing because the NSP is not included in this specific pre-training dataset. In addition, it is important to mention that only one sentence is used, and the '[SEP]' token is employed just once to indicate the "end" of the sentence.

4.1.2.3. Dataset after Combining MLM with TF-IDF

As mentioned in Chapter 3, we want to modify the MLM task. Instead of substituting tokens with randomly chosen ones from the Tokenizer's vocabulary list (MLM_80_10_10), we propose replacing them with tokens that have high TF-IDF values (MLM_80_(10_TF_IDF)_10).

During the implementation of the code and the creation of the dataset for both “MLM_80_10_10” and “MLM_80_(10_TF_IDF)_10”, Figure 4.1 illustrates that 1,720,575 tokens were selected, which represents 15% of all tokens (11,543,970) in the MLM process. Out of this 15%, 80% of the tokens (1,376,460 tokens) were masked, 10% of them (172,057 tokens) were replaced by random tokens from the **Tokenizer** or **Tokens with high TF-IDF scores**, and the remaining 10% (172,058 tokens) remained unchanged.

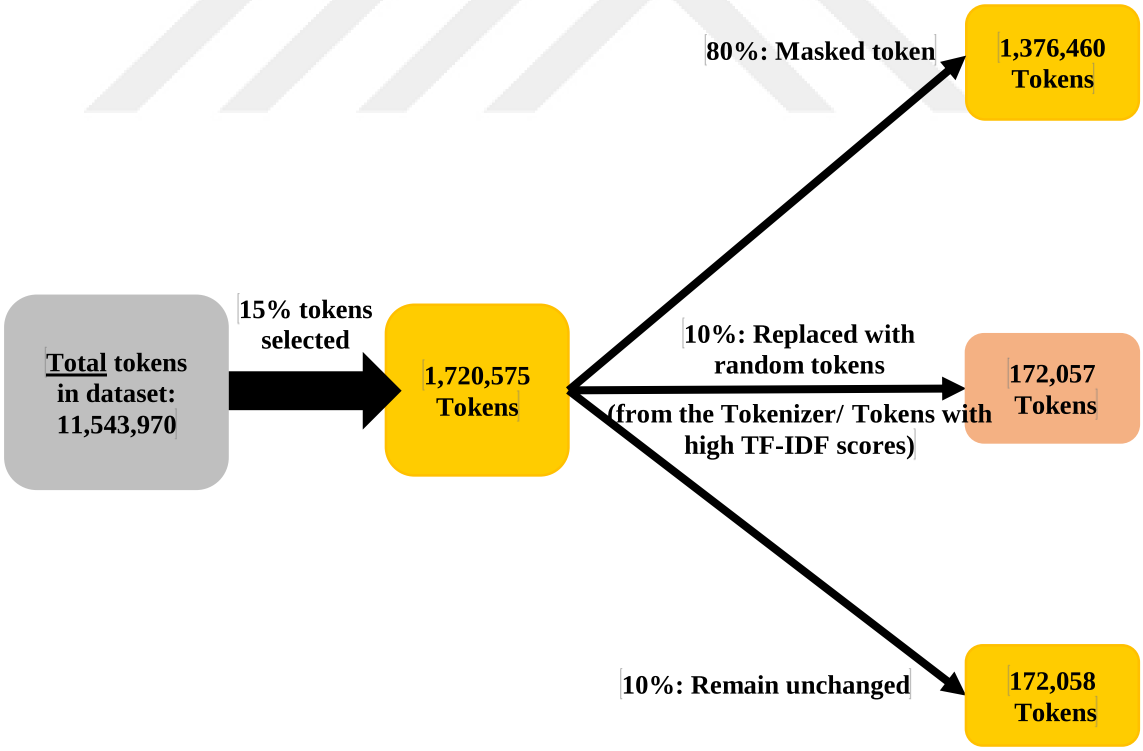

Figure 4.1: Token selection and transformation in MLM dataset creation.

As shown in Figure 4.1, 172,057 tokens will be replaced by random tokens. In the normal MLM_80_10_10 setting, these tokens are randomly selected from the Tokenizer. However, in our modified MLM_80_(10_TF_IDF)_10 approach, these tokens are chosen from the set of tokens with high TF-IDF scores, indicating their significance as keywords across all documents.

To implement our modified approach, we calculate TF-IDF scores for all tokens in the dataset, which consists of 124 documents, while excluding stopwords, punctuation, and numerical values from the text. Subsequently, we select 172,057 tokens with the highest TF-IDF scores. Table 4.7 presents the top 30 tokens out of the 172,057, ranked by their TF-IDF scores (all numbers rounded to 2 decimal places).

Table 4.7: Top 30 Tokens with highest TF-IDF scores.

Tokens	doc0	doc1	doc2	doc3	doc …	doc122	doc123	TF-IDF Scores
bir	0.37	0.16	0.27	0.30	…	0.25	0.26	35.38
sözleşme	0.17	0.09	0.08	0.10	…	0.09	0.11	11.98
nin	0.09	0.05	0.11	0.11	…	0.07	0.06	10.20
olarak	0.05	0.05	0.16	0.07	…	0.07	0.08	9.18
lar	0.05	0.04	0.08	0.10	…	0.05	0.05	7.99
bu	0.08	0.09	0.09	0.05	…	0.08	0.05	7.41
tarafından	0.06	0.04	0.08	0.08	…	0.09	0.06	7.11
yük	0.10	0.05	0.04	0.10	…	0.06	0.02	7.10
taraf	0.07	0.04	0.03	0.06	…	0.04	0.02	7.08
nın	0.13	0.16	0.03	0.09	…	0.04	0.07	7.03
kira	0.13	0.03	0.06	0.09	…	0.03	0.03	6.54
len	0.04	0.04	0.04	0.08	…	0.06	0.04	6.26
işbu	0.02	0.03	0.01	0.04	…	0.04	0.05	5.98
ler	0.06	0.04	0.05	0.06	…	0.04	0.08	5.92
herhangi	0.07	0.02	0.02	0.09	…	0.08	0.05	5.85
ilgili	0.05	0.04	0.04	0.04	…	0.03	0.07	5.37
talep	0.04	0.04	0.06	0.05	…	0.03	0.03	5.34
ifa	0.18	0.02	0.05	0.03	…	0.05	0.06	5.26
kabul	0.03	0.03	0.06	0.03	…	0.02	0.06	5.25
türk	0.04	0.04	0.04	0.06	…	0.03	0.04	5.10
hizmet	0.03	0.00	0.00	0.00	…	0.00	0.00	4.81
olan	0.03	0.02	0.05	0.03	…	0.04	0.03	4.77
ödeme	0.02	0.03	0.05	0.04	…	0.03	0.04	4.66
söz	0.06	0.01	0.04	0.02	…	0.02	0.02	4.63
taahhüt	0.02	0.03	0.05	0.03	…	0.02	0.02	4.59
konusu	0.01	0.03	0.13	0.03	…	0.06	0.04	4.56
hak	0.03	0.03	0.03	0.05	…	0.03	0.02	4.52
ilişkin	0.04	0.01	0.06	0.04	…	0.04	0.03	4.51
madde	0.06	0.02	0.05	0.04	…	0.04	0.04	4.48
teslim	0.03	0.03	0.03	0.04	…	0.03	0.02	4.42

Ignoring the mathematical details and specific TF-IDF scores in Table 4.7, Figure 4.2 displays the first 200 tokens with the highest TF-IDF scores. As shown, the token with the highest TF-IDF score is "bir", followed by "sözleşme", "nin", "olarak", "bu", "tarafından", "yük", and so on. In this list, you can observe that among the common tokens, several key tokens such as "sözleşme", "taraf", "talep", "hizmet", "kabul", "söz", "konusu", "hak", "hükümleri", "maddesi", "teminat", "esas", "mahkeme", "bedeli", and more are significantly represented.

bir, **sözleşme**, nin, olarak, lar, bu, tarafından, yük, **taraf**, nın, **kira**, len, işbu, ler, herhangi, ilgili, **talep**, ifa, **kabul**, türk, **hizmet**, olan, ödeme, **söz**, taahhüt, **konusu**, **hak**, ilişkin, madde, teslim, alan, halinde, tir, na, den, ga, eder, ları, diğer, yer, ver, ek, şekilde, uygun, ni, in, er, yazılı, göre, de, sözleşmesi, da, türlü, iş, dan, tır, le, olduğu, şirket, üzere, beyan, ci, ya, tem, genel, li, yerine, dir, yi, ki, **kapsamında**, ifade, nı, me, belirtilen, **yükümlülük**, olması, ancak, sı, nda, olup, sayılı, la, ların, üçüncü, bilgi, gün, olacaktır, si, gerekli, uyarınca, dır, at, **karşı**, **hüküm**, içerisinde, **süre**, birlikte, **hakkı**, edi, lerin, içinde, yapılan, edilen, **kanun**, ta, lik, nun, **hukuk**, sigorta, kadar, ndan, sa, **süresi**, hiçbir, durumunda, te, un, **ceza**, **karar**, ma, lı, arasında, alt, tarihi, olmak, hukuku, tan, yönetim, ait, an, özel, kredi, geçerli, sonra, ve, ak, eden, gizli, itibaren, **hükümleri**, ayrıca, ir, aykırı, şart, **kanunu**, kendisine, borç, bildirim, satıcı, yeni, sahip, yapılacak, edilmesi, başka, kurulu, alıcı, lık, sona, **fesh**, **zarar**, verilen, **maddesinde**, al, **sorumlu**, kişi, hükmü, tek, doğan, ca, aynı, ay, se, **esas**, iki, gerek, **vergi**, türkiye, proje, müşteri, ticaret, dahil, yükümlülüğü, imzala, kamu, durumda, **maddesi**, mal, inat, **teminat**, **mahkeme**, kendi, **yetkili**, üzerinde, hizmetleri, hisse, derhal, gereken, olduğunu, ayrı, **dava**, bulunan, lere, ten, önce, dışında, yükümlüdür, uygulama, bildir, im, tr, kesin, **onay**, sorumluluğu, **bedeli**, …

Figure 4.2: Top 200 tokens with highest TF-IDF scores.

Considering extracted 172,057 Tokens with most highest TF_IDF score which includes common tokens and key tokens, in in MLM_80_(10_TF_IDF)_10 we decided to select tokens from this list randomly with replacement that means selected tokens can be repeated. Then after applying our strategy, in tables 4.8, we try to show the first 150 (from 172,057 Tokens) random chosen tokens from Tokenizer vocabulary list in “MLM_80_10_10” and from tokens with high TF_IDF score in “MLM_80_(10_TF_IDF)_10”.

As Table 4.8 illustrates, the random tokens extracted from the Tokenizer include examples such as personal names (e.g., "Hakan"), country names (e.g., "ABD"), numbers, symbols (e.g., "##ジ"), and abbreviations (e.g., "CR"). Some of these words may appear only once in the entire set of documents, and some are quite common and not necessarily key tokens of significance across all documents. On the other hand, tokens obtained from the list of tokens with the highest TF-IDF scores are highly relevant to the main topic of legal texts. They represent important and significant terms within the texts. Examples include terms like "incelemesi", "yeterlik", "kaynat", "komisyon", "şarttır", "kanunu", "sonuçların", "işletmeler", "müdahalede", "hükümleri", "konusundaki", "enflasyonun", "teçhiz", and "hususunda." These terms are directly related to the subject matter and carry significant meaning in the legal context.

Table 4.8: Comparison of randomly chosen tokens in "MLM_80_10_10" and "MLM_80_(10_TF_IDF)_10".

Randomly selected tokens from Tokenizer In MLM_80_10_10	Randomly selected tokens from TF_IDF list In MLM_80_(10_TF_IDF)_10
bany, emisyon, larından, katil, ABD, akraba, Kayıp, departman, teki, Şili, biliyorsun, neleri, memiştim, çıkmadan, kapsayacak, maları, sezonun, kalbinde, ó, hücreleri, Antal, 550, Kes, erecek, mişler, saldırısı, 93, vatandaşları, tezah, üssü, mesiyle, ting, sıkıntılar, nemli, CR, spekülasyon, gör, Hakan, Yugoslavya, 46, kuruluşunun, arkan, NO, mantığı, ödenek, BA, malzemeleri, star, eğiyle, pank, roid, ic, aksi, yerinin, MAZ, hun, ecilik, hatırlar, Sonu, önceki, ĝ, bilim, Ø, lendiğini, Harvard, risk, var, Kle, DU, sağlanmıştır, memnuniyeti, tığımız, İran, layacağını, sektörünün, G, dizinin, 什, yükümlül, Balkan, şok, komit, unu, Kağıt, Süper, beklet, إ, milliyet, Edirne, yed, 😄, haftalar, ##törler, mıştık, ɡ, miyim, ây, endüstriyel, verilerini, ##ジ, keyifle, 000, cezasını , altının, kokteyl, Fo, DO, dağıt, _, imzalı, kazı, balı, Tebliğ, talım, Hak, yasalara, koşullarına, ebilen, ödev, ##Ä±, ##s, ##ılmadı, ödemeleri, チ, arıyoruz, tanır, Yarım, Üzgünüm, Yas, Kuv, yürütüyor, çıkarı, sper, düzenley, Bor, izleyici, sahiplerinin, ulaşmasını, Sout, eksiklik, eð, bulunmamaktadır, poz, şekil, deney, nü, yürüyüşe, SAT, ayrılmış, füze, Münih, Yolu, yel, Anal, Arkadaş, Korkut, çevredeki, ş, yenik, ##ılmaktadır, zenci, fah, Dergi, Finlandiya, baz, …	**incelemesi**, değişebilir, gömülü, alındıktan, **bütçeye**, savaşan, hepimizin, **yeterlik**, üreticiler, duyunca, ilah, **kaynat**, **komisyon**, bazında, etmişlerdir, dola, sağlarken, doğrulama, **şarttır**, doğruluk, zenginlik, temizliği, irtibata, **kanunu**, projesinin, **sonuçların**, huzur, büy, ayrılıp, görül, **işletmeler**, sıf, müdahalede, çıkarıyor, sunmaktayız, sük, **hükümleri**, taşıyan, odayı, opt, edeceksiniz, konusundaki, tasarım, sesi, **enflasyonun**, gösteriş, az, **teçhiz**, **hususunda**, adedi, istedim, öğretmenlere, çeki, **bütçeye**, yükselmesi, kadro, **işlemleri**, **fiyat**, kendilerinden, **anlaşma**, anlayışını, **siyasal**, düşmesine, **maddelerin**, öreninde, seansta, lakin, oral, arzu, tertip, getirdik, **şart**, karakterler, rot, **şeffaf**, kaybedecek, erkekleri, planlaması, görebilirsiniz, uzaklıkta, bilimleri, sistemler, **tutarak**, **devlete**, hazırlandı, istediği, ilerle, anahtarı, geçirdiği, ettiklerini, **borçlarını**, toplantısı, çıktılar, eyaletinde, arttı, açıdan, yapabiliyor, sarısı, sağlama, , **hizmetler** katılmış, **seçim**, **tüzel**, öncülüğünde, oynan, notlar, yansıması, gecikme, çevresel, yazımda, mekân, **belediyeler**, verilen, dileriz, karakterini, zorunda, iddi, geçirir, karşıyay, zorundayız, **çöküş**, etmediği, tahvil, kaybed, karakterin, stratej, odayı, kaydet, **yetin**, yolu, önleyici, değişti, **halle**, **mesel**, kaydır, kimselerin, belirgin, depr, sürücüler, **zarar**, yaşar, **istihdam**, ihti, …

Therefore, to emphasize the distinctions between the original "MLM_80_10_10" and our modified "MLM_80_(10_TF_IDF)_10", while referring to the sample dataset from the original MLM displayed in Table 4.6, Table 4.9 demonstrates the differences in the replaced token between both methods:

- **In "MLM_80_10_10"**, 'sonucu' (*English version*: "results") is replaced with a randomly selected token from the Tokenizer. In this example, it is replaced by "sokak" (*English version*: 'street').
- **In "MLM_80_(10_TF_IDF)_10"**, 'sonucu' is replaced with a randomly selected token from the tokens list with a high TF_IDF score. In this example, it is replaced by "Avukat" (*English version*: 'Lawyer').

As it is clear in Table 4.9, we have added tokens to the input that are more related to the text, and the model will encounter text with more significant tokens.

Table 4.9: Comparison of masked sentences in "MLM_80_10_10" and "MLM_80_(10_TF_IDF)_10".

Method	MLM Masked Sentence	MlM lable (original sentence)
MLM_80_10_10	"'[CLS]', **'İkinci'**, 'ihtimale', **'[MASK]'**, 'ise', ',', 'bu', **'[MASK]'**, 'bir', 'yanlışlık', **'sokak'**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', '[PAD]',… , '[PAD]' "	"'[CLS]', **'İkinci'**, 'ihtimale', **'göre'**, 'ise', ',', 'bu', **'hüküm'**, 'bir', 'yanlışlık', **'sonucu'**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', '[PAD]',… , '[PAD]' "
MLM_80_(10_TF_IDF)_10	"'[CLS]', **'İkinci'**, 'ihtimale', **'[MASK]'**, 'ise', ',', 'bu', **'[MASK]'**, 'bir', 'yanlışlık', **'Avukat'**, 'sevk', '##edi', '##lm', '##iştir', '.', '[SEP]', '[PAD]',… , '[PAD]' "	

4.2. MODELS RESULTS

After creating the dataset for the pre-training stage, we conducted pre-training on the original BERT model and several proposed models using two distinct methods: "further pre-training using BERTurk's weights" and "pre-training models from scratch". Additionally, we pre-trained these models using datasets of varying sizes. Subsequently, we fine-tuned them for two specific subtasks: NER and multi-label text classification. The results of these experiments will be discussed in the following sections.

4.2.1. Results of Models Trained on Weights from BERTurk

In our initial attempts, we trained our model using the weights from BERTurk. This involved downloading the BERTurk model (dbmdz/bert-base-turkish-cased) and then utilizing this model's weights to pre-train on our dataset, adapting the model's parameters accordingly. The results of the fine-tuning stage are presented in Table 4.10, where we pre-trained the original BERT model and five proposed models using the weights from BERTurk. It is important to mention that these results are based on 15 rounds of training: 5 rounds of pre-training on the pre-train dataset, followed by 10 rounds of fine-tuning (including NER and multi-label text classification tasks) on each pre-trained model using two labeled datasets for NER and multi-label text classification tasks.

Table 4.10: Fine-tuning results of models pre-trained on BERTurk weights.

Pre-trained models on BERTurk weights	Fine-tuned models					
	Multi-label Text classification epoch=1			NER epoch=1		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Original BERT: NSP_MLM_80_10_10	88.90	71.53	79.27	90.39	94.14	92.23
SOP_MLM_80_10_10	88.77	71.75	79.36	90.25	94.16	92.17
MLM_80_10_10	88.35	71.38	78.96	90.70	94.21	92.42
MLM_80_(10_TF_IDF)_10	88.74	71.65	79.29	90.24	94.19	92.17
MLM_80_(20_TF_IDF)_0	88.55	71.59	79.17	90.57	94.23	92.36

As shown in Table 4.10, there are minimal differences when NSP is replaced by SOP, NSP is removed, or when we modified the MLM task. These differences are on the order of hundredths of decimals. It appears that BERTurk, which was pre-trained on a substantial dataset comprising 35GB of data (44,046,976,662 tokens), is not significantly influenced by our much smaller dataset (50MB with 11,543,970 tokens). Consequently, we made the decision to proceed with pre-training our model from scratch.

4.2.2. Results of Models Trained from Scratch

In this section, we pre-trained the original BERT model and five proposed models from scratch. Here, we provide more detailed information regarding the results of fine-tuning these pre-trained proposed models:

4.2.2.1. Results of Replacing NSP with SOP

After substituting NSP with SOP, the results are presented in Table 4.11. A comprehensive analysis comparing the modified model (SOP_MLM_80_10_10) with the original BERT model (NSP_MLM_80_10_10) underscores the positive influence of this substitution on both NER and multi-label text classification tasks.

Table 4.11: Impact of replacing NSP with SOP on model performance.

Pre-trained models on BERTurk weights	Fine-tuned models					
	Multi-label Text classification epoch=1			NER epoch=1		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Original BERT: NSP_MLM_80_10_10	84.36	48.79	61.82	50.77	73.53	60.07
SOP_MLM_80_10_10	**84.93**	**54.71**	**66.55**	**51.22**	**73.93**	**60.51**

1. **In Multi-label Text Classification:** The F-measure experiences a notable improvement, surging from 61.82 in the original BERT model (NSP_MLM_80_10_10) to 66.55 in the SOP-based model (SOP_MLM_80_10_10). This improvement underscores the effectiveness of utilizing SOP in enhancing the model's ability to classify text into multiple categories.
2. **In NER:** The F-measure exhibits a slight yet meaningful advancement. Specifically, it rises from 60.07 in the original BERT model (NSP_MLM_80_10_10) to 60.51 in the model incorporating SOP (SOP_MLM_80_10_10). While the improvement may appear modest, it signifies the positive impact of SOP on the model's capacity to recognize named entities within text, contributing to overall performance gains.

4.2.2.2. Results of Retaining MLM and Removing NSP

In this section, we explore the impact of retaining MLM while removing NSP from BERT model. The results are detailed in Table 4.12 below, showcasing a comparison between the original BERT model (NSP_MLM_80_10_10) and the modified model (MLM_80_10_10).

Table 4.12: Impact of retaining MLM and removing NSP on model performance.

Pre-trained models on BERTurk weights	Fine-tuned models: Multi-label Text classification epoch=1			Fine-tuned models: NER epoch=1		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Original BERT: NSP_MLM_80_10_10	84.36	48.79	61.82	50.77	73.53	60.07
MLM_80_10_10	**86.18**	**61.88**	**72.03**	**70.33**	**82.33**	**75.86**

Upon removing the NSP component and focusing solely on MLM, we observe notable improvements in both the multi-label text classification and NER tasks:

1. **In Multi-label Text Classification:** The F-measure significantly increases from 61.82 in the original BERT model (NSP_MLM_80_10_10) to an impressive 72.03 in the modified model (MLM_80_10_10).
2. **In NER:** Similarly, the NER task benefits from this modification, with the F-measure rising from 60.07 in the original BERT model (NSP_MLM_80_10_10) to 75.86 in the modified model (MLM_80_10_10).

Additionally, when comparing the results of replacing NSP with SOP on model performance (as seen in Table 4.11), it becomes evident that, in general, retaining MLM and removing NSP has shown better performance compared to the alternative of replacing NSP with SOP.

4.2.2.3. Results of Combining MLM with TF-IDF

In this section, we delve into the results obtained after combining the MLM with TF-IDF. The performance outcomes for various models are summarized in Table 4.13.

Table 4.13: Impact of combining MLM with TF-IDF on model performance.

Pre-trained models on BERTurk weights	Fine-tuned models: Multi-label Text classification epoch=1			Fine-tuned models: NER epoch=1		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Original MLM_80_10_10	86.18	**61.88**	72.03	70.33	82.33	75.86
MLM_80_0_20	86.31	60.05	70.82	**71.85**	**83.47**	**77.23**
MLM_80_(10_TF_IDF)_10	86.59	61.75	72.09	68.73	81.62	74.62
MLM_80_(20_TF_IDF)_0	**86.61**	61.83	**72.15**	67.53	81.15	73.72

In the context of combining MLM with TF-IDF, we trained several model variations. In each model, after randomly selecting 15% of the tokens, specific token manipulation strategies were applied. To provide a detailed explanation of these strategies:

- **Original MLM_80_10_10:**
  - 80% of tokens are masked.
  - 10% of tokens are replaced with random tokens from the Tokenizer.
  - 10% of tokens remain unchanged.
- **MLM_80_0_20:**
  - 80% of tokens are masked.
  - 20% of tokens remain unchanged.
- **MLM_80_(10_TF_IDF)_10:**
  - 80% of tokens are masked.
  - 10% of tokens are replaced with random tokens selected from a token list with high TF-IDF scores.
  - 10% of tokens remain unchanged.
- **MLM_80_(20_TF_IDF)_0:**
  - 80% of tokens are masked.
  - 20% of tokens are replaced with random tokens selected from a token list with high TF-IDF scores.

The results and observations for both multi-label text classification and NER tasks are discussed below:

1. **Multi-label Text Classification:** In this task, generally, adding more random tokens has a positive impact on performance compared to the original MLM_80_10_10. Additionally, instead of selecting tokens from the Tokenizer, using tokens from a token list with high TF-IDF scores also has a positive effect:
   - **In the "MLM_80_0_20"**, where no random tokens are added, the F-measure decreases from 72.03 in "MLM_80_10_10" to 70.82. This result highlights the positive impact of random tokens on the multi-label classification task.
   - **In "MLM_80_(10_TF_IDF)_10"**, where random tokens are selected based on high TF-IDF scores, there is a slight improvement in F-measure from 72.03 to 72.09. It's worth noting that this dataset has a limited size (50 MB), and more substantial differences may arise with larger datasets.
   - **In "MLM_80_(20_TF_IDF)_0"**, the improvement persists as compared to "MLM_80_(10_TF_IDF)_10" by further incorporating random tokens (20%), leading to an increase in F-measure from 72.09 to 72.15.

2. **NER:** Conversely, in the NER task, adding random tokens has a negative impact on performance compared to the original MLM_80_10_10:
   - **In “MLM_80_0_20”**, where there are no random tokens added, the F-measure increases from 75.86 in the original “MLM_80_10_10” to 77.23. This result highlights that for NER, which focuses on each token individually, adding random tokens introduces too much distraction.
   - **In “MLM_80_(10_TF_IDF)_10”**, where random tokens are selected based on high TF-IDF scores, there is a slight decrease in F-measure from 75.86 to 74.62.
   - **In “MLM_80_(20_TF_IDF)_0”**: This negative impact of adding random tokens continues in “MLM_80_(20_TF_IDF)_0”, with a further reduction in F-measure from 74.62 in “MLM_80_(10_TF_IDF)_10” to 73.72.

To summarize, Table 4.14 presents the summarized results of fine-tuning on pre-trained models that were pre-trained from scratch. These results are based on a total of 18 rounds of training: 6 rounds of pre-training on the pre-train dataset, followed by 12 rounds of fine-tuning, which includes tasks such as NER and multi-label text classification.

Table 4.14: Summary of fine-tuning results for models pre-trained from scratch.

	Fine-tuned models					
Pre-trained models on BERTurk weights	**Multi-label Text classification epoch=1**			**NER epoch=1**		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Original BERT: NSP_MLM_80_10_10	84.36	48.79	61.82	50.77	73.53	60.07
SOP_MLM_80_10_10	84.93	54.71	66.55	51.22	73.93	60.51
MLM_80_10_10	86.18	**61.88**	72.03	70.33	82.33	75.86
MLM_80_0_20	86.31	60.05	70.82	**71.85**	**83.47**	**77.23**
MLM_80_(10_TF_IDF)_10	86.59	61.75	72.09	68.73	81.62	74.62
MLM_80_(20_TF_IDF)_0	**86.61**	61.83	**72.15**	67.53	81.15	73.72

As demonstrated in Table 4.14, it is evident that replacing NSP with SOP generally has a positive effect on the results. However, the removal of NSP/SOP has a much more significant impact on the results. In multi-label text classification, the addition of random tokens has a positive effect. The best-performing model in multi-label text classification is MLM_80_(20_TF_IDF)_0, where 20% of selected tokens are replaced with random tokens. Additionally, the proposed modification, which involves selecting random tokens from the token list with high TF-IDF scores instead of relying solely on the Tokenizer, has a positive effect on multi-label text classification. This demonstrates that challenging the model with more keywords (token list with high TF-IDF scores) has a positive impact on the performance of multi-label text classification.

In contrast, in the NER task, surprisingly, adding random tokens has a negative effect on the results. The best model for this task is MLM_80_0_20, where only the MLM is used, and 80% of randomly selected tokens are masked without replacing any random tokens.

4.2.3. Results of Models Trained on a Larger Corpus

In this section, we decided to pre-train our best models for both multi-label text classification "MLM_80_(20_TF_IDF)_0" and the NER task "MLM_80_0_20" on a larger corpus. Subsequently, we fine-tuned these two new models on the same labeled data for both NER and multi-label text classification tasks.

To achieve this, we augmented our initial 50 MB corpus by adding an additional 111 random documents, increasing the corpus size from 50 MB to 100 MB. Table 4.15 presents the results of this four-round training process (comprising 2 pre-training rounds and 2 fine-tuning rounds).

As shown in this table, increasing the volume of corpus during the pre-training stage had a positive impact on the performance of both the NER and multi-label classification tasks. In the NER task, there was a significant improvement in F-measure, increasing from 77.23 to 84.67. In multi-label classification, although the improvement was modest, it was still important, with the F-measure increasing from 72.15 to 72.23. This slight improvement may suggest that even larger corpus are required for more substantial improvements.

Table 4.15: Impact of corpus size used in pre-training stage on model performance.

	Fine-tuned models					
Corpus size in pre_training stage	**Multi-label Text classification model_type: MLM_80_(20_TF_IDF)_0**			**NER model_type: MLM_80_0_20**		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
50 MB	86.61	61.83	72.15	71.85	83.47	77.23
100 MB	**86.76**	**61.88**	**72.24**	**81.52**	**88.06**	**84.67**

In conclusion, this section demonstrates that utilizing larger corpus during the pre-training phase can lead to improved model performance, and the volume of data directly influences the models' effectiveness.

4.2.4. Results of Proposed Model's Comparison with BERTurk Model

In this section, we compare our domain-specific pre-trained models (trained on 100 MB with 11,543,970 tokens) with Schweter's BERTurk (pretrained on 35 GB with 4,404,976,662 tokens). We fine-tuned both models for NER and multi-label text classification using BERTurk. The results are presented in Table 4.16.

At first glance, it is evident that BERTurk performs better overall. There is approximately a 7.67% and 7.79% difference in the F-measure between our modified models and BERTurk in NER and multi-label text classification, respectively. However, it is crucial to underscore the significant differences in corpus sizes. Our models were pre-trained on a relatively modest 100 MB corpus, which constitutes a mere 0.2% of the scale of BERTurk's extensive 35 GB corpus (100 MB/35,000 MB). This substantial contrast in corpus size likely plays a pivotal role in the variance in performance observed between the models.

It is essential to highlight that our training process on a 100 MB corpus takes less than 4 hours, while training on such a large dataset (35 GB) would require several weeks.

Table 4.16: Performance comparison between our modified models and BERTurk.

pre_trained model		Fine-tuned models					
		Multi-label Text classification			NER		
		Precision	Recall	F-measure	Precision	Recall	F-measure
BERTurk Corpus size: 35 GB	**Model type:** NSP_MLM_80_10_10 (original BERT)	**88.88**	**72.59**	**79.91**	**90.75**	**94.25**	**92.46**
Our modified models Corpus size: 100 MB	**Model type:** MLM_80_(20_TF_IDF)_0	86.76	61.88	72.24	-	-	-
	Model_type: MLM_80_0_20	-	-	-	81.52	88.06	84.67

Due to GPU and time limitations, we could not pre-train our models on a larger corpus. However, there is a hypothesis that with a larger corpus related to legal text (around 4 GB), our models “MLM_80_(20_TF_IDF)_0” and “MLM_80_0_20” may perform similarly or even better than BERTurk, which is pre-trained on large and non-specific legal text.

5. DISCUSSION AND CONCLUSION

Recent advancements in language modeling have brought about significant progress in the field of NLP. The introduction of the Transformer Neural Network, along with techniques such as self supervised pre-training and transfer learning, has paved the way for transformative models like BERT. BERT has demonstrated its effectiveness by achieving outstanding results across a variety of NLP tasks, including sentiment analysis, text classification, NER and more (Devlin et al., 2018). While Google's BERT has shown remarkable performance, there remains an opportunity for further enhancements, particularly when customizing these models for specific domains and non-English contexts.

In our study, our primary focus was on enhancing the performance of the BERT model within the legal Turkish domain. Our research efforts were centered around making modifications to the pre-training phase, specifically targeting the MLM and NSP subtasks. We embarked on the task of pre-training specialized versions of the BERT Model from scratch using legal Turkish corpus. We pursued three main ideas to improve BERT's performance, which included replacing NSP with SOP, removing NSP, and combining MLM with TF-IDF, while also applying different masking strategies within the original MLM_80_10_10. Following this, we fine-tuned these customized models for NER and multi-label text classification. To evaluate their effectiveness, we conducted a comprehensive comparison of the results achieved by these customized models against the performance of the original BERT model.

After examining the YÖKTEZ database within the legal Turkish domain as well as other Turkish domains, we did not find any prior thesis studies (including master's and doctoral studies) focusing on the optimization and modification of the MLM and NSP subtasks to enhance BERT's performance. A systematic literature review of past studies confirmed that our modified approaches, particularly our adjusted MLM approach, are entirely unique and have not been explored in prior researches. In the Turkish legal domain, only two recent models have been introduced: BERTurk-Legal (Öztürk, 2023), designed for retrieving previous legal cases, and HukukBERT (Akça, 2023), focusing on multi-class and multi-label crime classification. These models emphasize further pre-training MLM-based BERTurk using legal domain-specific corpora. Unlike our study, they do not pre-train from scratch, and they do not directly compare the original BERT model (with NSP and MLM) to the MLM-based BERT. Their approach aims to enhance BERT's performance through additional training on legal corpora. In contrast, our study employs a fixed corpus and aims to enhance BERT's performance by modifying the MLM and NSP tasks to find the best pretrained model for multi-label classification and NER tasks.

Regarding the legal Turkish corpus (unlabeled data) and the labeled data required for both the pre-training and fine-tuning stages, for the pre-training stage, we obtained legal Turkish text from legal-related thesis documents available in the YÖKTEZ. After pre-processing this corpus, we randomly selected 50 MB of data, which consisted of 124 files, comprising a total of 347,147 sentences and containing 11,543,970 tokens (Table 3.1). Additionally, labeled data for the fine-tuning tasks were provided by NewMind, a pioneering legal tech company located in Istanbul, Turkey. The labeled data for NER task followed the BIO format and included 614,541 sentences (305 MB), covering 11 general labels that encompassed various categories such as 'Person', 'Law', 'Publication', 'Government' and others (Table 3.3). Furthermore, the labeled data for the Multi-label Text Classification stage comprised 175,279 sentences (63 MB) and encompassed 31 labels, including categories like 'Heading', 'Penalty', 'Transfer_and_assignment', 'Termination', 'Confidentiality' and more (Table 3.5). It is important to note that these labeled data are unique and have not been utilized in any previous studies due to their sensitive nature and proprietary ownership by NewMind. Consequently, they cannot be publicly disclosed or shared.

After more than 60 rounds of training in both the pre-training and fine-tuning stages for NER and multi-label text classification, in the following text, we conducted a comparison of the results of the proposed models with the original BERT model and with studies conducted by others.

Regarding the Results of Models Trained on Weights from BERTurk or from Scratch, in our initial attempts, we trained our proposed models using the weights from BERTurk. However, minimal differences were observed when NSP was replaced by SOP, NSP was removed, or when we combined the MLM task with TF-IDF (Table 4.10). These differences were on the order of hundredths of decimals. It appears that BERTurk, which was pre-trained on a substantial dataset comprising 35GB of data (44,046,976,662 tokens), is not significantly influenced by our much smaller dataset (50MB with 11,543,970 tokens). Consequently, we made the decision to proceed with pre-training our model from scratch. In literature review, several models like RomanianBERT (Dumitrescu et al., 2020), AraBERT (Antoun et al., 2020), FinnishBERT (Virtanen et al., 2019), RuBERT (Kuratov and Arkhipov, 2019), and BERTurk (Schweter, 2020) focus on pre-training monolingual BERT models from scratch, retaining the original BERT structure, including MLM and NSP. In contrast, our approach involves modifying the BERT structure. Additionally, there are BERT models that concentrate on further pre-training for specific domains, such as BERTurk-Legal (Öztürk, 2023) and HukukBERT (Akça, 2023), which perform additional pre-training on BERTurk using small Turkish legal corpora. Other examples include BioBERTurk (Türkmen et al., 2023a), which refines BERTurk for biomedicine, ABioNER (Boudjellal et al., 2021), which extends the pre-training of AraBERT with an Arabic biomedical corpus, and RadBERT (Yan et al., 2022), which adapts a BERT-based model for the English radiology report corpus through further pre-training. In contrast, our approach involves pre-training from scratch and modifying the BERT structure.

Regarding the Results of Replacing NSP with SOP in the original BERT model, a comprehensive analysis comparing the modified model (SOP_MLM_80_10_10) with the original BERT model (NSP_MLM_80_10_10) underscores the positive influence of this substitution on both NER and multi-label text classification tasks (Table 4.11). Although the F-measure experiences a notable improvement (4.73%) in multi-label Text Classification, in NER, the F-measure exhibits a slight yet meaningful advancement (0.44%), signifying the positive impact of replacing NSP with SOP on the model's performance, which was pre-trained on the small 50MB legal Turkish corpus. This positive impact could potentially be even greater on a larger corpus. In the literature, models like ALBERT (Lan et al., 2019), StructBERT (Wang et al., 2019), and LaMBERT (Milosheski, 2020), as well as BERTje (De Vries et al., 2019), employ SOP in the pre-training phase. In all these models, they achieved better results compared to the original BERT, similar to what we have achieved. However, it is important to note that these models not only replaced NSP with SOP but also introduced additional changes to the core structure of BERT, resulting in a new model or a modified version of BERT. For instance, in ALBERT, parameter-reduction techniques were added, while StructBERT incorporated an innovative structural pre-training approach that includes word and sentence structures. In LaMBERT, SOP was modified in the selection of paired sentences during training, opting for sentences from the same paragraph or a similar context, rather than random sentences from different paragraphs. In contrast, our study retains the main core of the BERT structure and solely replaces NSP with SOP, following the approach presented in ALBERT, without making further modifications. We demonstrate that replacing NSP with SOP, without additional changes, can enhance BERT's performance.

Regarding the Results of Retaining MLM and Removing NSP in the original BERT model, where we retained the MLM while removing the NSP (MLM_80_10_10), significant improvements were observed in both multi-label text classification and NER tasks (Table 4.12). In comparison to the original BERT model (NSP_MLM_80_10_10), the F-measure increased significantly by 10.21% in multi-label text classification and by 15.79% in NER. Furthermore, when comparing these results with those obtained by replacing NSP with SOP in model performance (Table 4.11), it becomes evident that, in general, retaining MLM and removing NSP has shown better performance compared to the alternative of replacing NSP with SOP. This consistency underscores the idea that NSP or SOP may introduce unnecessary complexity or confusion into the model's training, while focusing exclusively on MLM tasks tends to yield more favorable outcomes. In the literature, some main models like RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), and ELECTRA (Clark et al., 2020) also remove the NSP task, but they significantly change the structure of the BERT models, leading to entirely new models. In these studies, the authors did not compare SOP with NSP but simply removed NSP and showed better results compared to the original BERT. However, in this step of our approach, without making any changes to the original BERT structure, we first compared two versions of BERT results: one with MLM and NSP in the pre-training task and the other with MLM and SOP in the pre-training task, and then trained the MLM-based BERT model. In general, in contrast to ALBERT (Lan et al., 2019), StructBERT (Wang et al., 2019), and LaMBERT (Milosheski, 2020), and BERTje (De Vries et al., 2019), we observed that SOP also reduces BERT's performance. Therefore, our study suggests that both SOP and NSP may introduce confusion into the model, and it may be better to remove them.

Regarding the Results of Combining MLM with TF-IDF, in the Original MLM (Devlin et al., 2018), the "**10**" in "MLM_80_**10**_10" rule indicates that after randomly selecting 15% of the tokens (MLM selected tokens), 10% of these tokens are replaced by random tokens from the Tokenizer's vocabulary list. Instead of substituting tokens with randomly chosen ones from the Tokenizer's vocabulary list, we propose an innovative approach, replacing them with tokens that have high TF-IDF values, known as keywords across all documents (keywords list). We pretrained three different models: **1. MLM_80_0_20** (without replacing any random tokens),

2. MLM_80_(10_TF_IDF)_10, where 10% of MLM-selected tokens are replaced by random tokens from the keywords list, and **3. MLM_80_(20_TF_IDF)_0**, where 20% of MLM-selected tokens are replaced by random tokens from the keywords list. After analyzing the results (Table 4.13), it becomes evident that in multi-label text classification, when no random tokens are added in "MLM_80_0_20", the F-measure decreases (1.21%). However, when 10% of tokens are replaced with random tokens from either the Tokenizer or the keyword list, the F-measure improves. The improvement continues in “MLM_80_(20_TF_IDF)_0” with the addition of more random tokens. Moreover, when comparing “MLM_80_10_10” with “MLM_80_(10_TF_IDF)_10”, where random tokens are selected from the keywords list, there is a slight (0.06%) improvement in F-measure, indicating that our manipulated MLM has better performance than the original MLM. In general, by adding more random tokens from a token list with high TF-IDF scores and challenging the model with more keywords, the results improve significantly (0.12%) in “MLM_80_(20_TF_IDF)_0”. On the other hand, a different trend emerges in NER. Adding random tokens has a negative impact on performance compared to the original MLM. In “MLM_80_0_20”, where no random tokens are added, the F-measure increases (1.37%). However, in both “MLM_80_10_10”, where random tokens are replaced from the Tokenizer, and “MLM_80_(10_TF_IDF)_10”, where random tokens are selected from the keyword list, there is a decrease in F-measure. This negative impact of adding random tokens continues in "MLM_80_(20_TF_IDF)_0", with a further reduction in F-measure. This result highlights that for NER, which focuses on each token individually, adding random tokens introduces too much distraction, negatively affecting the overall performance. In literature, there are some models that have been pretrained with modified MLM, such as BERTje (De Vries et al., 2019) which instead of randomly masking single word pieces, masks consecutive word pieces. SpanBERT (Joshi et al., 2020), on the other hand, in addition to masking random contiguous words, uses special tokens to indicate span boundaries. RoBERTa (Liu et al., 2019) introduces dynamic MLM instead of static MLM. Furthermore, StructBERT (Wang et al., 2019) enhances the MLM task by introducing the Span Order Recovery task. ELECTRA (Clark et al., 2020) combines the MLM with a “replaced token detection” task, and XLNet (Yang et al., 2019) replaces MLM with permutation language modeling. These studies have all shown improved results by modifying the MLM. However, there is no study that attempts to enhance the MLM by combining it with TF-IDF or applying different masking strategies and investigating the tokens that are randomly replaced by other random tokens. Our innovative approach demonstrates that in NER, adding random tokens has a negative effect, while in multi-label classification, adding random tokens, especially those selected based on TF-IDF, has a positive impact on the model's performance.

In summarizing all the results (Table 4.14), it becomes evident that replacing NSP with SOP generally has a positive impact on the BERT model. However, the removal of NSP/SOP has a much more significant influence on the outcomes. In the context of multi-label text classification, the substitution of random tokens demonstrates a favorable effect. The best-performing model in multi-label text classification is MLM_80_(20_TF_IDF)_0, where 20% of MLM-selected tokens are replaced with random tokens from the keyword list. In contrast, in the NER task, the addition of random tokens negatively impacts the results. The most effective model for this task is MLM_80_0_20, where only the MLM is employed, and 80% of MLM-selected tokens are masked without any random token replacements.

It is worth noting that the corpus used in this study is relatively small (50 MB), and more substantial differences may arise with a larger corpus. Therefore, we decided to pre-train our best models for both multi-label text classification (MLM_80_(20_TF_IDF)_0) and the NER task (MLM_80_0_20) on a larger corpus (100 MB). The results show (Table 4.15) that utilizing a larger corpus directly influences the models' effectiveness. In the NER task, there was a significant improvement in F-measure (7.44%), and in multi-label classification, a slight improvement of F-measure (0.09%), which may suggest that even larger corpora are required for more substantial improvements. In general, our findings align with the recommendations in many different studies (e.g.: Yang et al., 2019; Liu et al., 2019; Qiu et al., 2020; Kalyan et al., 2021; Zhou et al., 2023) to train models on larger corpora for improved performance.

This thesis study, like all other studies, was conducted with certain limitations. In future studies, these limitations can be addressed to explore different research possibilities. Regarding the corpus used in the pre-training step, due to the sensitive nature of legal data and the unavailability of legal-related documents, the corpus used for pre-training was extracted from publicly available legal domain theses in 'YÖKTEZ'. However, it is recommended to incorporate texts from legal professionals' files to create a more representative corpus. Additionally, our focus was solely on legal Turkish text. It would be beneficial to pre-train our best models on texts from various fields in Turkish and then compare the results with the original BERT. This would allow us to potentially generalize our findings to the Turkish language as a whole, rather than solely within the legal field. Alternatively, we could explore the use of a multilingual corpus to evaluate the potential for generalizing results across different languages.

From a technical perspective, we exclusively focused on two downstream tasks, namely NER and multi-label text classification, due to GPU memory limitations and limited time for testing various experimental models. We did not extend our analysis to include other NLP downstream tasks, such as question answering and binary label text classification. This provides an opportunity for future research to explore these additional downstream tasks. Furthermore, we only trained all models, both in the pre-training stage and the fine-tuning stage, for a single epoch. It is advisable to conduct longer training sessions with multiple epochs to potentially achieve better results. Additionally, in our effort to enhance the MLM subtask by combining TF-IDF, our focus was on tokens rather than whole words. It is possible to adapt this approach to work at the whole-word level and investigate if it can yield improved results.

Moreover, we were constrained to using a small 50 MB legal corpus in the pre-training task. As mentioned in the results chapter, using a larger corpus during the pre-training phase can lead to improved model performance, as the volume of the corpus directly influences the effectiveness of the models. Therefore, it is advisable to work with larger datasets to achieve more robust generalization of results.

Considering the results of the comparison between our models (pre-trained on a 100 MB legal-specific domain corpus in less than 4 hours) and Schweter's BERTurk (pre-trained on a 35 GB non-specific domain corpus over several weeks), there is a hypothesis that with a larger legal text corpus (around 4 GB), our best models may perform similarly or even better than BERTurk. It is also recommended to consider the time required for pre-training and investigate whether smaller domain-specific corpora with shorter pre-training times can achieve similar or better results compared to larger non-specific domain corpora pre-trained over several weeks. This is another area worth exploring.

As a result, we believe that the findings of this research can serve as a guiding light for future studies in the field of NLP, not only within the legal Turkish domain but also in other specific domains and languages. Furthermore, we consider this study to be of significant importance because Turkish-based models, in contrast to English-based models, come with inherent limitations. Most Turkish-based models are pre-trained on non-specific corpora. In contrast, we focused on Turkish legal-specific domain models and demonstrated impressive performance surpassing that of the original BERT model. We hope that our models and our approach to training them can draw the attention of legal companies, especially innovative legal tech companies like NewMind. These models can be integrated into legal artificial intelligence applications, assisting lawyers and legal professionals in identifying statements, extracting essential information, and streamlining decision-making processes, thus reducing their heavy workloads and time consumption. Additionally, our models, pre-trained on a relatively small 50 MB corpus in less than 4 hours, set an important precedent for smaller companies with limited GPU resources, demonstrating that they can successfully train their own specific domain-modified BERT models with a smaller corpus. Therefore, we believe that this study, in addition to its positive impact on enhancing the performance of the original BERT model in the Turkish context, can significantly contribute to reducing time consumption and costs, as well as improving related processes in legal tech.

With this doctoral thesis, for the first time in Turkey and the world, we introduce a novel modified BERT model that outperforms the original BERT, and our unique approach is among the pioneering studies conducted in the Turkish legal-specific domain worldwide. We hope that the research findings will be particularly valuable in optimizing time consumption, costs, and processes within the realm of legal tech and in the development of pre-training models. These findings can provide assistance to legal tech companies and researchers working in the fields of NLP, computer science and informatics.

REFERENCES

Uncategorized References

Abdul-Mageed, M., Elmadany, A. and Nagoudi, E. M. B., 2020. ARBERT & MARBERT: deep bidirectional transformers for Arabic. *arXiv preprint arXiv:2101.01785*.

Acikalin, U. U., Bardak, B. and Kutlu, M., 2020. BERT Modeli ile Türkçe Duygu Analizi. In *2021 28th Signal Processing and Communications Applications Conference, SIU 2020- Proceedings*. Institute of Electrical and Electronics Engineers Inc.

Ainslie, J., Ontanon, S., Alberti, C., Cvicek, V., Fisher, Z., Pham, P., Ravula, A., Sanghai, S., Wang, Q. and Yang, L., 2020. ETC: Encoding long and structured inputs in transformers. *arXiv preprint arXiv:2004.08483*.

Akça, O., 2023. Natural language processings in legal domain: classification of Turkish legal texts, Thesis (Master), Marmara University.

Akyön, F. Ç., Çavuşoğlu, A. D. E., Cengiz, C., Altinuc, S. O. and Temizel, A., 2022. Automated question generation and question answering from Turkish texts. *Turkish Journal of Electrical Engineering and Computer Sciences,* 30**,** 1931-1940.

Antoun, W., Baly, F. and Hajj, H., 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Aytan, B. and Şakar, C. O., 2023. Deep learning-based Turkish spelling error detection with a multi-class false positive reduction model. *Turkish Journal of Electrical Engineering and Computer Sciences,* 31**,** 581-595.

Bai, H., Shi, P., Lin, J., Tan, L., Xiong, K., Gao, W. and Li, M., 2020. SegaBERT: Pre-training of Segmentaware BERT for Language Understanding. *arXiv preprint arXiv:2004.14996*.

Bayrak, G. and Issifu, A.M., 2022. 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),* 425-430.

Bayrak, G., Toprak, M. Ş., Ganiz, M. C., Kodaz, H. and Koç, U., 2022. Deep learning-based brain hemorrhage detection in ct reports. *Challenges of Trustable AI and Added-Value on Health.* IOS Press.

Beltagy, I., Lo, K. and Cohan, A., 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Beltagy, I., Peters, M. E. and Cohan, A., 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Boudjellal, N., Zhang, H., Khan, A., Ahmad, A., Naseem, R., Shang, J. and Dai, L., 2021. ABioNER: a BERT-based model for Arabic biomedical named-entity recognition. *Complexity*, 2021**,** 1-6.

Bressem, K. K., Adams, L. C., Gaudin, R. A., Tröltzsch, D., Hamm, B., Makowski, M. R., Schüle, C.-Y., Vahldiek, J. L. and Niehues, S. M., 2020. Highly accurate classification of chest radiographic reports using a deep learning natural language model pre-trained on 3.8 million text reports. *Bioinformatics,* 36**,** 5255-5261.

Budur, E., Özçelik, R., Güngör, T. and Potts, C., 2020. Data and representation for turkish natural language inference. *arXiv preprint arXiv:2004.14963*.

Çelik, E. and Dalyan, T., 2023. Unified benchmark for zero-shot Turkish text classification. *Information Processing & Management,* 60**,** 103298.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N. and Androutsopoulos, I., 2020. LEGAL-BERT: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P. and Androutsopoulos, I., 2019. Large-scale multi-label text classification on EU legislation. *arXiv preprint arXiv:1906.02192*.

Chan, B., Schweter, S. and Möller, T., 2020. German's next language model. *arXiv preprint arXiv:2010.10906*.

Chowdhury, G. G., 2010. *Introduction to modern information retrieval*, Facet publishing.

Clark, K., Luong, M.-T., Le, Q. V. and Manning, C. D., 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Coban, O., Yağanoğlu, M. and Bozkurt, F., 2023. Domain Effect Investigation for Bert Models Fine-Tuned on Different Text Categorization Tasks. *Arabian Journal for Science and Engineering*, 1-18.

Çöltekin, Ç., Doğruöz, A. S. and Çetinoğlu, Ö., 2023. Resources for Turkish natural language processing: A critical survey. *Language Resources and Evaluation,* 57, 449-488.

Darji, H., Mitrović, J. and Granitzer, M., 2023. German BERT model for legal named entity recognition. *arXiv preprint arXiv:2303.05388*.

De Carvalho, A. C. and Freitas, A. A., 2009. A tutorial on multi-label classification techniques. *Foundations of Computational Intelligence Volume 5: Function Approximation and Classification*, 177-195.

De Vries, W., Van Cranenburgh, A., Bisazza, A., Caselli, T., Van Noord, G. and Nissim, M., 2019. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*.

Devlin, J., Chang, M. W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Di Liello, L., Gabburo, M. and Moschitti, A., 2021. Efficient pre-training objectives for transformers. *arXiv preprint arXiv:2104.09694*.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S. M. and Weischedel, R. M., 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, 2(1), 837-840.

Dumitrescu, S. D., Avram, A. M. and Pyysalo, S., 2020. The birth of Romanian BERT. *arXiv preprint arXiv:2009.08712*.

Öztürk, C. E., 2023. *Retrieving turkish prior legal cases with deep learning,* Thesis (Master), Bilkent University.

Er, M. J., Venkatesan, R. and Wang, N., 2016. An online universal classifier for binary, multi-class and multi-label classification. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 003701-003706.

Eronen, J., Ptaszynski, M. and Masui, F., 2023. Zero-shot cross-lingual transfer language selection using linguistic similarity, *Information Processing & Management*, 60, 103250.

Farahani, M., Gharachorloo, M., Farahani, M. and Manthouri, M., 2021. Parsbert: Transformer-based model for persian language understanding. *Neural Processing Letters*, 53, 3831-3847.

Gao, J. and Lin, C. Y., 2004. Introduction to the special issue on statistical language modeling. *ACM Transactions on Asian Language Information Processing (TALIP),* 3(2), 87-93.

Grandini, M., Bagli, E. and Visani, G., 2020. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.

Graves, A., 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
Grishman, R. and Sundheim, B. M., 1996. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.
Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J. and Poon, H., 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3**,** 1-23.
He, P., Gao, J. and Chen, W., 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.
He, P., Liu, X., Gao, J. and Chen, W., 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
Iter, D., Guu, K., Lansing, L. and Jurafsky, D., 2020. Pretraining with contrastive sentence objectives improves discourse performance of language models. *arXiv preprint arXiv:2005.10389*.
Jelinek, F., 1998. *Statistical methods for speech recognition*, MIT press.
Jiang, Z. H., Yu, W., Zhou, D., Chen, Y., Feng, J. and Yan, S., 2020. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33**,** 12837-12848.
Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F. and Liu, Q., 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.
Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L. and Levy, O., 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8**,** 64-77.
Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N. and Wu, Y., 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.
Kalyan, K. S., Rajasekharan, A. and Sangeetha, S., 2021. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*.
Kaplan, J., Mccandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J. and Amodei, D., 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
Kesgin, H. T., Yuce, M. K. and Amasyali, M. F., 2023. Developing and Evaluating Tiny to Medium-Sized Turkish BERT Models. *arXiv preprint arXiv:2307.14134*.
Khan, M. Z. 2021. Comparing the Performance of NLP Toolkits and Evaluation measures in Legal Tech. *arXiv preprint arXiv:2103.11792*.
Kitaev, N., Kaiser, Ł. and Levskaya, A., 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.
Köksal, A. and Özgür, A., 2021. Twitter dataset and evaluation of transformers for Turkish sentiment analysis. In *2021 29th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 1-4
Koloski, B., Pollak, S., Škrlj, B. and Martinc, M., 2021. Extending neural keyword extraction with TF-IDF tagset matching. *arXiv preprint arXiv:2102.00472*.
Kombrink, S., Mikolov, T., Karafiát, M. and Burget, L., 2011. Recurrent Neural Network Based Language Modeling in Meeting Recognition. In *Interspeech* , 11, 2877-2880.
Kuratov, Y. and Arkhipov, M., 2019. Adaptation of deep bidirectional multilingual transformers for Russian language. *arXiv preprint arXiv:1905.07213*.

Lample, G. and Conneau, A., 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. and Soricut, R., 2019. ALBERT: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H. and Kang, J., 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics,* 36**,** 1234-1240.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Li, J., Sun, A., Han, J. and Li, C., 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering,* 34**,** 50-70.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Deng, H. and Ju, Q., 2020. Fastbert: a self-distilling bert with adaptive inference time. *arXiv preprint arXiv:2004.02178*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Malmsten, M., Börjeson, L. and Haffenden, C., 2020. Playing with Words at the National Library of Sweden--Making a Swedish BERT. *arXiv preprint arXiv:2007.01658*.

Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y., Romary, L., De La Clergerie, É. V., Seddah, D. and Sagot, B., 2019. CamemBERT: a tasty French language model. *arXiv preprint arXiv:1911.03894*.

Masala, M., Iacob, R. C. A., Uban, A. S., Cidota, M., Velicu, H., Rebedea, T. and Popescu, M., 2021. jurBERT: A Romanian BERT model for legal judgement prediction. In *Proceedings of the Natural Legal Language Processing Workshop*, 86-94.

Masala, M., Ruseti, S. and Dascalu, M., 2020. Robert–a romanian bert model. In *Proceedings of the 28th International Conference on Computational Linguistics*, 6626-6637.

Mengusoglu, E. and Deroo, O., 2001. Turkish LVCSR: Database Preparation and Language Modeling for an Aglutinative Language. In *IEEE International Conference on Acoustics Speech And Signal Processing*, IEEE, 6, 4018-4018.

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

Milosheski, L., 2020. LaMBERT: Light and Multigranular BERT.

Nadkarni, P. M., Ohno-Machado, L. and Chapman, W. W., 2011. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18**,** 544-551.

Oflazer, K., 2014. Turkish and its challenges for language processing. *Language resources and evaluation,* 48**,** 639-653.

Oflazer, K. and Saraçlar, M., 2018. *Turkish natural language processing*, Springer.

Özer, H. and Korkmaz, E. E., 2022. Transmorph: a transformer based morphological disambiguator for Turkish. In *Turkish Journal of Electrical Engineering and Computer Sciences,* 30**,** 1897-1913.

Oztürk, Y., Thomas, I. and Gadjeva, S., 2023. Morphological Resources for the Study of Turkish Derived Nouns. In *Fourth International Workshop on Resources and Tools for Derivational Morphology,* 89.

Panda, S., Agrawal, A., HA, J. and Bloch, B., 2021. Shuffled-token detection for refining pre-trained roberta. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, 88-93.

Park, S., Moon, J., Kim, S., Cho, W. I., Han, J., Park, J., Song, C., Kim, J., Song, Y. and Oh, T., 2021. Klue: Korean language understanding evaluation. *arXiv preprint arXiv:2105.09680*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L., 2018. Deep contextualized word representations. arXiv preprint. *arXiv preprint arXiv:1802.05365*.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N. and Huang, X., 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences,* 63**,** 1872-1897.

Rachman, F. H., 2020. Twitter sentiment analysis of Covid-19 using term weighting TF-IDF and logistic regresion. In *6th Information Technology International Seminar (ITIS)*, IEEE, 238-242.

Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I., 2018. Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8)**,** 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P. J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research,* 21**,** 5485-5551.

Ravichandiran, S., 2021. *Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT*, Packt Publishing Ltd.

Ryu, M., 2021. [RE] ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.

Sang, E. F. And De Meulder, F., 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Sanh, V., Debut, L., Chaumond, J. and Wolf, T., 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schweter, S., 2020. BERTurk-BERT models for Turkish. *Zenodo*, 3770924.

Shmidman, S., Shmidman, A. and Koppel, M., 2023. DictaBERT: A State-of-the-Art BERT Suite for Modern Hebrew. *arXiv preprint arXiv:2308.16687*.

Spärck Jones, K., 2004. IDF term weighting and IR research lessons. *Journal of documentation,* 60**,** 521-523.

Sun, P., Wang, L. and Xia, Q., 2017. The keyword extraction of Chinese medical web page based on WF-TF-IDF algorithm. In *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, IEEE, 193-198.

Sun, S., Cheng, Y., Gan, Z. and Liu, J., 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y. and Zhou, D., 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.

Tang, B., Cao, H., Wu, Y., Jiang, M. and Xu, H., 2013. Recognizing clinical entities in hospital discharge summaries using Structural Support Vector Machines with word representation features. In *BMC medical informatics and decision making*, BioMed Central, 1-10.

Taylor, W. L., 1953. "Cloze procedure": A new tool for measuring readability. *Journalism quarterly,* 30**,** 415-433.

Türkmen, H., Dikenelli, O., Eraslan, C., Çalli, M. C. and Özbek, S. S., 2023a. BioBERTurk: Exploring Turkish Biomedical Language Model Development Strategies in Low-Resource Setting. *Journal of Healthcare Informatics Research***,** 1-14.

Türkmen, H., Dikenelli, O., Eraslan, C., Çalli, M. C. and Özbek, S. S., 2022. Developing Pretrained Language Models for Turkish Biomedical Domain. *IEEE 10th International Conference on Healthcare Informatics (ICHI)*, IEEE, 597-598.

Türkmen, H., Dikenelli, O., Eraslan, C., Çallı, M. C. and Özbek, S. S., 2023b. Harnessing the Power of BERT in the Turkish Clinical Domain: Pretraining Approaches for Limited Data Scenarios. *arXiv preprint arXiv:2305.03788*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., Ginter, F. and Pyysalo, S., 2019. Multilingual is not enough: BERT for Finnish. *arXiv preprint arXiv:1912.07076*.

Wang, W., Bi, B., Yan, M., Wu, C., Bao, Z., Xia, J., Peng, L. and Si, L., 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N. and Zhou, M., 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33**,** 5776-5788.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D. and Metzler, D., 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Yadav, V. and Bethard, S., 2019. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*.

Yan, A., Mcauley, J., Lu, X., Du, J., Chang, E. Y., Gentili, A. and Hsu, C. N., 2022. RadBERT: Adapting transformer-based language models to radiology. *Radiology: Artificial Intelligence*, 4**,** e210258.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R. and Le, Q. V., 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems,* 32.

Yao, L., Pengzhou, Z. and Chi, Z., 2019. Research on news keyword extraction technology based on TF-IDF and TextRank. In *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, IEEE, 452-455.

Yeung, C. M., 2019. *Effects of inserting domain vocabulary and fine-tuning BERT for German legal language.* University of Twente.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q. and Yang, L., 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33**,** 17283-17297.

Zhang, J., Zhao, Y., Saleh, M. and Liu, P., 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, PMLR, 11328-11339.

Zhang, M. L. and Zhou, Z. H., 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition*, 40**,** 2038-2048.

Zhang, Y., 2016. *Integrating Encyclopedic Knowledge into Neural Network Language Models.* National Research Center.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J. and Dong, Z., 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q. and He, L., 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.

Zhuohao, W., Dong, W. and Qing, L., 2021. Keyword Extraction from Scientific Research Projects Based on SRP-TF-IDF. *Chinese Journal of Electronics*, 30**,** 652-657.

CURRICULUM VITAE

Personal Information	
Name Surname	FARNAZ ZEIDI
Place of Birth	
Date of Birth	Tarih girmek için tıklayın veya dokunun.
Nationality	❒ T.C. ☑ Other:
E-Mail Adress	
Web Page	

Education Information Undergraduate	
University	Sheikhbahaee University
Faculty	Faculty of Engineering
Department	Department of Information Technology
Year of Graduation	30.09.2010

Masters	
University	Mazandaran University of Science and Technology
Name of Institute	Computer Engineering Institute
Department	Department of Information Technology
Program	Information Technology Engineering

Ph.D.	
University	İstanbul Üniversitesi
Name of Institute	Fen Bilimleri Enstitüsü
Department	Department of Informatics
Program	Programme of Informatics

Articles and Publications

International Academic Papers:

Zeidi, F., Azar, L., Arslan, V. and Erol, Ç., 2022. A Hybrid Model Focusing on Data Pre-Processing in Diabetes Diagnosis. *Cybernetics and Systems*, 54(7), 1199-1211.

Zeidi, F. and Sanayei, A., 2015. Prioritization of Critical Success Factors in ERPII Implementation: Case Study in Iran. *International Journal of Information Science & Management*, 13(SpecialIssue), 1_15.

Zeidi, F. and Babaheidari, M. M., 2015. Prioritization of critical success factors in knowledge-management using the AHP method. *Cumhuriyet Üniversitesi Fen Edebiyat Fakültesi Fen Bilimleri Dergisi*, 36(3), 1813-1820.

National Conference Papers in Iran:

Zeidi, F., 2015. Investigating Critical Success Factors for Software Projects: An Interpretive Structural Modeling Approach, *International Conference on Management, Culture and Economic Development*.

Babaheidari, M. M., and **Zeidi, F**., 2015. Development of a model for assessing the maturity of customer knowledge management in organizations, *International Conference on Management, Culture, and Economic Development*.

Sanayei, A., Shirazi, B. and **Zeidi, F.**, 2012. Integration model of E-commerce and ERP based on Web Services, *National Conference on Information Technology and Economic Jahad*, 356-364.

Sanayei, A., Shirazi, B. and **Zeidi, F.**, 2012. Integration approach of ERP and E-commerce, *2nd National conference on Soft Computing and Information Technology*.

Zeidi, F., Alikia, A. and Shirazi, B., 2012. Integration model of ERP, CRM and SCM in E-commerce Environment, *Second National Conference on Information and Communication Technology*.

Tashakkorian, M., **Zeidi, F.** and Momeni, H., 2011. Offering a weighted graph-based for Test Case Prioritization in web services regression testing, *First CSUT Conference on Computer, Communication, Information Technology* , 88-95 .

Books:

Zeidi, F. and Azar, L., 2023. *A Study of Students' Opinions About Media Platforms and Distance Education in the Covid-19 Pandemic*, in: Gulsecen, S., Sharma, S.K. and Kocoglu, F.O. (Eds.), Pandemic and the Critical Role of Knowledge Management. Istanbul University Press, pp. 159-178.

Taşci, F., Erdoğan, S., Erken, M., Erdoğan, G., **Zeidi, F.** and Gülseçen, S., 2022. *Adaptation of International Students to Turkish Language: a Case Study of Istanbul University (Uluslararası Öğrencilerin Türk Diline Uyumları: İstanbul Üniversitesi Örneği)*, Harf publication. (In Turkey)

Sanayei, A., **Zeidi, F.** and Karimiyan, M., 2015. *Systems Integration ERP, CRM and SCM with focus on E-commerce (ERPII)*, Jahad daneshgahi publication. (In Iran)