COMPARATIVE ANALYSIS OF NLP MODELS FOR ICD AND ATC CLASSIFICATION IN TURKISH MEDICAL DATASETS

(TURKCE TIBBI VERISETLERINDE ICD VE ATC SINIFLANDIRMALARI ICIN NLP MODELLERININ KARSILASTIRMALI ANALIZI)

by

D a m l a B u s r a O z s o n m e z

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

November 2023

TABLE OF CONTENTS

LIST OF SYMBOLS

ICD : International Classification of Diseases

ATC : Anatomical Therapeutic Chemical

WHO : World Health Organization

NLP : Natural Language Processing

OCR : Optical Character Recognition

BERT : Bidirectional Encoder Representations from Transformers

IR: Information Retrieval

HTTP : Hypertext Transfer Protocol

API : Application Programming Interface

SVC : Support Vector Classification

RBF : Radial Basis Function

TPE : Tree-structured Parzen Estimator

RegEx : Regular Expression

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This study incorporates various methodologies to classify the Turkish medical texts with the aim of identifying ICD (International Classification of Diseases) & ATC (Anatomical Therapeutic Chemical) codes, leveraging various NLP approaches. The primary challenge involves comprehending the context and accurately categorizing medium to long documents into correct classes. Another challenge encompasses the acquisition of a labeled dataset with all categories, given the limited data resources in Turkish, which is a low resource language. Two distinct datasets are acquired for this study. The first dataset, which focuses on specific ICD-CM-10 C-Types, consists of medical summaries in Turkish language and was externally sourced. The second dataset, which is a new dataset including drug manuals covering all ATC types in Turkish is curated. After dataset creation, text processing has been implemented to obtain better performance results. SVC (Support Vector Classifier), FastText and various BERT models, such as BERTurk Uncased, BERTurk Large, ConvBERTurk, ElectraBERTurk and DistilBERTurk are used to classify the documents. Hyperparameter tuning is also applied to harness the potential of BERT and FastText, resulting in a notable 86% overall F1-Score on the Turkish medical summaries dataset and DistilBERTurk, the Turkish version of DistilBERT, resulted in 92.3% overall F1-Score on the same dataset. The most notable accomplishment, however, was for the Turkish Drug Manuals ATC code dataset, where the performance results, 96% F1-score using FastText, 95.9% F1-score using BERTurk Uncased and 94% using SVC models are obtained. These results highlight the effectiveness of the used approaches in Turkish text classification.

ÖZET

Bu çalışma, çeşitli NLP yaklaşımlarından yararlanarak ICD (Uluslararası Hastalık Sınıflandırması) ve ATC (Anatomik Terapötik Kimyasal) kodlarını belirlemek amacıyla Türkçe tıp metinlerini sınıflandırmak için çeşitli metodolojileri içermektedir. Bu calismadaki baslica kritik nokta medikal metinlerdeki bağlamın anlaşılması ve orta/uzun boyutlu belgelerin dogru siniflandirilmasidir. Diğer bir zorluk ise sinirli veri kaynaklarina sahip olan Turkce gibi dusuk kaynakli bir dilde olasi butun kategorilere sahip etiketli bir veri seti elde etmektir. Bu calisma icin iki farkli veri seti kullanilmistir. Ilk veri seti tibbi makale ozetlerinin spesifik olarak ICD-CM-10 C kodlu hastalik siniflari ile etiketlendirilmistir, Bu veri seti baska bir ekip tarafindan hazirlanmis ve dis kaynaklardan temin edilmistir. Ikinci veri seti ise olasi butun ATC kodlarini kapsayan Turkce Ilac kilavuzlarini (Kisa Urun Bilgisi dokumanlari) icermektedir. Veri seti olusturma adimindan sonra daha iyi performans sonuclari elde etmek icin metin onisleme uygulandi. BERTurk Uncased, BERTurk Large, ElectraBERTurk and DistilBERTurk gibi bircok farkli BERT modelleri, SVC ve FastText dahil olmak uzere cesitli modeller her iki veri seti uzerinde uygulandi ve Turkce Medikal Ozetler Verisetinde %86 lik bir F1-skoru elde edildi. DistilBERT'in Turkce versiyonu olan DistilBERTurk modeli ise bu veri setinde %92.3 F1-Skoru elde edilmesini saglamistir. En yuksek basari sonuclari ise Turkce Ilac Kullanim Kilavuzu veri seti uzerinde elde edildi. FastText ile %96, BERTurk Uncased modeli ile %95.9 F1-skoru ve SVC ile %94 F1 skoru elde edilmistir. Bu sonuclar her iki modelin de yontem olarak etkili oldugunu vurgulamaktadir.

1. INTRODUCTION

Finding solutions to existing problems or bringing existing solutions to a more optimal point is only possible with data insight, which undeniably shows how knowledge extraction is a necessity in a variety of fields. The competition between organizations brought about the necessity of continuous increase in efficiency and productivity, resulting in an increase in automation in all types of business lines.

Information Retrieval (IR) is a field of study to facilitate the knowledge extraction by being applied to many data types such as image, video, text and sound. IR is the system which translates the data into the information and thus focuses on retrieving relevant documents. Natural Language Processing (NLP) frequently plays an important role for information extraction since it aims to understand the hidden relations and entities in the text data, which helps to increase automation and efficiency in IR systems. Furthermore, NLP has integrated into the daily lives thanks to its broader impact. It has many uses in our daily life such as chatbots, search engines, language translation, text summarization and spell checking for the textual data resources from social media (Wong, Li, & Wong, 2016), education (Chen et al., 2020) and finance (Liu et al., 2021). Moreover, the application of NLP can be found in medicine.

Medicine is undeniably one of the most critical areas of our lives, as it is of paramount importance to the health of living beings. It also covers a non-negligeable part (11%) of global GDP (World Health Organization, 2022). Medicine covers not only the areas such as the provision of medical services, pharmaceutical industry, medical equipment but also provider and payer (“Sub-Sectors in the Health Care Industry,” n.d.). The provider involves the organizations and professionals who deliver and administers a health care service. Hospitals, clinics, doctors, pharmacists, and the other healthcare professionals are included in this sub-sector whereas government and insurance companies are the

payers since they are entitled to cover the expenses on behalf of patients. Patient is the center of healthcare.The collaboration between providers and payers is pivotal to ensure patient care. For the patient to receive high-quality care, this collaboration must be optimized in the healthcare ecosystem. ATC (Anatomical Therapeutic Chemical) and ICD (International Classification of Diseases) codes play integral roles in this system. ATC classification system is a system for drug categorization based on drug's therapeutic and pharmacological properties, whereas ICD is a system for disease classification which categorizes diseases according to their symptoms. Thus, ATC codes are mostly used by pharmacists and ICD codes are mostly used by hospitals.

ICD coding system is created by WHO (World Health Organization) and it provides world-wide statistics for mortality, death causes and diseases. The governments and many insurance companies have mandated the inclusion of ICD codes in prescriptions and medical procedures to facilitate healthcare management, as well as disease monitoring.

Pharmaceutical companies prioritize information extraction for purposes such as drug development and discovery, following market trends, data integration for EHR (Electronic Health Records) and regulatory compliance. Most pharmaceutical companies have not yet been able to fully automate these processes; many tasks such as document labeling are still being carried out manually by employees. However, in long term, training employees in domain-specific tasks costs time and money. In brief, it will be more efficient to automate tasks such as document categorization and labeling whenever possible.

This study aims to employ ATC classification on Turkish medical drug instructions dataset. The findings from our prior research, ICD classification on the Turkish medical summaries' dataset, are also incorporated into this study. As there are more than two different categories for each dataset, multiclass classification will be the primary focus of this study. Each category will be assigned to a document, whether it is a medical summary or a drug usage manual. The medical summary dataset has been created by another research team, while the drug usage manual has been generated within the scope of this study. The drug manuals are downloaded in PDF format from the website of Ministry of

Health and the labels are generated via Web Scraping from an online resource. The PDF formats are converted to the plain text format (TXT) via text-based PDF converter and scanned image-based converter. As there should be enough data for meaningful analysis and model training, datasets are filtered to ensure that the number of documents in a specific category exceeds 20. Different BERT (Bi-directional Encoder Representations from Transformers) architectures are used to finetune the Turkish BERT models. As per the base line, Fast Text is used since it is known as more robust to the words not existing in vocabulary and it can handle large text corpora. preprocessing steps are applied to both the datasets to obtain better performance results (F1-score).

The paper is structured as follows: the Literature Review section describes the recent studies made in this research area, the Proposed Approach section explains the techniques used for dataset collection, data & text processing and models used for this study. Furthermore, the performance measurement and results will be analyzed in Experiments & Results section. The paper will be finalized with Conclusion section where the results will be discussed.

2. LITERATURE REVIEW

Information extraction from text data has a long history that dates to many years ago. Edmundson and Wyllys (1961) conducted a study on automatic abstracting and indexing, which involved experiments relying on utilization of automatic scanners capable of reading text and transmitting word by word to a computer (Edmundson & Wyllys, 1961). They sorted the words by their frequency, which was also used to select the key sentences to print out as automatic abstracts. In addition to this measure, they also aimed to explore ways to assign weight to the importance of words. They developed their own rules for word selection such as the word's position (e.g., whether it is in the title or within the first paragraph) and its alignment with the domain. While this work may not resemble today's classification tasks, it has laid the groundwork for the techniques currently in use. A year later, Vickery (1962) started a discussion on the concept of cross-classification, also known as multi-classification, by emphasizing that an entity could be classified in different ways within various categories (Vickery. 1962). He explained that a rabbit can be classified as a fur-bearing animal, a rodent, an herbivore and thus the cross-classification must be adaptable to accommodate the interests of users. This criterion still remains one of the fundamental principles of contemporary text classification.

In this section, the studies related to ATC & ICD classification in the medicine area. ATC classification is a standard to match drugs with the relevant system in the body, as well as the chemical substances. As they are both medical codes, they can also be used for conducting statistical analyses together. Kırmızı et al. (2017) investigated the statistical analysis of Dentists' antibacterial-containing prescriptions in Turkiye by matching ICD-10 categories with ATC Level-3 and Level-5 categories for "J01", which is an ATC level-2 code. As they obtained indication – drug category information, they concluded that the most used drug was *Amoxilin + Enzyme Inhibitor*, having the ATC code *J01CR02*; for the ICD codes *K01, K02* (usage rate > .70). Weibrecht, Endel, & Zechmeister employed

Word2Vec model to predict the probable ICD codes based on ATC codes for the Austrian Health data encompassing hospital and sick leave diagnoses (2019).

Croset et al. implemented a mapping between drugs and ATC codes and then converted those into a Web Ontology Language (OWL) representation. By integrating DrugBank and ATC codes, they also enabled question-answering with the expressions involving the ATC code and the drug compound.

Iorio et al. (2010) conducted research on the drug repurposing based on ATC groups where they found a significant relationship between drugs sharing the same Mode of Action (MoA) and those grouped based on ATC codes. KEGG (Kyoto Encyclopedia of Genes & Genomes) dataset is used to predict ATC codes via the chemical-chemical interactions and chemical-chemical similarities (Chen et al., 2012). They achieved maximum 78.49% accuracy with similarity-based approach and 67.67% with interaction-based approach for ATC Level 1 codes while obtaining accuracy values ranging from 18% to 0% for higher level ATC codes (e.g. level 2,3,4 and 5).

Electronic Health Records (EHR) are also good resources for medical text classification since they contain a wide range of information related to medical history for the patients. Laboratory results and imaging reports are the most used EHRs and thus they are the most heavily analyzed EHRs. Yet, various studies conducted on text based EHRs. A study has been conducted on the extraction of ATC codes from the German drug prescriptions, which contain both structured and unstructured data, using a combination of techniques such as Regular Expressions (RegEx) and word similarities applied on drug ingredients (Reinecke et al., 2023).

Automation is important in many industries since there is more need for efficiency and productivity more than ever. NLP is a great facilitator for automating manual and repetitive tasks such as information extraction and data processing. Gnjidic et al. (2015) conducted a study that compared the accuracy of manually extracting ATC codes from medical texts with the automation extraction of ATC codes. The automation process involved extracting medications and medication components from free-text data of Excel

files. The components have been later used to match ATC codes with the help of a database containing ATC index (Gnjidic et al., 2015). Finally, it has been concluded that the automated labeling shows high levels of sensitivity (83.8% for 3-digit ATC level) compared to the manual labeling; yet the authors of the study suggest that automated methods have the potential to be useful for large-scale medication-related research.

Various machine learning and deep learning methods have been applied for ATC multi-label classification. Cheng et al. used a *ML-GKR* (Multi-Label Gaussian Kernel Regression) classifier to establish a hybrid ATC classifier *iATC-mHyb* (2017) by using *ChEBI* database. SVM (Support Vector Machines), in combination with the *RAKEL* (Random k-Labelsets) algorithm is applied to graph network data (Zhou et al., 2020). A fusion of LSTM networks and multiple linear regression (*hMuLab*) classifiers outperformed the single LSTM classifier and the single *hMuLab* model (Nanni, Brahnam, & Maguolo, 2020).

As with the ATC classification, numerous studies can be found related to the ICD classification. *MIMIC-III* (Medical Information Mart for Intensive Care III) dataset is used to classify discharge summaries with ICD-10 Codes (Hsu, Chang, & Chang, 2020). They performed extensive text processing techniques such as removing all the punctuation, word breakers and stop words to create TF-IDF and Word2Vec vectors. They compared SVM, CNN, LSTM and HAN (Hierarchical Attention Network) models and obtained the best result with CNN model as 76% micro-F1.

PubMedBERT was the first Automatic ICD coding study with a transformer model (Pascual et al., 2021). As evident from its name, the model is pretrained on PubMed text (3.1 billion words, 21GB) containing information about discharge summaries (medical notes created at the end of a stay in a medical facility). As the documents were long, they applied three strategies to avoid memory issues while finetuning: First 512 tokens from the front part, first 512 tokens from the last tokens and first 256 tokens from the front part and 256 tokens (mixed strategy) from the last part of the summaries. They obtained the best result as 88.65% AUC Score for the data following the mixed strategy. Blanco et al. (2020) studied Spanish EHR data by emerging contextual embedding vectors (ELMo &

FastText) to create RNN, Feed Forward Neural Network, Neural-Net Language Model (NNLM) as the text representation layers. They obtained 63% F-score by using Bi-GRU with the combination of Contextual ELMo embeddings.

ICD Classification has also been studied using Turkish medical texts. A recommender system to semi-automatically assign top-k ICD codes to free-text patient records (EHR) is presented (Arifoğlu et al., 2014). They used Bag-of-Words vectors and Lucene scoring calculating TF-IDF & Cosine Similarity as well as Borda count. Çelikten & Bulut developed a Turkish Medical Summaries Dataset and they pioneered employing BERT models on this dataset (2021). Ozsonmez & Acarman compared the performance of FastText and BERT and they observed a slight difference (4.5% F1-Score) between FastText & BERT and their conclusion suggests that FastText could serve as a practical classifier as well (2023).

3. PROPOSED APPROACH

In this section, the overview begins with the explanation of ICD codes and the used datasets. Web Scraping is performed to collect drug manuals files in pdf format. As models cannot directly process data in PDF format, file conversion to plain text format (TXT) is realized using a Python library, which converts PDFs not containing scanned images, and Tesseract OCR to convert the PDF files having scanned images. After the file conversion, the dataset creation step is performed, and the FastText & BERT models are utilized to obtain the performance results.

3.1 ICD & ATC Codes Overview

As various languages are employed in healthcare industry, standardized methods are needed to categorize documents for healthcare institutions. ATC & ICD codes serve this crucial purpose by enabling quality monitoring. In this section, the explanations for the utilized codes will be provided.

3.1.1 ICD Codes Overview

The ICD Coding is a designed system to provide standardization on causes and consequences of human diseases & death worldwide (World Health Organization, n.d.). ICD is mainly used for disease or death statistics such as primary, secondary, tertiary care and death certifications. The ICD codes are specifically used by healthcare providers, insurance companies and government for clinical and administrative functions. Their significance in Türkiye also aligns with their global purposes.

In this study, only ICD-CM-10 C Code data is used. The dataset is explained with more details in Section 3.2.1. Table 3.1 shows the corresponding location for ICD-CM-10 C Category, which involves “Malignant Neoplasm”.

Table 3.1: ICD-CM-10 C Types

ICD Code	Corresponding Location for Malignant Neoplasm
C01	Base of Tongue
C02	Other & Unspecified Parts of Tongue
C03	Gum
C04	Floor of Mouth
C05	Palate
C06	Other & Unspecified Parts of Mouth
C07	Parotid Gland
C08	Other & Unspecified Major Salivary Glands
C10	Oropharynx
C14	Other & Ill-Defined Sites in the Lip, Oral Cavity & Pharynx
C23	Gallbladder

3.1.2 ATC Codes Overview

ATC code is a unique code assigned to a medicine based on the anatomical or pharmacological group and it can be categorized at five different levels (World Health Organization, n.d.). Those levels are:

- **ATC 1st level**: Anatomical & Pharmacological groups
- **ATC 2nd level**: Pharmacological & Therapeutic subgroups
- **ATC 3rd & 4th level**: Chemical, Pharmacological or Therapeutic subgroups
- **ATC 5 level**: Chemical substance subgroups

In this study the documents will be categorized by making use of the second level of ATC codes, such as *R01*, even though there are 5 different levels available. 69 different ATC codes are used as labels. Table 3.1 explains the first level ATC codes and Appendix B. explains all the ATC codes that are selected for this study.

Table 3.2: Corresponding Anatomical Groups for 1st Level ATC Codes

ATC 1st Level	Anatomical Groups
A	**A**limentary Tract & Metabolism
B	Blood & Blood Forming Organs
C	Cardiovascular System
D	Dermatologicals
G	Genito Urinary System & Sex Hormones
H	Systematic Hormonal Preparations (Excl. Sex Hormones & Insulins)
J	Antiinfectives For Systemic Use

L	Antineoplastic & Immunomodulating Agents
M	Musculo-Skeletal System
N	Nervous System
P	Antiparasitic Products, Insecticides & Repellents
R	Respiratory System
S	Sensory Organs
V	Various

3.2 Datasets

In this section, the utilized datasets and their respective class distributions will be explained. As shown in Figure 3.1 and in Figure 3.4, there is a class imbalance in both datasets. No specific adjustments are made to address dataset imbalances since a suitable metric such as F1-score is used. FastText and BERT models are both capable of handling the imbalanced data with the correct metrics such as F1-score, which balances between precision and recall (Madabushi, Kochkina, & Castelle, 2020). FastText is capable of imbalanced data handling due to the Huffman trees; which results in having less depth for small number of classes and having more depth for large number of classes [20].

3.2.1 Turkish Medical Summary Dataset

Turkish medical summaries dataset contains the medical data, spanning from 1976 to 2014, which follows a similar structure to that of the OHSUMED dataset. It has been meticulously compiled by Azer Celikten and Hasan Bulut, who dedicated significant efforts (2021). The original dataset contains 1575 files and 24 International Classification of Diseases (ICD) Codes (C01-C23). Nevertheless, since some of the ICD Codes contain very little data (< 30); they are exempted from the training and testing steps in this study. Figure 3.1 shows the selected categories and the class distribution within the dataset.

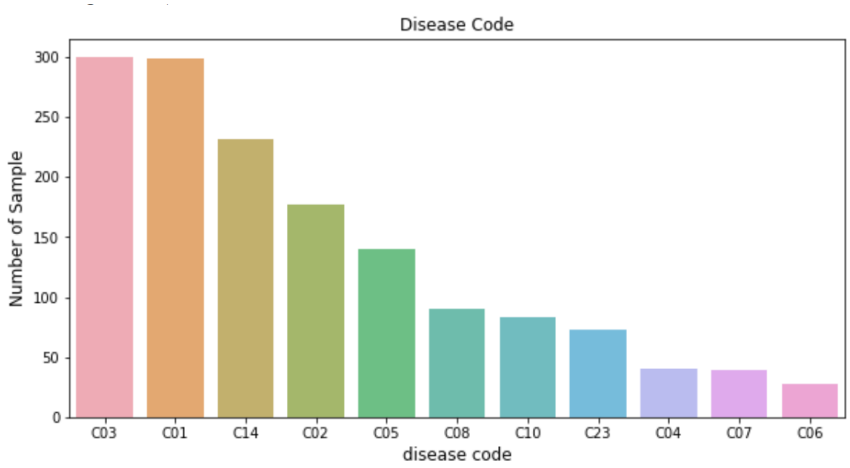

Figure 3.1: Class Distribution for each ICD Type

3.2.2 Turkish Drug Manuals Dataset

In this section, the steps for data collection and dataset creation are covered.

3.2.2.1 PDF File Acquisition and TXT Conversion

Turkish drug manuals dataset is obtained by Turkish Ministry of Health website[1]. The HTML table is obtained, and all the PDF documents are saved under "*Kub Dokumani*". 14228 PDF files are obtained via HTTP request.

PDF files are converted to plain text (TXT) via PyMuPDF (Fitz as its older name) which facilitates document manipulation, especially for Text-Based PDFs. It iterates through each page and each page is converted to text data. As the result, all the text data for each page are gathered for a single document. Yet, as its limitation, PyMuPDF generates TXT files smaller than 50 bytes (containing either empty txt file or only a few ASCII characters) for Scanned Image PDFs, which indicates that OCR (Optical Character Recognition) must be employed for converting those files to prevent data loss as much as possible. Each page of the PDF files is converted to image, with 300 dpi (Dots Per Inch), which is an indicator for image resolution. After this step, OCR is applied to the generated

[1] https://www.titck.gov.tr/kubkt

images to extract text and the complete text is obtained by appending all the text from each page.

The generic flowchart is shown in Figure 3.2.

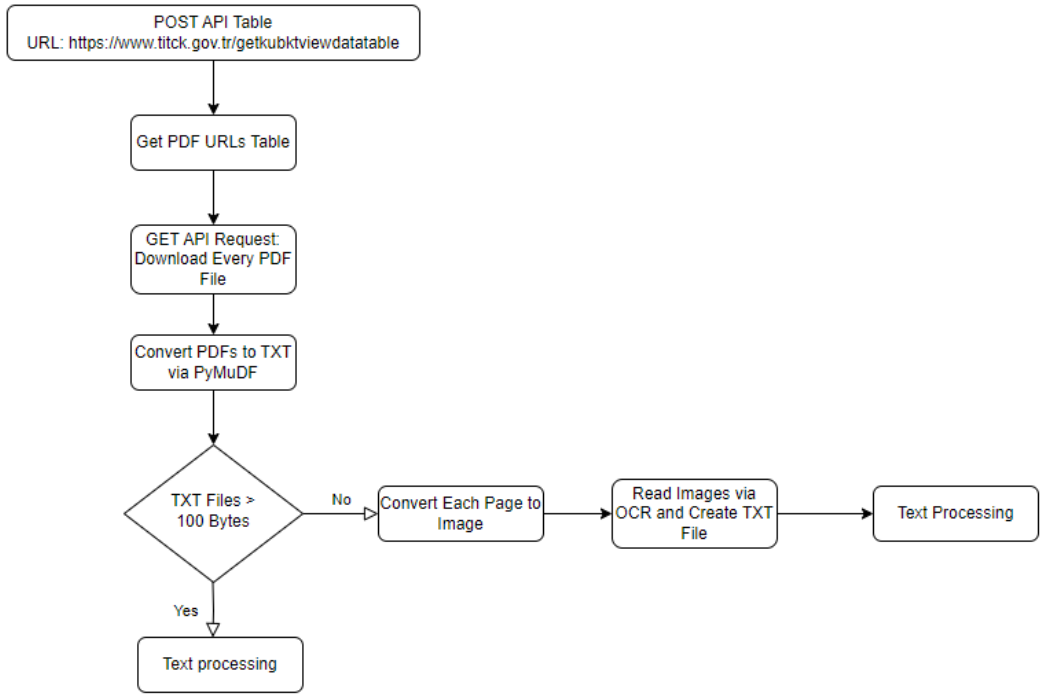

Figure 3.2: Flowchart for Turkish Drug Manuals Dataset Collection

3.2.2.2 Labeling Documents: ATC Code Assignment

As not all documents contained the ATC code within their text, the website *ilacabak.com* is utilized as a source to obtain the labels (Ilacabak, 2023). A Scraper is developed for this specific purpose. The scraper involved searching the drug names into the site's search box and retrieving the HTML content of the first search result. The necessary ATC code is thus extracted within the retrieved content from a specific field called *SB.atc,* which means "Ministry of Health ATC". As the intended categorization is ATC 2nd level, the first 3 characters are preserved.

RENNIE 100 ML SÜSPANSİYON

KT:	Kullanma talimatı için tıklayınız (Hastalar için)
KUB:	Kısa ürün bilgisi için tıklayınız (Hekimler için)
Etkin madde:	**antiasit kombinasyonlar**
ATC:	ATC SINIFLAMASI - A - SİNDİRİM SİSTEMİ VE METABOLİZMA A02 MİDE A02A ANTİASİTLER A02AX Antiasit, diğer kombinasyonlar **A02AXXX** antiasit kombinasyonlar
SB.atc:	A02AD01
Reçete Durumu:	Reçetesiz ile satılır.
İlaç Firması:	BAYER TÜRK KİMYA SAN. LTD. ŞTİ.
Barkod :	8699546705757
Satış Fiyatı:	**49.80 ₺**

Sitemizde ilaç satışı yapılmamaktadır.
İlaç fiyatlarının belirtilmesi Akılcı İlaç Kullanımına katkı amaçlıdır.
İlaç fiyatları TC Sağlık Bakanlığı Türkiye İlaç ve Tıbbi Cihaz Kurumu (TİTCK) tarafından haftalık olarak yayınlanan listelerden alınmıştır.
Belirtilen ilaç fiyatı eczaneler için önerilen satış fiyatı olup değişkenlik gösterebilir.
Fiyatlar güncellenmemiş olabilir.

Son Güncelleme:	23 Eylül 2023

Figure 3.3: Example Web Content for Label Extraction

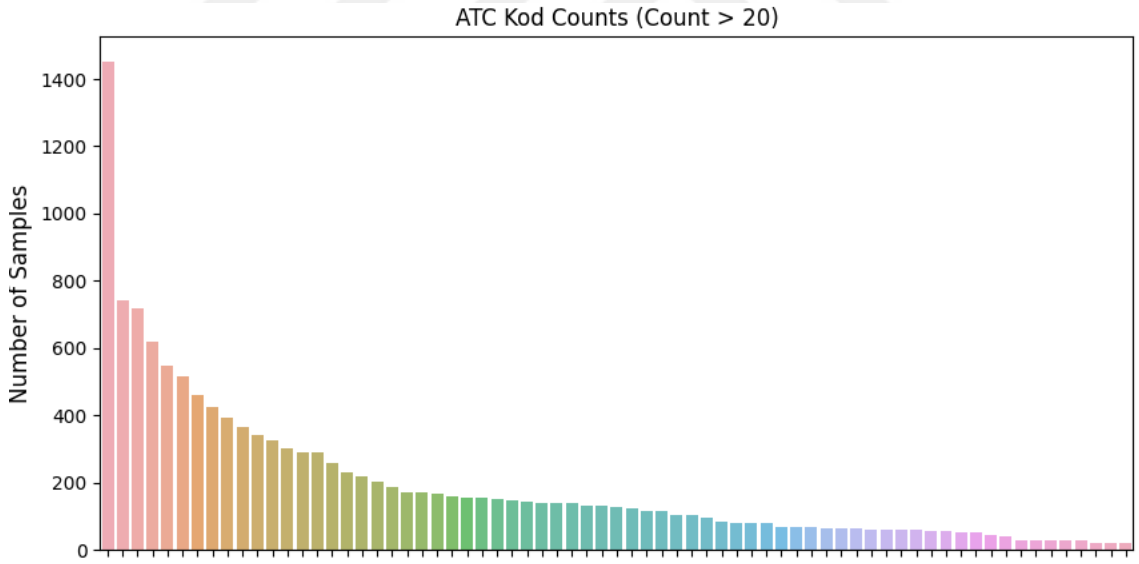

Figure 3.4: ATC Distribution Across the Turkish Drug Manuals Dataset

3.3. Data Preprocessing

In this section, the preprocessing steps applied to both datasets are discussed. Since the main dataset covered by this study is ATC dataset, further preprocessing steps are performed for this dataset. Performance results among different preprocessed ATC datasets are explained in detail in the Experiments and Results Section; in addition, it is

stated in the Conclusion section how much different preprocessing methods affect the result.

3.3.1 Preprocessing for ICD Dataset

In certain instances, the original text files exhibit typographical errors and non-ASCII characters. To rectify this, Regular Expressions (RegEx) is used for cleaning those characters. It is important to note that we do not undertake stop-word removal and punctuation elimination, as both FastText and BERT are equipped to manage these scenarios. *BERTurk Uncase* Tokenizer will be used to perform text tokenization, which includes converting the text to lowercase (Hugging Face, 2023).

3.3.2 Preprocessing Steps for ATC Dataset

After converting PDF files to plain-text TXT files, the text files are inspected and any sections containing unintelligible text (due to OCR) are deleted for the sake of clarity and coherence. All alphanumeric characters in the text are removed specific to FastText since very low performance score (10%) is obtained when this step is not performed.

The first 512 tokens are selected from each document as this text segment contains critical details including the drug's active and inactive ingredients (the chemical information), indications (diseases), the targeted system or organ(s) in addition to the suggested doses and the instructions for use. This selection is based on the fact that the ATC code, which is already a categorization system by the targeted organ/system and its chemical structure, further complements this data.

In addition, the following steps are performed on the selected 512 tokens for *Count Vectors* used in the machine learning model SVM to reduce dimensionality and noise by removing the variations of words:

- Converting text to lowercase: Commonly used lowercasing method in English is updated to convert the Turkish character "I" to "ı" rather than "i"
- Stop words removal: Turkish stop words list is obtained from Zemberek (Akin,

2017) and the stop words are removed from the texts.

- Lemmatization and Stemming:
    - Stemming is a technique used to reduce words to their unchanged root form, by removing suffixes or prefixes, which does not guarantee providing a valid word (*bileşim-> bileş, etkin -> etk)*
    - Lemmatization is another word reducing technique that converts word into their base forms, guaranteeing a semantically meaningful result (*bilgisi -> bilgi*, *içerir -> içermek*)
    - They are applied separately applied to the text to analyze the impact of those two methods on the performance.

3.4. Feature Extraction

Count Vectors are employed as the features for the SVM which is the baseline model of this study as they are computationally less intensive compared to TF-IDF vectors.

3.5. Models

In this section, the foundational principles and distinctive characteristics of the utilized models are explained. Learning from data can be realized in three different ways: Unsupervised, semi-supervised and supervised learning. Supervised learning involves training a learning model with labeled data, unsupervised learning focuses on clusters in unlabeled data and semi-supervised learning is realized through the combination of labeled and unlabeled data during the training process. This study examines classification models like SVM, FastText and BERT, which are considered supervised learning techniques.

3.5.1 Support Vector Classification (SVC)

Support Vector Classification is a type of Support Vector Machine algorithm for classification. The classification principle is realized by separating different classes by finding the best optimal hyperplane possible. The hyperplanes can come in various

forms such as linear, polynomial, sigmoid or sphere (Scikit-learn, 2023). The complexity of hyperplane may give more appropriate results for non-linear tasks. In addition to hyperplane structure, the parameters such as Gamma and Regularization affect how inclusive the decision boundaries can be or not. As indicated by Figure 3.5, an RBF (Radial Basis Function) Kernel can perform more detailed class separation compared to linear kernel. This model is used as the baseline model for this study with those parameters:

- Kernel: The boundary type. It is selected as RBF (Gaussian Distribution).
- Gamma: It determines RBF width, set to *1/(n_features·X.variance())*
- Regularization Coefficient: The ratio between training and margin. Selected as 1.

SVC can offer several advantages for text classification such as:

- Ability to capture non-linearity
- Robust to outliers
- Support for imbalanced data
- Handling sparse data effectively.

However, it also comes with certain disadvantages such as:

- High demand for computational resources, including memory & processing power.
- Slower performance due to time complexity depending on kernels
- Limited capacity to learn complex patterns in comparison to neural networks

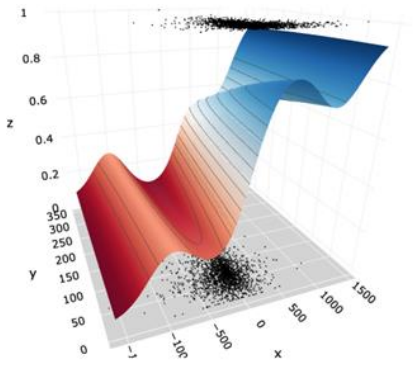

Figure 3.5: An Example 3-Dimensional RBF Kernel Visualization
(Dobilas, 2021)

3.5.2 FastText

Capturing the semantic essence of the text data is realized through the mathematical representation of text, a.k.a vectors. There are various techniques to extract vectors such as TF-IDF (Term Frequency - Inverse Document Frequency), Count Vectors, N-grams, and BoW (Bag of Words), which could be used as one-hot encoded. These methods generate sparse vector spaces proportional to the number of words within corpora, thus the computational cost is usually high with one-hot encoded text representation methods.

Word embeddings (*Word2Vec*/*GloVe*) are the trained vectors from the corpora to capture relationships between words, thus they are in a more compact form than the one-hot encoded text representations, which also results in having a reduced dimensionality for the vector space. The main idea behind this approach is that the words appear together within the same context more frequently. They can be used for predicting the context words from a target word (*Skip-Gram*) or predicting the target from the context words (*CBOW- Continuous Bag of Words*). As the training is highly depended on the corpus, they cannot generate vectors for out-of-vocabulary words and the words are only interpreted as their appearance within the context, without utilizing the internal structure of the words.

FastText is implemented by Facebook AI Research (FAIR) team and the team aimed specifically to reduce the computation complexity and the training loss compared to neural networks (Joulin et al., 2016). It is an improved embedding method by using the subword information by introducing *Character N-Grams* along with *Skip-gram*. Character N-Grams are the consecutive characters within the words, which helps to extract the understanding for morphologically rich languages such as Turkish (Akdemir et al., 2020). To provide further clarification, the character 3-grams for the word "*uygulama*" could be represented as: <*uy*, *uyg*, *ygu*, *gul*, *ula*, *lam*, *ama*, *ma*>. This approach also solves the issue for representing rare and out-of-vocabulary words.

Once the character n-grams are generated, they are averaged into a hidden text representation, which is utilized as the input for the linear transformation. Hierarchical

softmax is used during the neural network training to calculate the probability distribution based on the Huffman tree algorithm resulting in less testing time when searching for the most probable class within a large list of number of classes (Akdemir et al., 2020). Figure 3.6 shows the main steps for FastText classification.

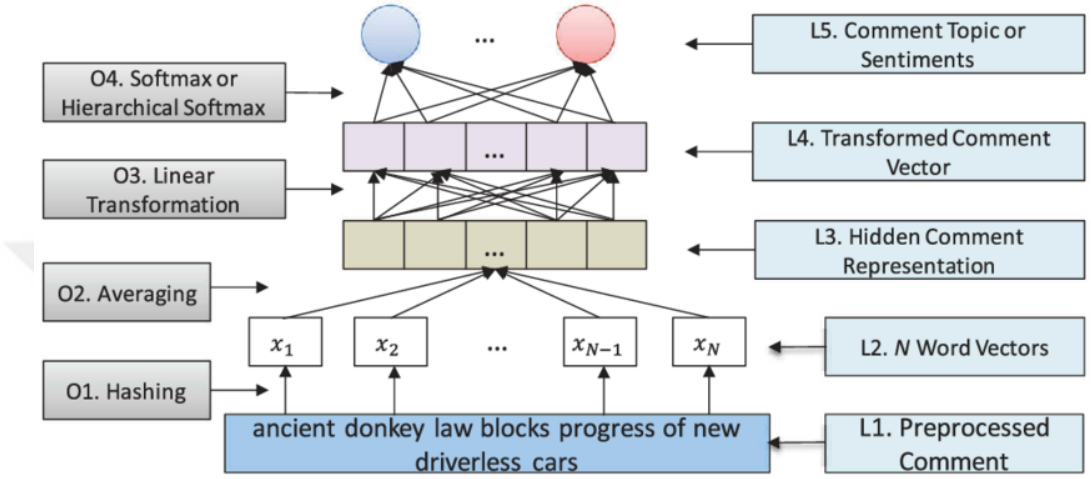

Figure 3.6: FastText Model Architecture (Zhou, Yang, & Zhang, 2020)

3.5.3 BERT (Bidirectional Encoder Representations from Transformers)

While FastText enables improved word representations through its algorithm, the FastText word embeddings are not contextualized; the meaning of the word still be the same within a sentence even though the surrounding context is different. The lack of contextualization could be a possible limitation in tasks where understanding the intricate relationship within the text is crucial.

Attention mechanism provided by BERT, a specific type of Transformer model, provides a great solution for the contextualization problem. Before explaining BERT Architecture, it is important to explain Attention Mechanism & Transformer structures.

3.5.3.1 Attention Mechanism & Transformer Architecture

Attention mechanism is a technique involving assigning different weights to different

parts of an input sequence. Query (Q), Key (K) and Value (V) vectors are utilized to generate the attention mechanism. Its formula is:

$$Attention(Q, K, V) = softmax\left(QK^T / \sqrt{d_k}\right)V \quad (1)$$

As this formula shows, the attention score is obtained by the dot product of Query and Key; and the calculated attention scores are applied on the value vector. As the weight scores are depended on Query and Key, the mechanism enables to assign different weights on the different input tokens.

Transformers are the deep learning models having multiple layers consisting of encoder and decoder layers which use Multi Head Attention and Feed Forward Networks. Multi Attention Heads acts independently on different parts of input sequence. Figure 3.7 shows the attention head visualization, created with a tool called *BERTVIZ* (Vig, 2019), between the token "tablet" and the other tokens within the given sentence for the layer 2. Intense colors show a strong attention weight for the other token. The strongest attentions for *tablet* within the layer 2 is assigned to the tokens *650, CLS, kapli* and *Her.* As there are 12 layers and 12 Attention heads, different weights can be provided by 144 distinct combinations. **'CLS'** (Classification) and **'SEP'** (Separator) are the special tokens to indicate the input sequence and the different parts (Sentences) within the input sequence respectively.

The sequential operations run in parallel and the model does not involve recurrence and convolution yet there is a necessity for positional information (Vaswani et al., 2017). As Figure 3.8 shows, Positional embeddings are fed within input embeddings at the bottom of encoder and decoder sections.

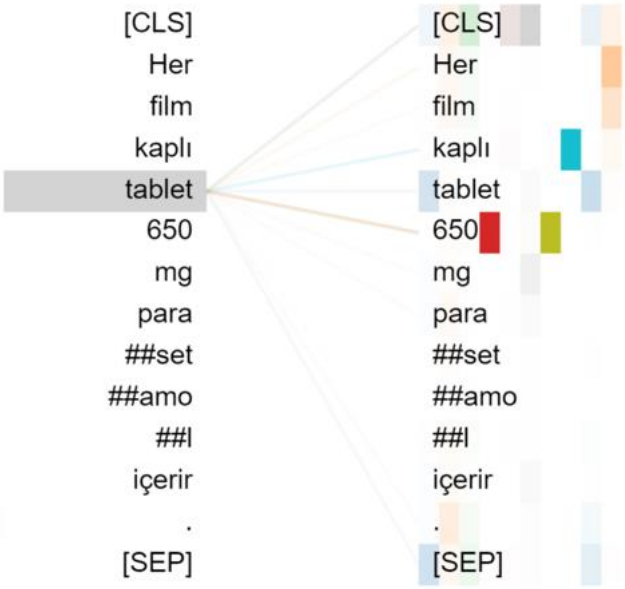

Figure 3.7: Attention Visualization for “Her film kaplı tablet 650 mg parasetamol içerir.”

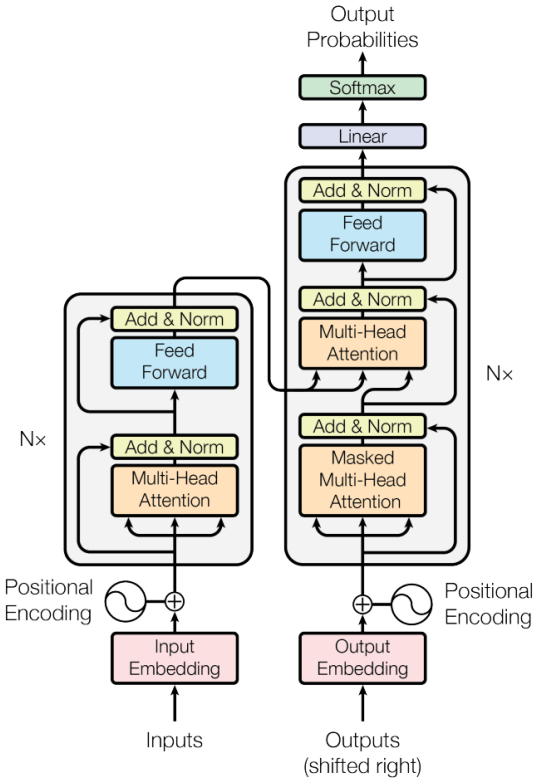

Figure 3.8: Transformer Model Architecture

3.5.3.2 BERT Architecture

BERT is a state-of-the-art Transformer based encoder model initially designed by Google (Devlin et al., 2018) to enhance the effectiveness of transfer learning. The transfer learning is the method by which a model, after being trained for one task, can be generalized by being used for another task. In other words, BERT models are pretrained unsupervised on immense corpora on two tasks called Masked Language Modeling (MLM) masking tokens one by one and trying to predict the actual token; and Next Sentence Prediction (NSP) aiming to extract relations between sentences (Devlin et al., 2018). The representations are obtained bi-directionally; from both the left and right side of tokens.

BERT Input embedding layer consists of position embeddings, existing in Transformer models, as well as additional token embeddings and segment embeddings (Devlin et al., 2018). BERT tokenizers are capable of creating those embeddings; each sentence are assigned a different segment id and word or subword tokenization, which is preferred more as it serves better handling for rare or out-of-vocabulary words, is applied to obtain token embeddings. The *Uncased* tokenizers lowercase the tokens; whereas *Cased* tokenizers keeps capitalized and lowercase words together (Huggingface, 2023).

After pretraining, BERT is ready to be used (supervised) on other tasks such as sentiment analysis, entity recognition and automatic categorization. Generalization for various tasks is realized through finetuning. As it can be observed in Figure 3.9, majority of the pretrained model remains intact; only task-specific layers, a custom classification head, are added on top of the existing model. Learned representations within pretrain models, such as general language rules, relationships between words and sentences, domain-based information are mapped for the targeted classes in this layer.

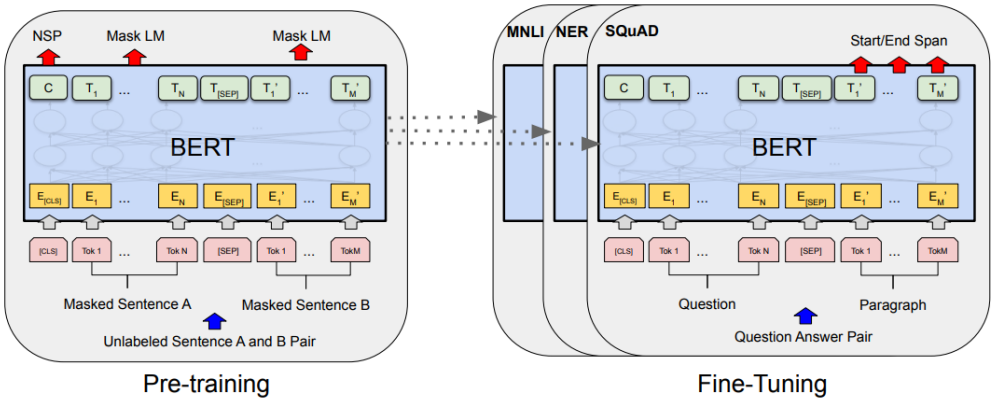

Figure 3.3.9: BERT Architecture for Pretraining and Finetuning (Devlin et al., 2018)

3.5.3.3 Employed BERT Variants

Turkish BERT variants are trained with many corpora, including Turkish OSCAR corpus, a Wikipedia dump, various OPUS corpora and a corpus provided by Kemal Oflazer (Schweter, 2020). The different BERT models used in this study include the same pretraining configurations as the original English BERT model such as:

- Vocabulary size: 32000 (128000 for BERT Large)
- Maximum Position Embeddings: 512
- Number of Attention Heads: 12
- Number of Hidden Layers: 12
- Intermediate Size (Number of Neurons in Hidden Layers): 3072

The BERT variants utilized in this study are:

1. **BERTurk** (uncased): Variation of BERT Base for English. The training corpus size is 35GB and the corpus contains approximately 4.4 billion tokens (Huggingface, 2023). All the variants are trained on the same data.
2. **BERTurk Large** (uncased): It shares the same structure with BERTurk, but the vocabulary size is 128K.
3. **ELECTRA BERTurk** (uncased): It contains an additional discriminator model

which changes input masking process by “corrupting” the input token by the generator network then it predicts whether each token was generated by the discriminator (Clark et al., 2019). The creators of this method argue that owing to this generate-discriminate model, the model can benefit from all tokens and therefore ELECTRA will be more effective and faster than BERT Base.

4. **ConvBERTurk** (uncased): Convolution is deployed on attention heads in the architecture. Jiang et al. state that while trying to map all heads globally in the self-attention mechanism, the cost during memory and computation is high; on the top of it some heads are already working only to obtain local connections, which can be rearranged by the Convolutional models to both increase redundancy and obtain global and local connections together for the input embeddings.
5. **DISTILBERTurk** (cased): Distillation is when a larger model, or a model seen as a "teacher"; makes another smaller model, or a model seen as a student, reproduce its behavior (Sanh, 2020). The student is trained with the distillation loss of the teacher model, the BERT architecture is also adjusted by the removal of Token Embeddings and Poolers and the layers are reduced for the sake of computational efficiency (Sanh, 2020)

The performance results of these variants are explained in the Experiments and Results section.

3.6 Technical Requirements

This study is carried out using various libraries in the Python programming language. The used libraries and tasks are:

- *BeautifulSoup*: Parsing HTML content for obtaining PDF links & Obtaining ATC labels
- *Requests*: Downloading PDF files
- *PyMuDF*: Converting text-based PDFs to plain text format (TXT)
- *Pillow*, *Pytesseract*: Converting pages to images for scanned-based PDFs and reading image to plain text format (TXT)

- *Snowball Stemmer*: Stemming for Turkish words
- *Zemberek & Zeyrek*: Lemmatization & Stop words removal for Turkish words
- *Pandas*: Data manipulation & analysis
- *Sklearn*: Extracting Count Vectors, data splitting, training SVC model, obtaining classification reports
- *RegEx*: Removing non alphabetic characters & unwanted non-ASCII characters
- *HuggingFace*: Tokenizing & Finetuning BERT Models
- *FastText*: Training with FastText Model
- *Optuna*: Hyperparameter Optimization for BERT models

Google Colaboratory (Colab) notebooks used to run the codes on cloud environment. This environment provides various types of virtual machines for GPU such as K80, T40, V100 and A100. The file conversions are realized on 2 x vCPU with 12GB RAM, whereas the high memory demanding tasks such lemmatization, stemming and SVC trainings are realized on 2 x vCPU with 52GB RAM. Hyperparameter optimizations for BERT models, requiring extremely high level of GPU capacity, run on V100 GPUs, which are observed as faster than T4.

4. EXPERIMENTS & RESULTS

In this section, the experimental results for ATC alongside with from the findings from our prior research, ICD classification are explained. More advanced techniques such as hyperparameter optimization and lexical normalization are applied for ATC classification as it is the primary focus for this study.

4.1. Performance Evaluation Criteria

As a result of classification models, a label is predicted for each data. The following indicators are used for performance measure (F1-Score):

- *True Positive* (**TP**): If data is correctly predicted for an actual positive class
- *True Negative* (**TN**): If data is correctly predicted for an actual negative class
- *False Positive* (**FP**): If data is predicted as positive class given for a negative class
- *False Negative* (**FN**): If data is predicted as negative class given for a positive class

Precision is the metric showing the ratio for how many of the positive classes are caught; whereas recall is the metric showing how many of the positive classes are truly positive. Given TP, TN, FP and FN rates; precision, recall and F1 scores are calculated with the formulas below:

$$Precision = TP/(TP + FP) \tag{2}$$

$$Recall = TP/(TP + FN) \tag{3}$$

$$F1 = 2 * Precision * Recall/(Precision + Recall) \tag{4}$$

It may be thought that only the accuracy value will be sufficient for measurement. However, the datasets on which performance measurement is made may not always be

balanced: Some classes might outnumber the other classes and, in these cases, if only accuracy value is considered, there will be a positive bias towards the dominant class and the mistakes made in the non-dominant classes will be less important. F1 scoring considers both false negatives and false positives equally.

In cases where more than one class exists, different approaches can be used to measure the metric to be used, which is called *averaging*. Averaging could be calculated on class or instance based; if a measurement is made on instance basis, it is called micro averaging, and if it is done on a class basis, it is called macro averaging. The formulas for macro and micro measurements are given below:

$$Micro\ Precision = Instance\ TP/(Instance\ TP + Instance\ FP) \quad (5)$$

$$Micro\ Recall = Instance\ TP/(Instance\ TP + Instance\ FN) \quad (6)$$

$$Micro\ F1 = 2 * Micro\ Precision * Micro\ Recall/(Micro\ Precision + Micro\ Rec) \quad (7)$$

$$Macro\ Precision = avg(Precision\ per\ classes) \quad (8)$$

$$Macro\ Recall = avg(Recall\ per\ classes) \quad (9)$$

$$Macro\ F1 = 2 * Macro\ Prec * Macro\ Rec/(Macro\ Prec + Macro\ Rec) \quad (10)$$

Although micro measurements show slightly better results in favor of the dominant class than macro values in imbalanced datasets, it has been observed that both values could be used as criteria in these datasets. In this study, since measuring the success of the model over every instance that can come as input is prioritized, the selected main criterion is micro F1. The macro and micro results will be compared in detail in the next section.

4.2. Model Optimizations & Optimization Results

Hyperparameters are the parameters that control the learning factors such as convergence, performance objectives (e.g. Min. loss or Max. F1-Score), the running time or cycles, dimensions and topological properties; therefore they heavily affect the operating time and the performance results of the models. In this regard, it is important to find and utilize the most optimal hyperparameter combination possible between cost (time) and

successful performance results to maintain the quality of classification. For the sake of avoiding the curse of dimensionality and shortening the run time, hyperparameter combinations are searched within selected sets of intervals for both FastText and BERT models. However, hyperparameter optimization was not performed for BERT models in the ICD classification; instead, various combinations and their results are recorded.

4.2.1 Hyperparameter Optimization for FastText

FastText offers its library called **Autotune** for hyperparameter tuning. It will be sufficient to specify training set, evaluation set, performance metric and the autotune duration/model size for this task.

4.2.2 Hyperparameter Optimization for BERT Models

There are many parameters that need to be tuned for structurally complex BERT models. Various tuning algorithms such as Grid Search, Randomized Search, Bayesian Search can be used. Still, it is important to consider the tradeoff between the exploration and the exploitation provided by the algorithms. Grid Search provides very high deterministic exploration, but it requires heavy computational resources; on the other hand Random Search requires less computational resources but it provides a randomly chosen configurations.

Since both methods can be interpreted as at the extremes, Tree-structured Parzen Estimator, a Bayesian Search technique selected as Hyperparameter Optimization algorithm, which falls between them in terms of exploration and exploitation. While searching for hyperparameters, Gaussian probabilistic function is used. By using some of the instances within hyperparameter space, good and bad points are distinguished through the objective function (e.g. minimizing loss or maximizing F1). The points to be selected continue to be selected from the created “good points” region. Optuna provides TPESampler so that the probability function can be updated as a with the help of the objective function applied on the previous trials. In this study, the selected hyperparameter space is consisted of:

- **Learning Rate**: [1e-5,5e-5]
- **Number of Train Epochs**: [2,5], resolution =1
- **Per Device Train Batch Size**: [8,16]

4.3. Results for Turkish Medical Summaries Dataset

80% of the data set was selected to be used for training and 20% for validation. The selected hyperparameters are: *Dim*:500, *epoch*:30, *learning_rate*:0.1. Various hyperparameter combinations and their overall F1-Scores are included in the experimental set.

As shown in Table 4.1, the best score is 92.1% overall Micro F1-Score with DistilBERTurk Model. As a result, those hyperparameters were selected for the maximal F1 Score:

- Autotune duration:300s, Epoch=100, wordNgrams=1

Table 4.1: F1 Scores per Each Model

Model	Learning rate	Epoch	Ngrams	Overall F1-Score
FastText	0.1	30	4	40%
FastText	0.1	30	1	83.3%
FastText-Autotuned	0.1	100	1	86%
BERTurk	2e-5	3	-	75.5%
BERTurk	4e-5	10	-	86.9%
BERTurk	5e-5	10	-	91.3%
BERTurk	1e-3	10	-	15%
DistilBERTurk	5e-5	10	-	90%
DistilBERTurk	**1e-4**	**5**	**-**	**92.1%**
DistilBERTurk	5e-4	5	-	30%
ElectraTurk	1e-4	5	-	79.7%
ElectraTurk	1e-4	10	-	87%

Different parameters are used in our experiment set such as the learning rate and epoch number. An increase in these parameters after the optimal values causes the result in the underfitting.

Summaries' dataset, it is shown in detail how different architectures with different parameters change the result and the performance results such as 86% overall F1-Score with FastText, 92.1% with DistilBERTurk and 92% with BERTurk are obtained; the ICD-10 Codes (C Type) are successfully predicted.

4.4. Results for ATC - Turkish Drug Manuals Dataset

In this section, the performance results for SVC, FastText and BERT Variants are shown. Experiments were carried out on the set specified in Table 4.2. Employed models, GPU type, the costly hour for GPU type and the training times are mentioned in Table 4.3. As the model hypertuning couldn't be completed due to the memory error with T4, only the the average time per epoch is provided for T4. For V100, the total time spent on hyperparameter tuning is also included.

Table 4.2: Experiments and Models

Technique Applied on Dataset	Model
Lemmatization + Count Vectors	SVC
Stemming + Count Vectors	SVC
Only Lowercasing, Objective: Micro F1	All BERTs
Non-Alphabet Chars. Removal, Objective: Micro F1	All BERTs
Non-ASCII Chars Removal, Objective: Micro F1	FastText
Non-Alphabet Chars. Removal, Objective: Micro F1	FastText

Table 4.3: Average Execution Time for Training

Model (Per Epoch)	GPU Type	GPU Cost/hour (Compute Unit)	Time/Epoch (Min)	Total Optimization Time (Min)
SVC	-	-	28	-
FastText	-	-	5	5

BERTurk Uncased	V100	5.5	5	120
ElectraBERTurk	V100	5.5	4.5	107
ConvBERTurk	V100	5.5	4	99.5
BERTurk Large	V100	5.5	4.1	92.4
DISTILBERTurk	V100	5.5	2.1	64
BERTurk Uncased	T4	2.2	12	Cuda Memory Error
ElectraBERTurk	T4	2.2	12	357
ConvBERTurk	T4	2.2	12	Cuda Memory Error
BERTurk Large	T4	2.2	11.5	Cuda Memory Error
DISTILBERTurk	T4	2.2	6.3	Cuda Memory Error

4.4.1 SVC Results

Table 4.3 shows the micro F1 and Macro Precision, Recall and F1 scores. The best score for Stemming applied dataset is 94% overall micro F1 score and 88% overall macro F1 score. The reason behind this 6% difference is the data imbalance as macro results calculates the average ratio per class. The best score for Lemmatization is 93% and the macro F2 is 85%. A sharp decrease for macro-Recall (83%) might indicate that model misses the True Positives to some extent. Figure 4.1 and Figure 4.2 show the top 10 worst and best performing ATC categories. The model cannot correctly categorize any "*Other Gynecological*" or *Antifungal* drug in the test data set. On the other hand, it correctly categorizes all Antacid, Gastrointestinal Disorder, Beta Blocker, Antibiotic and Vitamin D drug information.

Table 4.3: SVC Results

Technique Applied on Dataset	Micro F1	Macro Precision	Macro Recall	Macro F1
Stemming + Count Vectors(4000)	0.94	0.92	0.86	0.88

Lemmatization + Count Vectors (4000)	0.93	0.91	0.83	0.85

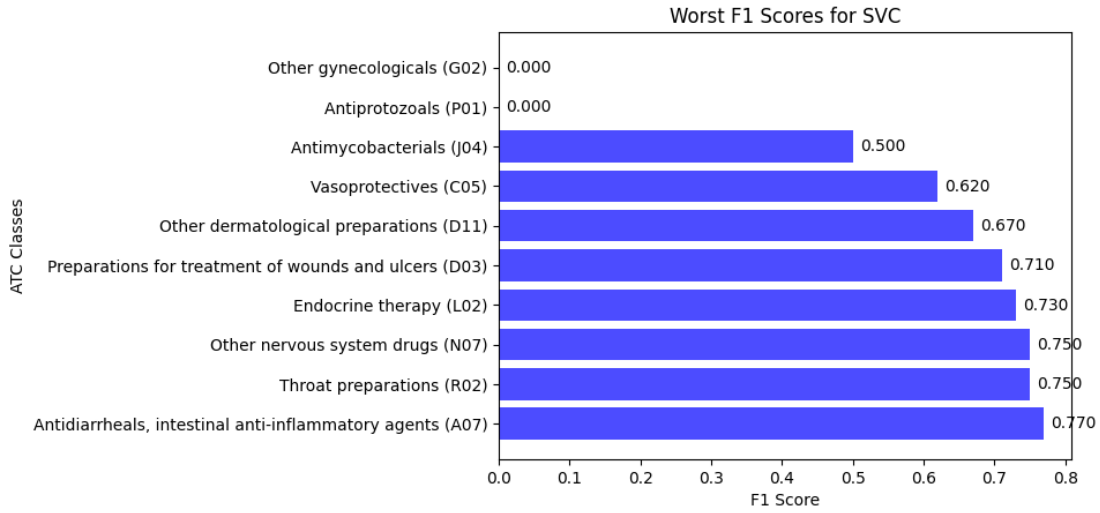

Figure 4.1: Top 10 Worst Performing ATC Classes with SVC

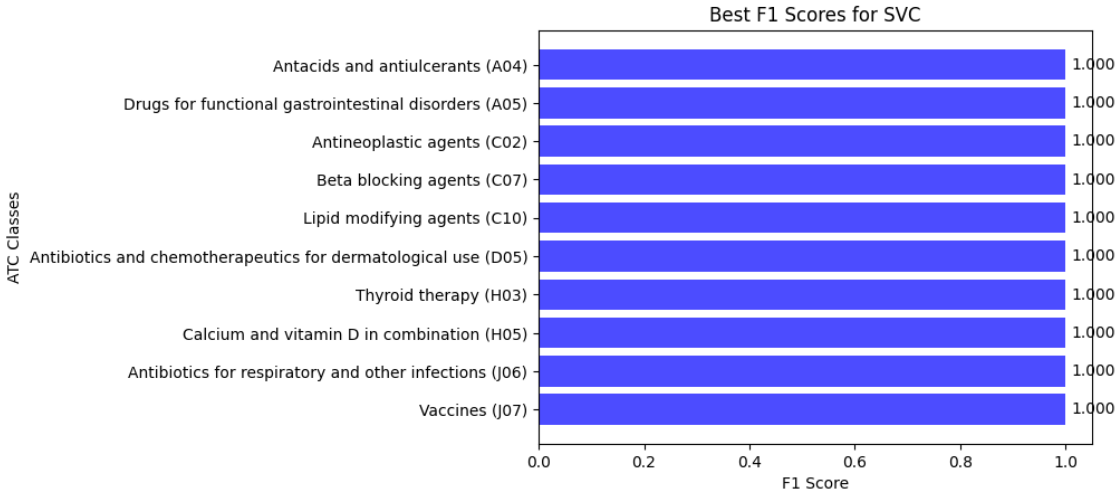

Figure 4.2: Top 10 Best Performing ATC Classes with SVC

4.4.2 FastText Results

The best hyperparameters for FastText is:

- Autotune duration:300s
- Dim: 42
- Epoch: 83
- Loss: Softmax

- Learning Rate: 0.266
- wordNgrams: 1

The micro F1 and macro results are shown in Table 4.4. FastText is heavily affected by text preprocessing: when non-ASCII characters (symbols) and numbers are kept in the text, the noise causes high ambiguity. Figure 4.3 and Figure 4.4 indicate the top 10 worst and best performing ATC categories for FastText. Other Gynecologicals, Antimycobacterials and vasoprotectives are the worst performing ATC codes, whereas Vitamin D/Calcium combinations, Antibiotics, dermatological usage antibiotic drug information are all correctly categorized. Comparison of the top 10 and bottom 10 categories with other classes will be discussed in the Conclusion section.

Table 4.4: FastText Performance Results

Technique Applied on Dataset	Micro F1	Macro Precision	Macro Recall	Macro F1
No Preprocessing	0	0	0	0
Lowercasing	0	0	0.03	0
Non Alphabet Char. Removal	0.96	0.93	0.92	0.92

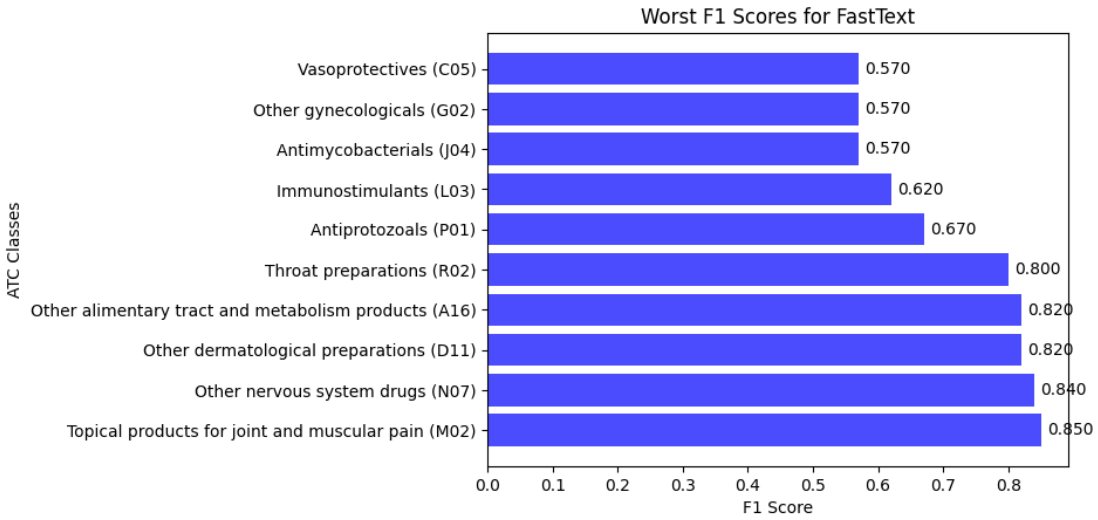

Figure 4.3: Top 10 Worst Performing ATC Classes with FastText

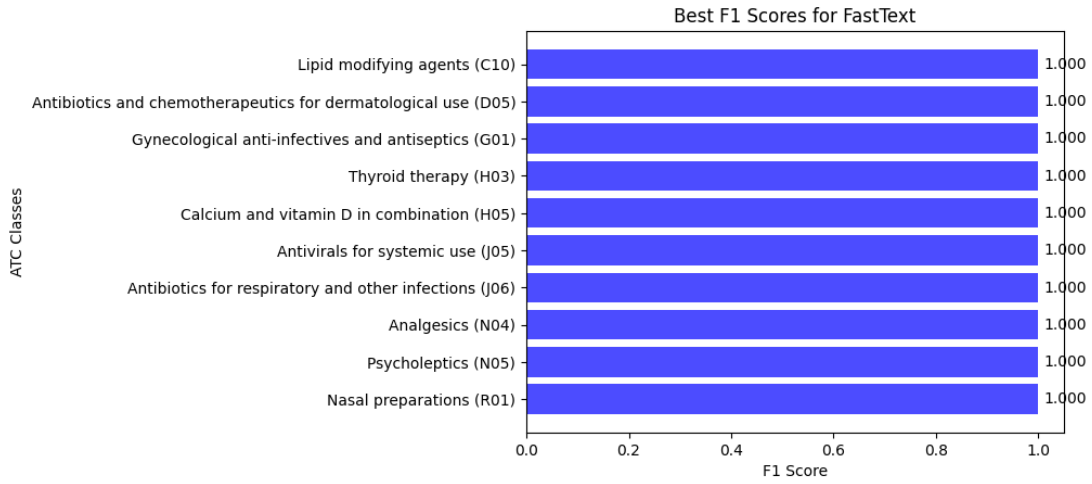

Figure 4.4: Top 10 Best Performing ATC Classes With FastText

4.4.3 BERT Results

4.4.3.1 Hyperparameter Optimization with Only Lowercasing

In this experiment, lowercasing is applied to the dataset in addition to BERTurk tokenizer. Table 4.4, 4.5, 4.6, 4.7 show the obtained overall micro F1 scores per trial number, the learning rate provided by TPE Sampler per specific trial, the number of epochs used for training per specific trial, and Per device batch size for training per specific trials. As each trial are independent to each other, Early Stopping or producing lower F1 scores might happen between subsequent trials. As Optuna trials do not provide the suggested hyperparameters in case of Early Stopping, those hyperparameters are indicated as '-'.

Table 4.4: Hyperparameter Optimization Results for BERTurk Uncased

Trial Number	Learning Rate	Num of Train Epochs	Per-device Train Batch Size	Overall Micro F1-Score

0	1.25e-05	5	8	0.940
1	3.37e-05	2	16	0.923
2	**4.60e-05**	**3**	**16**	**0.949**
3	1.65e-05	5	16	0.943
4	3.02e-05	3	8	0.949
5 (Pruned)	-	-	-	0.865
6 (Pruned)	-	-	-	0.873
7 (Pruned)	-	-	-	0.800
8 (Pruned)	-	-	-	0.813
9 (Pruned)	-	-	-	0.943

Table 4.5: Hyperparameter Optimization Results for ElectraBERTurk

Trial Number	Learning Rate	Num of Train Epochs	Per-device Train Batch Size	Overall Micro F1-Score
0	1.35e-05	5	16	0.740
1	4.97e-05	2	8	0.851
2	1.39e-05	2	16	0.386
3	2.12e-05	3	16	0.710
4	4.75e-05	4	16	0.920
5 (Pruned)	-	-	-	0.319
6 (Pruned)	-	-	-	0.574
7	**3.97e-05**	**5**	**8**	**0.948**
8 (Pruned)	-	-	-	0.632
9 (Pruned)	-	-	-	0.275

Table 4.6: Hyperparameter Optimization Results for ConvBERTurk

Trial Number	Learning Rate	Num of Train Epochs	Per-device Train Batch Size	Overall Micro F1-Score
0	1.74e-05	4	8	0.904
1	**3.09e-05**	**4**	**8**	**0.934**
2	2.16e-05	4	16	0.871
3	2.73e-05	2	16	0.770
4	4.81e-05	3	16	0.904
5 (Pruned)	-	-	-	0.705
6 (Pruned)	-	-	-	0.715
7 (Pruned)	-	-	-	0.421

8 (Pruned)	-	-	-	0.747
9	3.45e-05	4	16	0.932

Table 4.6: Hyperparameter Optimization Results for BERTurk Large

Trial Number	Learning Rate	Num of Train Epochs	Per-device Train Batch Size	Overall Micro F1-Score
0	1.61e-05	4	8	0.906
1	1.45e-05	4	16	0.871
2	**2.89e-05**	**4**	**16**	**0.918**
3	4.41e-05	2	16	0.859
4	1.64e-05	4	8	0.913
5 (Pruned)	-	-	-	-
6 (Pruned)	-	-	-	-
7 (Pruned)	-	-	-	-
8 (Pruned)	-	-	-	-
9 (Pruned)	-	-	-	-

Table 4.7: Hyperparameter Optimization Results for DISTILBERTurk Cased

Trial Number	Learning Rate	Num of Train Epochs	Per-device Train Batch Size	Overall Micro F1-Score
0	1.02e-05	3	8	0.316
1	1.27e-05	5	16	0.490
2	1.61e-05	4	16	0.510
3	2.09e-05	3	16	0.473
4	1.36e-05	3	8	0.450
5	2.14e-05	5	16	0.652
6 (Pruned)	-	-	-	-
7	**4.00e-05**	**4**	**8**	**0.827**
8	3.59e-05	2	16	0.460
9 (Pruned)	-	-	-	0.538

Table 4.8 gathers the best F1 scores obtained by each hyperparameter tuning experiments. In addition, Figure 4.1 and Figure 4.2 shows the 10 worst and best predicted ATC codes since it is challenging to show 69 categories together. Throat Preparations, Anti fungals,

Antihistamines are predicted with 0% F1 score and drugs for Antacids, Gastrointestinal disorders, diabetes and vaccines are predicted best with 100% F1 Score.

Table 4.8: Best Results for Micro F1 Objective

BERT	Learning Rate	Num of Train Epochs	P.D. Batch Size	Overall Micro F1-Score
BERTurk Uncased	4.59e-05	3	16	0.949
ElectraBERTurk	3.97e-05	5	8	0.948
ConvBERTurk	3.08e-05	4	8	0.939
BERTurk Large	2.89e-05	4	16	0.917
DISTILBERTurk	1.36e-05	3	8	0.827

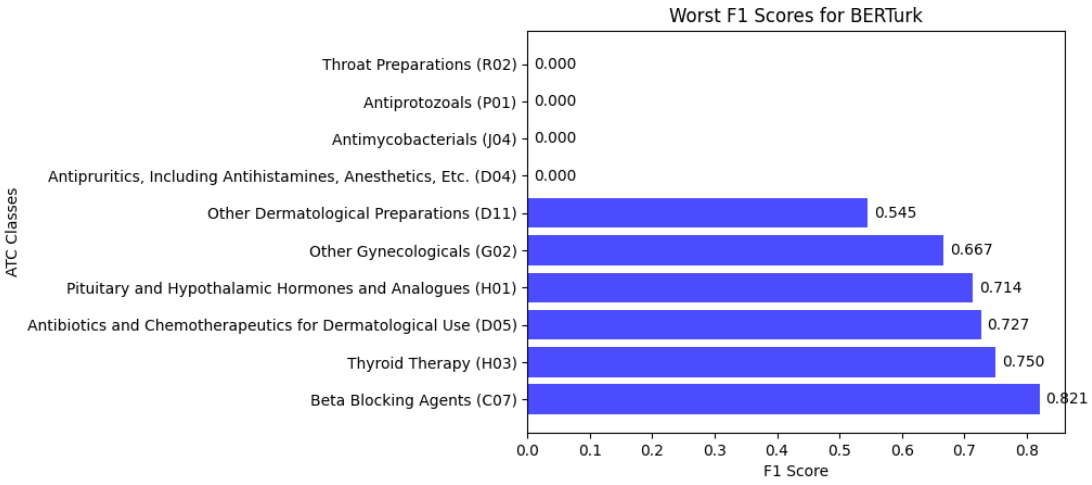

Figure 4.5: Top 10 ATC Classes Performing Worst with BERTurk

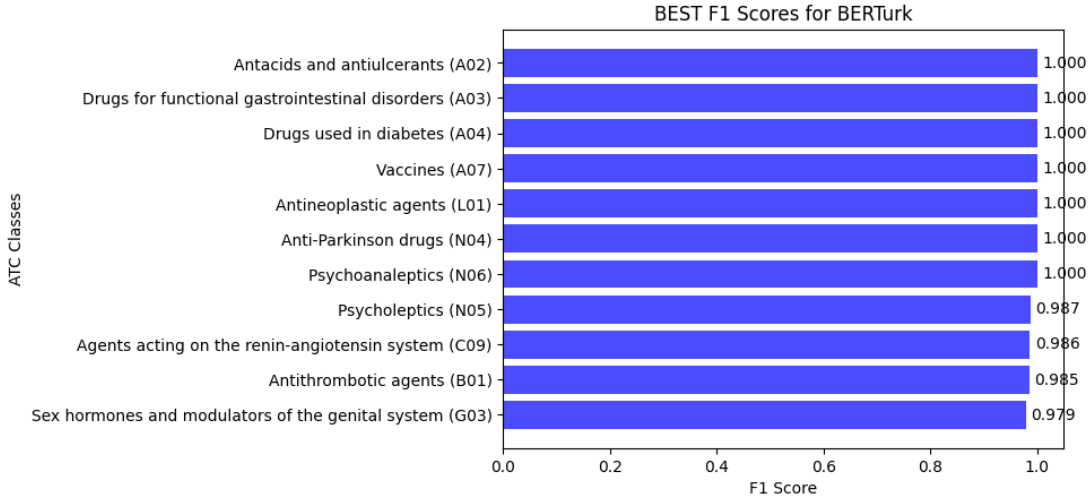

Figure 4.6: Top 10 ATC Classes Performing Best for BERTurk

4.4.3.2 Hyperparameter Optimization with Non-Alphabet Character Removal

Table 4.9: Hyperparameter Optimization Results for BERTurk

Trial Number	Learning Rate	Num of Train Epochs	Per-device Train Batch Size	Overall Micro F1-Score
0	3.56e-05	3	16	0.948
1	2.25e-05	2	8	0.925
2	**3.45e-05**	**4**	**16**	**0.959**
3	3.23e-05	3	16	0.943
4	4.637e-05	2	16	0.933
5 (Pruned)	-	-	-	0.779
6 (Pruned)	-	-	-	0.807
7 (Pruned)	-	-	-	0.882
8 (Pruned)	-	-	1-	0.876
9 (Pruned)	-	-	-	0.890

Table 4.10: Hyperparameter Optimization Results for Electra BERTurk

Trial Number	Learning Rate	Num of Train Epochs	Per-device Train Batch Size	Overall Micro F1-Score
0	1.79e-05	4	16	0.74
1	2.36e-05	3	8	0.829

2	3.87e-05	3	16	0.849
3	4.14e-05	5	16	0.942
4	3.14e-05	3	8	0.857
5 (Pruned)	-	-	-	-
6 (Pruned)	-	-	-	-
7 (Pruned)	-	-	-	-
8	**4.58e-05**	**5**	**8**	**0.946**
9 (Pruned)	-	-	-	-

Table 4.11: Hyperparameter Optimization Results for ConvBERTurk

Trial Number	Learning Rate	Num of Train Epochs	Per-device Train Batch Size	Overall Micro F1-Score
0	**4.45e-05**	**3**	**16**	**0.904**
1	1.96e-05	4	16	0.880
2	1.69e-05	2	16	0.618
3	1.69e-05	5	16	0.902
4	2.08e-05	4	16	0.885
5 (Pruned)	-	-	-	-
6 (Pruned)	-	-	-	-
7 (Pruned)	-	-	-	-
8	3.20e-05	2	16	0.795
9	2.35e-05	3	16	0.868

Table 4.12: Hyperparameter Optimization Results for BERTurk Large

Trial Number	Learning Rate	Num of Train Epochs	Per-device Train Batch Size	Overall Micro F1-Score
0	**3.70e-05**	**5**	**8**	**0.950**
1	4.18e-05	4	8	0.940
2	3.56e-05	2	16	0.875
3	2.87e-05	3	8	0.892
4	2.27e-05	3	16	0.890
5	3.95e-05	3	16	0.930
6 (Pruned)	-	-	-	0.504
7 (Pruned)	-	-	-	0.754
8	3.20e-05	2	16	0.629
9	3.96e-05	3	16	0.575

Table 4.13: Hyperparameter Optimization Results for DISTILBERTurk

Trial Number	Learning Rate	Num of Train Epochs	Per-device Train Batch Size	Overall Micro F1-Score
0	2.74e-05	2	16	0.388
1	1.38e-05	5	8	0.644
2	4.99e-05	3	8	0.573
3	1.94e-05	2	16	0.326
4	1.68e-05	3	8	0.563
5	2.82e-05	4	16	0.709
6	3.82e-05	4	16	0.778
7 (Pruned)	-	-	-	0.203
8	**4.73e-05**	**5**	**8**	**0.900**
9 (Pruned)	-	-	-	0.228

5. CONCLUSION

In this thesis, a Turkish Medical Drug Manual Dataset is created, in addition a Turkish Medical Summaries dataset obtained by external resources is analyzed for medical category classifications, respectively for ATC & ICD prediction purposes. Various classifications are optimized and are utilized for the chosen performance criteria.

11260 PDF files are converted to plain text format with the help of OCR or a text-based PDF converting library. The first 512 words containing the most important information for ATC classification are selected and the data set was created by obtaining ATC categories from an online website using the HTML parsing. By applying different pre-processing methods (lowercasing, non-Alpha character removal, stemming and lemmatization) to the original data set, it was investigated whether there was a difference in performance. It can be concluded that the non-alpha character removal is a must whereas preprocessing has a slight positive influence on BERT variants.

Different classifiers such as Support Vector Classifier, FastText and various BERT variants are implemented to obtain the best possible performance measurement criteria, micro F1 score. Hyperparameter optimization on FastText and BERT models are performed to find the best possible training parameters, i.e. hyperparameters. As a result, the best performance is achieved in the following situations: When all non-alphabet characters on the dataset are removed, BERTurk, a Turkish BERT variant trained in the same structure and configuration as the BERT Base model, produces 96% micro F1 score. At the same time, 96% F1 score can be obtained with FastText in the same data set where this preprocessing is applied. Although FastText and BERT seem to give head-to-head results, FastText cannot produce a tangible result when this preprocessing technique is not performed.

SVC also achieves a good micro F1 score with a slight difference, 93%. Stemming and lemmatization resulted in a minimal difference of only 1% between micro F1 results; Recall values are lower than precision in both cases and lemmatization gives lower recall (83%) values than stemming (86%). However, another factor that is crucial to note is that lemmatization takes approximately 4 hours, while stemming takes 10-15 minutes. Therefore, stemming may be preferred over lemmatization.

For the dataset where minimal text cleaning was performed, only the Turkish BERT tokenizer converted to lowercase letters, a 95% F1 score was obtained with BERTurk Uncased and Electra BERTurk models. DISTILBERTurk produced the lowest performance result within BERT models, in both data sets with (90%) and without pre-processing (82.7%). That is, the reduction of model layers and information distillation had a negative effect compared to the original BERT architecture. In addition, BERTurk Large, in which 128k words can be used, produced lower results in the selected hyperparameter space, although it has the same architecture as BERTurk Uncased.

Tree Parzen Estimator was used for hyperparameter optimization. This algorithm served as a sampler for the Optuna library, that is, it suggested hyperparameters per Optuna trial with the probability distribution algorithm it provided. However, on Optuna, each trial runs independently, so if one trial does not give sufficient results, it does not mean that the next trial will also give inadequate results. A constant increase or decrease cannot be observed in successive trial results. Still, early stopping used in the trials saved unnecessary waiting time.

Finally, the models are compared in terms of cost and time only on CPU and different GPUs, T4 and V100 on the cloud environment (Google Colaboratory). While the average hourly pricing coefficient (Compute Unit) is 5.5 for V100, it is 2.2 for T4. V100 is a better option in case of cumulative cost, as the GPU related tasks run in T4 approximately three times than V100 and V100 only costs 2.5 times than T4 hourly. The DISTILBERTurk model is trained about half the time of other BERT models, which may be explained by the fact that the model architecture is smaller than the others because the layers are reduced. The fact that only 5-6 percent lower performance scores are achieved while

saving 50 percent of time and cost indicates that DISTILBERT can be a viable option.

5.1 Thesis Contribution

With this study, a Turkish drug corpus is prepared for the first time. Although many different drugs specific to Turkiye are produced and there is even a Turkish Pharmaceutical Agency (TITCK), such as EMA or FDA, classification (ATC code indicating the disease and the relevant organ or system) on drugs, was carried out for the first time. At the same time, a detailed Bayesian (TPE) hyperparameter optimization is performed for the first time on Turkish BERT models and this optimization is analyzed in terms of time and cost.

5.2 Limitations and Future Work

The limitations of this study are as follows: the dataset is consisted of a few dominant classes and it contains too many categories (69), the fact that Turkish language is a low-resource language and has not been studied as much as English, the necessity of manual checks for the converted plain text files, the fact that each model takes approximately 2 hours (V100 GPU) even though Bayesian optimizer is used since 10 trials are employed and the average recommendation by TPE optimizer is approximately 3 epochs.

As future work, it is planned to train an NER that can automatically extract drug entities and a Turkish question and answer mechanism about drugs on this corpus that we have created as our distant target. As for our near-term goal, we are considering using ATC codes to detect possible interacting drugs, so we plan to include disease information in - addition, which means that graph-based models need to be implemented.

REFERENCES

Wong GK, Li SY, Wong EW (2016) Analyzing academic discussion forum data with topic detection and Data Visualization. 2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE). doi: 10.1109/tale.2016.7851779

Chen X, Zou D, Cheng G, Xie H (2020) Detecting latent topics and trends in educational technologies over four decades using structural topic modeling: A retrospective of all volumes of Computers & Education. Computers & Education 151:103855. doi: 10.1016/j.compedu.2020.103855

Liu Y, Zhang S, Chen M, et al (2021) The sustainable development of financial topic detection and trend prediction by data mining. Sustainability 13:7585. doi: 10.3390/su13147585

World Health Organization. (2022). Global spending on health: rising to the pandemic's challenges (Meeting report). https://www.who.int/publications/i/item/9789240064911

Sub-sectors in the Health Care Industry. HSM. (n.d.). https://centers.fuqua.duke.edu/hsm/home/students/career-info/sub-sectors-in-the-health-care-industry/

European Medicines Agency. (2023). Pre-authorisation guidance. Product name, product information, and prescription status section. https://www.ema.europa.eu/en/human-regulatory/marketing-authorisation/pre-authorisation-guidance#3.1-product-name,-product-information-and-prescription-status-section

Kırmızı, N. İ., Aydın, M., Koyuncuoğlu, C. Z., Aksoy, M., Kadı, E., Alkan, A., & Akıcı, A. (2017). DİŞ HEKİMLİĞİ FAKÜLTELERİNDE VE DİĞER AĞIZ DİŞ SAĞLIĞI KURUMLARINDA ANTİBAKTERİYEL REÇETELENME DURUMLARININ ARAŞTIRILMASI. RELATION, 25, 34.

Weibrecht, N., Endel, F. and Zechmeister, M. (2019) "Evaluating ATC-ICD: Assessing the relationship between selected medication and diseases with machine learning", International Journal of Population Data Science, 4(3). doi: 10.23889/ijpds.v4i3.1292.

Croset, S., Hoehndorf, R., & Rebholz-Schuhmann, D. (2012). Integration of the anatomical therapeutic chemical classification system and drugbank using owl and text-mining. GI Workgroup Ontologies in Biomedicine and Life Sciences (OBML).

Chen L, Zeng W-M, Cai Y-D, Feng K-Y, Chou K-C (2012) Predicting Anatomical Therapeutic Chemical (ATC) Classification of Drugs by Integrating Chemical-Chemical Interactions and Similarities. PLoS ONE 7(4): e35254. https://doi.org/10.1371/journal.pone.0035254

Iorio, F., Bosotti, R., Scacheri, E., Belcastro, V., Mithbaokar, P., Ferriero, R., ... & di Bernardo, D. (2010). Discovery of drug mode of action and drug repositioning from transcriptional responses. Proceedings of the National Academy of Sciences, 107(33), 14621-14626.

Reinecke, I., Siebel, J., Fuhrmann, S., Fischer, A., Sedlmayr, M., Weidner, J., & Bathelt, F. (2023). Assessment and Improvement of Drug Data Structuredness From Electronic Health Records: Algorithm Development and Validation. JMIR medical informatics, 11, e40312. https://doi.org/10.2196/40312

Gnjidic, D., Pearson, S. A., Hilmer, S. N., Basilakis, J., Schaffer, A. L., Blyth, F. M., Banks, E., & High Risk Prescribing Investigators (2015). Manual versus automated coding of free-text self-reported medication data in the 45 and Up Study: a validation study. Public health research & practice, 25(2), e2521518.

https://doi.org/10.17061/phrp2521518

Edmundson, H. P., & Wyllys, R. E. (1961). Automatic abstracting and indexing—survey and recommendations. Communications of the ACM, 4(5), 226-234.
VICKERY, B.C. (1962), "CLASSIFICATION FOR DOCUMENTATION", Aslib Proceedings, Vol. 14 No. 8, pp. 243-247. https://doi.org/10.1108/eb049889

Nanni, L., Brahnam, S., & Maguolo, G. (2020). Anatomical therapeutic chemical classification (ATC) with multi-label learners and deep features. International Journal of Natural Computing Research (IJNCR), 9(3), 16-29.

Peng, Y., Wang, M., Xu, Y., Wu, Z., Wang, J., Zhang, C., ... & Tang, Y. (2021). Drug repositioning by prediction of drug's anatomical therapeutic chemical code via network-based inference approaches. Briefings in bioinformatics, 22(2), 2058-2072.

Cheng, X., Zhao, S. G., Xiao, X., & Chou, K. C. (2017). iATC-mHyb: a hybrid multi-label classifier for predicting the classification of anatomical therapeutic chemicals. Oncotarget, 8(35), 58494–58503. https://doi.org/10.18632/oncotarget.17028

C. -C. Hsu, P. -C. Chang and A. Chang, "Multi-Label Classification of ICD Coding Using Deep Learning," 2020 International Symposium on Community-centric Systems (CcS), Tokyo, Japan, 2020, pp. 1-6, doi: 10.1109/CcS49175.2020.9231498.

Blanco, A., Perez-de-Viñaspre, O., Pérez, A., & Casillas, A. (2020). Boosting ICD multi-label classification of health records with contextual embeddings and label-granularity. Computer methods and programs in biomedicine, 188, 105264.

Pascual, D., Luck, S., & Wattenhofer, R. (2021). Towards BERT-based automatic ICD coding: Limitations and opportunities. arXiv preprint arXiv:2104.06709.

Arifoğlu, D., Deniz, O., Aleçakır, K., & Yöndem, M. (2014). CodeMagic: semi-automatic assignment of ICD-10-AM codes to patient records. In Information Sciences and Systems

2014: Proceedings of the 29th International Symposium on Computer and Information Sciences (pp. 259-268). Springer International Publishing.

A. Çelikten and H. Bulut, "Turkish Medical Text Classification Using BERT," 2021 29th Signal Processing and Communications Applications Conference (SIU), 2021, pp. 1-4, doi: 10.1109/SIU53274.2021.9477847

Ozsonmez, D. B., & Acarman, T. (2023, February). A Case Study: Disease Code (ICD-10) Classification in Turkish Medical Summary Dataset. In International Conference on Intelligent Sustainable Systems (pp. 537-545). Singapore: Springer Nature Singapore.

Madabushi, H. T., Kochkina, E., & Castelle, M. (2020). Cost-sensitive BERT for generalisable sentence classification with imbalanced data. arXiv preprint arXiv:2003.11563

Hugging Face. (n.d.). dbmdz/bert-base-turkish-cased. Hugging Face. https://huggingface.co/dbmdz/bert-base-turkish-cased

Akın, A. A. (2017). AHMETAA/Zemberek-NLP: NLP tools for Turkish. GitHub. https://github.com/ahmetaa/zemberek-nlp/tree/master

Scikit-learn. (2023). Support Vector Classification (SVC). scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

A. Akdemir, T. Shibuya and T. Güngör, "Subword Contextual Embeddings for Languages with Rich Morphology," 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 2020, pp. 994-1001, doi: 10.1109/ICMLA51294.2020.00161.

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

Schweter, S. (2020). BERTurk - BERT models for Turkish (Version 1.0.0). Zenodo. https://doi.org/10.5281/zenodo.3770924

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

Jiang, Z. H., Yu, W., Zhou, D., Chen, Y., Feng, J., & Yan, S. (2020). Convbert: Improving bert with span-based dynamic convolution. Advances in Neural Information Processing Systems, 33, 12837-12848.

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

APPENDICES

Appendix A. Turkish Medical Summary Example

İnterferon ve interferon tedavisinde son gelişmelerin gözden geçirilmesi.

İnterferon antiviral aktivitesi olan glikoprotein yapısında bir maddedir.Son yıllarda saflaştırılması,farmakolojik etki ve yan etkilerinin,etki spektrumunun belirlenmesi önemli adımlar atılmıştır.Viral ve maliğn hastalıklarda klinik uygulamalar artmıştır.

Varisella zoster,diğer herpes grubu viruslar,sitomegalovirus,Epstein-Bar virusu,hepatit B virusu enfeksiyonları ve solunum yolları virus enfeksiyonlarında interferon tedavisi ile yüz güldürücü sonuçların alındığı belirtilmiştir.Bu yazımızda,güncel bir konu olan,interferon ve tedavisindeki gelişmeler gözden geçirildi.

Appendix B. Table for ATC Code Explanations & Counts in Turkish Medical Drug Manuals Dataset

Table Appendix B.1: Turkish Medical Drug Manuals Dataset

ATC 2nd Level	Anatomical Groups	Count
J01	Antibacterials for systemic use	1453
L01	Antineoplastic Agents	744
C09	Renin-Angiotensin System Agents	717
N06	Psychoanaleptics	619
N05	Psycholeptics	546
R03	Obstructive Airway Diseases	516
M01	Anti-inflammatory and Antirheumatic Products	461
A10	Diabetes Drugs	427
N03	Antiepileptics	393
B05	Blood Substitutes & Perfusion Solutions	366
N02	Analgesics	342
R05	Cough & Cold Preparations	326
S01	Ophtalmologicals	303
C10	Lipid Modifying Agents	292
A02	Drug for Acid Related Disorders	290
L04	Immunosuppresants	257
G04	Urologicals	230
B01	Antithrombotic Agents	220
J05	Antivirals for Systemic Use	202
M03	Muscle Relaxants	187

V03	All Other Therapeutic Products	172
A11	Vitamins	171
R06	Antihistamines for Systemic Use	166
B02	Antihemorrhagics	160
D01	Antifungals for Dermatological Use	155
B03	Antianemic Preparations	155
M05	Drugs for Bone Diseases	152
C01	Cardiac Therapy	149
J02	Antimycotics for Systemic Use	144
A03	Functional Gastrointestinal Disorders	140
M02	Topical Products for Joint & Muscular Pain	139
C07	Beta Blocking Agents	138
N04	Anti Parkinson Drugs	133
D07	Corticosteroids, Dermatological Preparations	133
G03	Sex Hormones & Modulators of Genital System	128
N01	Anesthetics	122
A12	Mineral supplements	118
R01	Nasal Preparations	116
C08	Calcium Channel Blockers	105
J06	Immune Sera & Immunoglobulins	104
A04	Antiemetics & Antinauseants	96
A07	Antidiarrheals, Intestinal Anti-inflammatory	85
V08	Contrast Media	79
A01	Stomatological Preparations	79
N07	Other Nervous System Drugs	79
D06	Antibiotics/Chemotherapeutics for Dermatological Use	69
G01	Gynecological Antiinfectives & Antiseptics	69
J07	Vaccines	67
A06	Constipation Drugs	66
L02	Endocrine Therapy	65
D10	Anti acne Preparations	64
A16	Other Alimentary Tract & Metabolism Products	62
H02	Corticosteroids for Systemic Use	60
C03	Diuretics	59
H01	Pituitary & Hypothalamic Hormones & Analogues	59
C02	Antihypertensives	57
D11	Other Dermatological Preparations	55
H05	Calcium Homeostasis	54
C05	Vasoprotectives	53
L03	Immunostimulants	46
A05	Bile & Liver Therapy	39
R02	Throat Preparations	29
H03	Thyroid Therapy	29
P01	Antiprotozoals	28
D03	Treatment of Wounds & Ulcers	27
D04	Antipruritcs incl. Antihistamines & Anesthetics	27
G02	Other Gynecologicals	22
J04	Antimycobacterials	21

Appendix C. Turkish Drug Manual Example

KISA ÜRÜN BİLGİSİ

1. BEŞERİ TIBBİ ÜRÜNÜN ADI

ABILIFY 10 mg tablet

2. KALİTATİF VE KANTİTATİF BİLEŞİM

Etkin madde:
Aripiprazol 10 mg

Yardımcı maddeler:
Laktoz monohidrat (sığır sütü) 62,18 mg

Yardımcı maddeler için 6.1'e bakınız.

3. FARMASÖTİK FORM

Tablet.
Bir tarafına "A-008" ve "10" basılmış pembe, modifiye dikdörtgen tablet.

4. KLİNİK ÖZELLİKLER

4.1 Terapötik endikasyonlar
ABILIFY yetişkin ve ergenlerde (13-17 yaş) şizofreni tedavisinde (akut şizofreni epizodlarının tedavisinde ve idame tedavisi sırasında klinik düzelmenin devamlılığında) endikedir.

ABILIFY yetişkinlerde Bipolar I Bozuklukla ilişkili akut manik epizodların tedavisinde ve son epizodu manik ya da karma olan bipolar I hastalarında stabilitenin sağlanması ve reküransın önlenmesinde endikedir.

ABILIFY antidepresan tedaviye dirençli majör depresyon hastalarında antidepresan tedaviyi güçlendirmek için ekleme tedavisi olarak endikedir.

ABILIFY pediyatrik hastalarda (6 -17 yaş) başkalarına karşı agresyon, kendini kasıtlı olarak yaralama, öfke nöbetleri ve ruh halinin hızla değişmesi semptomları da dahil olmak üzere, otistik bozukluk ile ilişkili irritabilitenin tedavisinde endikedir.

4.2 Pozoloji ve uygulama şekli
Pozoloji / uygulama sıklığı ve süresi:
Yetişkinlerde
Şizofrenide
ABILIFY'ın önerilen başlangıç dozu öğünlerin zamanı dikkate alınmaksızın günde tek doz verilen 10 mg/gün veya 15 mg/gün'dür. ABILIFY'ın idame dozu günde 15 mg'dır. Klinik çalışmalarda ABILIFY'ın 10-30 mg/gün doz aralığında etkili olduğu gösterilmiştir. Günlük maksimum doz 30 mg'ı aşmamalıdır.

Bipolar Manide
ABILIFY, öğünlerin zamanı dikkate alınmaksızın günde tek doz olarak verilmelidir, başlangıç dozu genellikle günde 15 veya 30 mg'dır. Eğer gerekliyse, doz ayarlaması 24 saatten daha kısa sürede yapılmamalıdır. Antimanik etkililiği (3-12 hafta) 15-30 mg/gün doz aralığı için klinik