REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ANALYSIS OF CUSTOMER FEEDBACK REPLIED TO AN AIRLINES SUPPORT ACCOUNT VIA TWITTER WITH NATURAL LANGUAGE PROCESSING METHODS

Tolga KURTULUŞ

MASTER OF SCIENCE THESIS
Department of Statistics
Statistics Program

Supervisor
Assist. Prof. Dr. Erhan ÇENE

June, 2023

REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ANALYSIS OF CUSTOMER FEEDBACK REPLIED TO AN AIRLINES SUPPORT ACCOUNT VIA TWITTER WITH NATURAL LANGUAGE PROCESSING METHODS

A thesis submitted by Tolga KURTULUŞ in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 21.06.2023 in Department of Statistics, Statistics Program.

Assist. Prof. Dr. Erhan ÇENE
Yildiz Technical University
Supervisor

Approved By the Examining Committee

Assist. Prof. Dr. Erhan ÇENE, Supervisor
Yildiz Technical University _____

Assoc. Prof. Dr. Serpil KILIÇ DEPREN, Member
Yildiz Technical University _____

Assist. Prof. Dr. Mehmet Fatih KARAASLAN, Member
Marmara University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Analysis of Customer Feedback Replied to an Airlines Support Account via Twitter with Natural Language Processing Methods supervised by my supervisor, Assist. Prof. Dr. Erhan ÇENE. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Tolga KURTULUŞ

Signature

Dedicated to my grandfather who passed away
and my precious family

ACKNOWLEDGEMENTS

I am deeply grateful to my esteemed advisor, Dr.Öğr.Üyesi Erhan ÇENE, who provided unwavering support throughout my master's education and guided me with invaluable insights during the entire process of this research. Special thanks to my beloved life partner Ezgi AKTAN for her constant support, encouragement, and motivation in every situation and my valuable friend Alihan ÖZTÜRK for his continuous encouragement and support since the beginning of my master's education. Furthermore, I would like to thank my precious parents and valuable brother Atilla KURTULUŞ.

Tolga KURTULUŞ

TABLE OF CONTENTS

LIST OF ABBREVIATIONS vii

LIST OF FIGURES viii

LIST OF TABLES ix

ABSTRACT x

ÖZET xii

1 INTRODUCTION 1

1.1 Literature Review 1

1.2 Objective of the Thesis 3

2 NATURAL LANGUAGE PROCESSING 4

2.1 What is natural language? 4

2.2 What is Natural Language Processing(NLP)? 5

2.3 NLP Techniques 5

2.3.1 Rule-based Techniques 5

2.3.2 Statistical Approach Techniques 6

2.3.3 Machine Learning Techniques 6

2.3.4 Deep Learning Techniques 7

2.4 NLP Pipeline 13

2.4.1 Sentence Segmentation 13

2.4.2 Word tokenization 14

2.4.3 Stemming 15

2.4.4 Lemmatization 15

2.4.5 Stop word analysis 16

2.4.6 Dependency parsing 17

2.4.7 Part-of-speech (POS) tagging 17

2.5 BERT Model 18

2.5.1 BERT Model performance measurements 19

2.6 Turkish BERT Model - BERTurk 21

3 APPLICATION **23**
3.1 Data Set 23
3.2 Data Preprocessing 24
3.3 Data Modeling 26
3.4 Turkish Sequential Model 27
3.5 English Sequential Model 29
3.6 BERT and BERTurk models 32

4 CONCLUSION **34**
4.1 Reliability of the data and generalizability of the study 34

REFERENCES **36**

PUBLICATIONS FROM THE THESIS **39**

LIST OF ABBREVIATIONS

BERT	Bidirectional Encoder Representations from Transformers
DT	Decision Tree
EN	English
GLUE	General Language Understanding Evaluation
ML	Machine Learning
MLM	Masked Language Model
NLP	Natural Language Processing
NB	Naive Bayes
NN	Neural Network
RNN	Recurrent Neural Networks
SQuAD	Stanford Question Answering Dataset
SVM	Support Vector Machine
SWAG	Situations With Adversarial Generations
TDD	Turkish Language Depository
THY	Turkish Airlines
TPU	Tensor Processing Unit
TR	Turkish
URL	Uniform Resource Loader

LIST OF FIGURES

Figure 2.1 Simple Neural Network infographic 8
Figure 2.2 RNN Architecture 9
Figure 2.3 RNN "Hello" example 11
Figure 2.4 Transformers architecture example 12
Figure 2.5 Tokenization example from OpenAI Tokenizer 14
Figure 2.6 BERT Model Infographic 18
Figure 2.7 BERT Performance on SQUAD benchmark 20
Figure 2.8 BERT Performance on SWAG benchmark 20
Figure 2.9 BERT Performance on GLUE benchmark 21
Figure 3.1 Dataset preview on TDD.ai website 23
Figure 3.2 Sentiment labels grouped by languages 25
Figure 3.3 Flow Chart of Sequential Model Architecture 26
Figure 3.4 Turkish Sequential Model Training Results 28
Figure 3.5 English Sequential Model Training Results 30

LIST OF TABLES

Table 2.1 Table of BERTurk models 22
Table 3.1 Turkish Sequential Model Architecture 28
Table 3.2 Turkish Sequential Model Confusion Matrix Table 29
Table 3.3 English Sequential Model Architecture 30
Table 3.4 English Sequential Model Confusion Matrix Table 31
Table 3.5 BERT testing results table 32
Table 3.6 BERTurk testing results table 33
Table 4.1 BERTurk Table 34
Table 4.2 BERT English Table 35

ABSTRACT

Analysis of Customer Feedback Replied to an Airlines Support Account via Twitter with Natural Language Processing Methods

Tolga KURTULUŞ

Department of Statistics
Master of Science Thesis

Supervisor: Assist. Prof. Dr. Erhan ÇENE

Twitter.com has become a well-known social media platform that has increased its number of users in recent years and almost every internet user is either a member or is aware of it. Twitter users can interact with each other by expressing themselves through texts called "tweets" and sharing photos/videos embedded in these tweets. Apart from individual users, corporate companies have also taken their place on Twitter, and many support departments have opened accounts to communicate directly with their customers. In this study, the tweets of an airline company interacting with passengers through its Twitter account were used as the dataset of the study. The feedbacks and responses to the support account were filtered to be used only in Turkish and English languages and the sentiment of the tweet was labeled for both languages. In the labeling process, 3 different sentiment categories were identified and manual labeling was performed to indicate whether the tweet was "positive", "negative" or "neutral". This list of tweets was analyzed with natural language processing methods for both languages and the success of classifying the sentiments of the tweets was measured. The modeling study was carried out in R programming language with the help of the TensorFlow library and simplified the word groups in the tweets with the help of many operations such as filtering alphanumeric characters, cleaning emojis and website links, and cleaning stopwords. After the data cleaning steps, a 3-layer Artificial Neural Network architecture was created to build a natural language processing model. Two natural language processing models were trained separately for each

language, Turkish and English. When the trained natural language processing models were tested, a 94.04% successful prediction rate for classifying Turkish tweets and a 93.69% prediction rate for English tweets were calculated. To understand the generalizability of the model result metrics, they were compared with the pre-trained BERT-Turkish (uncased) (for Turkish) and BERT (uncased) (for English) models to find the most optimal approach for natural language processing studies.

Keywords: Twitter, neural language processing, model training, sentiment analysis, BERT models

ÖZET

Bir Havayolu Şirketinin Destek Hesabına Twitter Üzerinden İletilen Müşteri Geri Bildirimlerinin Doğal Dil İşleme Yöntemleri ile Analizi

Tolga KURTULUŞ

İstatistik Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Assist. Prof. Dr. Erhan ÇENE

Twitter.com son yıllarda kullanıcı sayısını artıran ve hemen hemen her internet kullanıcısının ya üye olduğu ya da haberdar olduğu tanınmış bir sosyal medya platformu haline gelmiştir. Twitter kullanıcıları "tweet" adı verilen metinlerle kendilerini ifade ederek ve bu tweetlerin içine yerleştirdikleri fotoğraf/videoları paylaşarak birbirleriyle etkileşime geçebilmektedir. Bireysel kullanıcıların yanı sıra kurumsal şirketler de Twitter'da yerini almış, birçok destek departmanı birimi hesap açarak müşterileriyle doğrudan iletişim kurmayı hedeflemiştir. Bu çalışmada, bir havayolu şirketinin Twitter hesabı üzerinden yolcularla etkileşime geçtiği tweetler, çalışmanın hedef veri seti olarak kullanılmıştır. İlgili destek hesabına gelen geri bildirimler ve cevaplar sadece Türkçe ve İngilizce dillerinde kullanılmak üzere filtrelenmiş ve her iki dil için de belirlenen tweetin duygusu etiketlenmiştir. Etiketleme işleminde 3 ayrı duygu kategorisi belirlenmiş ve tweetin "olumlu", "olumsuz" ya da "nötr" olduğunu belirtmek için manuel etiketleme yapılmıştır. Kullanılan bu tweet listesi, her iki dil için doğal dil işleme yöntemleri ile incelenmiş ve tweetlere ait duyguları sınıflandırma başarısı ölçülmüştür. Modelleme çalışması TensorFlow kütüphanesi yardımıyla R programlama dili ile gerçekleştirilmiş olup, alfanümerik karakterleri filtreleme, emoji ve website linklerinin temizliği, etkisiz kelimelerin temizliği gibi birçok işlem yardımı ile tweetler içinde yer alan kelime gruplarının sadeleştirilmesi sağlanmıştır. Veri temizlik adımlarının ardından doğal dil işleme modeli kurmak için 3 katmanlı bir Yapay Sinir Ağı mimarisi oluşturulmuştur. Türkçe ve İngilizce olmak üzere her iki dil için

ayrı ayrı iki doğal dil işleme modeli eğitilmiştir. Eğitilen doğal dil işleme modelleri test edildiğinde, Türkçe tweetleri sınıflandırma konusunda %94.04'lük başarılı tahmin oranına sahipken İngilizce tweetler için %93.69'lük tahmin oranı hesaplanmıştır. Model sonuç metriklerinin genellenebilirliğini anlamak için önceden eğitilmiş BERT-Turkish (uncased) (Türkçe için) ve BERT (uncased) (İngilizce için) modelleri ile karşılaştırılmış ve doğal dil işlemesi çalışmaları için mümkün olan en optimal yaklaşımın bulunması hedeflenmiştir.

Anahtar Kelimeler: Twitter, doğal dil işleme, model eğitimi, duygu analizi, BERT modelleri

1
INTRODUCTION

1.1 Literature Review

Within the scope of this thesis, the literature review was conducted primarily on research examining customer feedback through e-commerce and social media channels. Studies in the fields of machine learning and deep learning, which are used in sentiment analysis studies, are also examined on various social media platform datasets. In addition, industry-specific hybrid methods and machine learning-based studies, as well as studies comparing performance with pre-trained models, are examined.

To respond effectively to customer reviews, companies need to identify consumer review platforms, understand their characteristics, and continuously evaluate their impact on consumer purchase decisions. According to research conducted to make sense of user behavior prior to the purchase of a product or service, 98% of the sample of the target population read reviews prior to a purchase, while 60% said they read them frequently or fairly frequently [1].

A study of 568,454 product reviews of 74,258 products and 256,059 users that Amazon, a popular e-commerce company, offered for sale over a ten-year period revealed how sentiment analysis can help identify consumer behavior and overcome these risks to meet consumer satisfaction [2]. Another study on the feedback received from the Twitter source, such as the data set included in the study, explained that classifying the feedback as positive or negative, it would serve to make sense of the target company by collecting comments from users around the world [3]. A study on Facebook, which evaluated users who follow Turkish Airlines' page, proved that social networks can be used to improve revenue-generating indicators such as customer satisfaction and purchase likelihood in a very short period of time [4]. The findings of this study provided explanations for customer behavior and showed that most of the company's Facebook users were highly likely to purchase a Turkish Airlines ticket.

Natural language processing methods, called NLP, are defined as the processing of spoken language used by humans during communication with each other, and its meaningful transfer to computers and interpretation with mathematical formulas [5]. With the usage of these methods, any specific language can be easily modeled by computers, regardless of the restrictions in the language family to which the spoken language belongs. Since customer feedback is of a type that can be written by any user worldwide, regardless of the written language, there is a need to model it with an approach that has the ability to make sense of every language [6]. In addition, the scarce availability of multilingual annotated datasets limits the progress of multilingual and cross-lingual sentiment analysis and makes it difficult to compare these techniques to determine the best-performing approach.

Sentiment analysis is defined as the automatic identification of emotion, opinion and subjectivity in a given text or content [7]. In a sentiment analysis study on airline service feedback, several Machine Learning (ML) algorithms such as Naive Bayes (NB), Support Vector Machine (SVM), and Decision Tree (DT) were used and each of these approaches produced different results [8]. In a study, deep autoencoders were trained with a semi-supervised approach and achieved successful results in sentiment analysis [9]. In another study, an approach that performs sentiment analysis on Twitter using convolutional neural networks is discussed. In this study, a simple single-level convolutional neural network was used in a three-stage approach with different learning methods at each stage [10].

Another study was conducted on satirical news identification task in Turkish news articles [11]. In the classification phase of this study, accuracy rates of five supervised learning algorithms (Naive Bayes, Logistic Regression, Support Vector Machines, Random Forest and k-nearest neighbor algorithm) with three widely utilized ensemble methods (i.e. AdaBoost, bagging, and random subspace) have been considered. Based on the results, the study proved that the Random Forest Algorithm resulted in the highest performance, with a classification accuracy of 96.92% for satire detection in Turkish. For deep learning-based architectures, the classification accuracy of 97.72% with the recurrent neural network architecture with attention mechanism is achieved.

1.2 Objective of the Thesis

Many companies and organizations have recently been using Sentiment Analysis methods to better understand their customers and accurately understand their needs. These analytical methods enable the interpretation of customer attitudes and behaviors through statistical approaches. Customers often share their experiences and feelings about a company's product or service in written and/or visual texts in a positive or negative way. These posts generally aim to inform other users about different qualities such as the quality of the service received, the durability of the product, and the price-performance ratio. In this way, users who research the product on the internet can benefit from the experiences of previous users and experience a more effective purchasing process. Feedback both before and after the service purchase also plays an important role in user experience sharing. Especially in cases where the service received is disrupted or incomplete, the dialogs established with the support and assistance units provide the most comprehensive customer opinion about the service. Such feedback and support processes are mostly carried out through social media channels, internet forums and complaint-sharing platforms. Twitter.com, which has recently grown in popularity, has become a social media platform that almost every internet user is a member or is aware of. Users can express themselves with texts called "tweets" and share photos/videos, allowing interaction between members. Corporate firms as well as individual users have taken their place on Twitter and have opened many support line/unit accounts to communicate directly with their customers. Corporate firms, which receive comments from their customers on products/services through these accounts, are almost as prominent as popular users in actively using social media. At this point, many airline companies that took place on Twitter via support accounts, receive thousands of customer comments and feedback daily. These comments can be either negative for many different reasons such as delayed flights and slow baggage delivery processes, or positive such as new flight destination announcements, promotional ticket campaigns, in-flight catering service quality. In addition to these positive and negative comments, there are also neutral feedbacks that do not contain any emotion in the meaning of the sentence.

The main objective of this thesis is to use a machine learning model based on natural language processing techniques to analyze customer feedback from THY, a valuable company in our country, in order to support the business operations of the Customer Relations units and to advance previous research in this area with fresh perspectives. This study also evaluates the performance of machine learning models based on languages and contrasts the pre-trained machine learning models BERT[12] and BERTurkish[13].

2
NATURAL LANGUAGE PROCESSING

2.1 What is natural language?

Natural languages can be defined as language rules created on words formed with the help of an alphabet consisting of letters and the use of that language among people. The most important reason for the emergence of natural languages is based on the need for people to communicate with each other. People who can establish this communication in both verbal and written formats have enabled the words of the languages they use to be enriched and transferred to other people. Natural languages, which can develop themselves over time, can be affected by the vocabulary of the language in different ways when exposed to the influence of other spoken/written languages. As a result of this influence, each language has experienced different developments over time when evaluated in terms of the society in which it is spoken. Thanks to technological advances in today's world, the use of natural languages in computer environments has become widespread. In particular, the widespread use of the internet in recent years and the more instantaneous communication between internet users has enabled spoken languages to be stored in text form by computers. The alphabet of 0 and 1, which is called computer language and is represented by the on and off of electrical signals, is the basis of the communication established with the help of computers. As the storage and processing units in computers have developed over the years and have higher memory, the processing time of computer language has shortened. As a result of this shortening of time and technological advances, the spoken languages used among users can be collected by computers in databases. Thanks to databases where word and sentence-based text content can be stored, different business problems have emerged over the years. With the help of statistical and mathematical calculations, these problems are aimed to be solved as a result of analytical processes and the research field called "natural language processing" has appeared.

2.2 What is Natural Language Processing(NLP)?

Natural Language Processing (NLP) is a growing field of study that aims to improve the way humans interact with computers by leveraging the power of human language. This interdisciplinary domain seeks to improve communication not only between individuals but also across linguistic boundaries. There has been an increase in the number of comprehensive studies and investigations aimed at advancing Natural Language Processing in recent years.

Researchers have used multiple strategies to address the complex challenges of natural language understanding and generation, combining deep learning techniques, machine learning algorithms, statistical analysis methods, and rule-based systems. This hybrid methodology enables the creation of robust and versatile NLP solutions capable of comprehending and producing human-like language expressions.

Natural language processing research covers a wide range of academic disciplines as well as practical applications. NLP provides a comprehensive framework for addressing a wide range of linguistic issues, from simple tasks like spelling correction to complex endeavors like machine translation systems. Language learning applications also benefit from NLP techniques, allowing learners to learn new languages more effectively. Furthermore, personal assistant applications use NLP to enable seamless human-computer interactions, allowing users to communicate their intentions, receive relevant information, and perform tasks with ease.

Natural Language Processing, in short, acts as a vital link between humans and computers, enabling more intuitive and efficient communication. As research in this field advances, we can expect significant breakthroughs that will revolutionize the way we interact with technology and leverage the potential of human language in a variety of domains.

2.3 NLP Techniques

2.3.1 Rule-based Techniques

Rule-based methodologies in Natural Language Processing (NLP) include the development of linguistic rules and patterns that help in the processing of natural language. These techniques are extremely useful for tasks such as spelling correction and grammar checking. Rule-based systems analyze and manipulate linguistic structures using pre-designed rules, providing comprehensibility and

control in NLP applications. Notably, this approach has the advantage of being relatively simple and efficient to implement. However, certain limitations must be acknowledged, such as the inability to capture the contextual semantics of words and the proclivity for misjudgment when confronted with instances of sarcasm or irony.

2.3.2 Statistical Approach Techniques

The use of statistical methodologies in the field of Natural Language Processing (NLP) involves examining linguistic data using probability theory and statistical inference techniques. This approach mostly depends on finding patterns in large volumes of text. By recognising these trends, the system can develop its own understanding of human language. Statistics dependent approaches make it easier to estimate language models, which are critical in determining the probability of word sequences. In turn, such models enable the completion of a variety of tasks, including but not limited to language generation and speech recognition. Furthermore, statistical analysis methods help in the advancement of machine translation systems by facilitating parallel corpus alignment and the identification of translation probabilities.

2.3.3 Machine Learning Techniques

Machine learning algorithms allow computers to learn patterns and relationships from large amounts of data, without being explicitly programmed. This ability makes ML an ideal approach for NLP tasks, where the complexity and ambiguity of human language make traditional rule-based methods challenging. ML provides the foundation for many NLP applications by enabling computers to learn from data and make predictions or decisions without being explicitly programmed. Various machine learning methodologies have been applied to NLP problems, revolutionizing the field and enabling significant advancements in areas such as text classification, sentiment analysis, machine translation, question-answering systems, and chatbots.

2.3.3.1 Supervised Learning

Supervised learning involves training a model with a labeled dataset, in which the input data is accompanied by predetermined labels or categories. This method is especially useful for creating models that tackle classification and regression problems, allowing for variable prediction based on established patterns. Explanatory variables enable the development of models capable of addressing

regression-based variable prediction and classification tasks. In the field of natural language processing (NLP), supervised learning techniques can be applied to a wide range of scenarios, from simple tasks like distinguishing between spam and non-spam messages to more complex tasks like sentiment analysis. It is now possible to train a model to accurately extract the sentimental polarity of a given text using supervised learning on tweet datasets, for example. To achieve the desired results in NLP applications, well-established supervised learning algorithms such as support vector machines (SVM) and decision trees are also widely used.

2.3.3.2 Unsupervised Learning

Unsupervised learning, a significant branch of machine learning, involves training models with unlabeled data without any explicit guidance. This approach allows algorithms to autonomously discover intricate patterns and structures embedded in data, far more than the capabilities of supervised learning. This type of learning has gained significant traction, particularly in studies involving non-tabular data types such as audio, image, and video, which has caused a recent surge in scientific research endeavors. Unsupervised learning has emerged as a popular methodology in the field of natural language processing (NLP), with applications in a variety of tasks such as document clustering, topic modeling, and word embeddings. Latent Dirichlet Allocation (LDA) and word2vec are two popular examples of unsupervised learning techniques used in NLP research. Because of their effectiveness in extracting meaningful representations from unstructured textual data, these techniques have received a great deal of attention and recognition.

2.3.4 Deep Learning Techniques

Deep learning has evolved as an extremely powerful and significant technique in Natural Language Processing (NLP) research. It has aided in the development and building of models with the amazing ability to autonomously ingest and understand complex linguistic representations. Notably, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which include iterations such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), have been widely used in a variety of tasks ranging from sentiment analysis to named entity recognition to machine translation. These models are unrivaled in their capacity to grasp and encapsulate contextual information while also exploiting the deep sequential connections hidden within textual data.

2.3.4.1 Neural Networks

Neural Networks(NN) technique, which is used in many studies in the fields of Statistics and Computer Science, has imitated the functioning of our brain and is based on the structure of the brain. There are small dots called neurons placed in the human brain and through these dots, information flow is provided in our brain. Neurons form a network by sending and receiving signals to one another in this information flow where these networks are known as neural networks. In scientific studies, NN is a series of methods in which the sequence of variables obtained according to their weights on the target variable and the explainability of the target variable is calculated. In its most general form, NN methods are created by extracting the attributes of computer content such as audio, text, visual, and video as input in layers. The "Input Layer" layer formed with the help of these attributes is the first step of NN studies. By using the right approaches in this step, a successful NN model that can explain and predict the target variable can be established. In the "Hidden Layer" layer, the vector weights with which the attributes in the first layer individually represent the predicted target variable are calculated numerically. With the weighting method, the attributes are intended to represent the target variable both among themselves and individually, so that the information can be completely covered by the model. As a result of these processes, the "Output Layer" layer, which is the last step, contains the target variable and produces an output as a result of the calculations of the previous layers. In NN modeling studies, although the first and last layers are always created only once, the number of "Hidden Layer" layers can be increased depending on the scope of the study, the scalability of the data and many other reasons. Neural Network models can be basically illustrated at Figure 2.1.

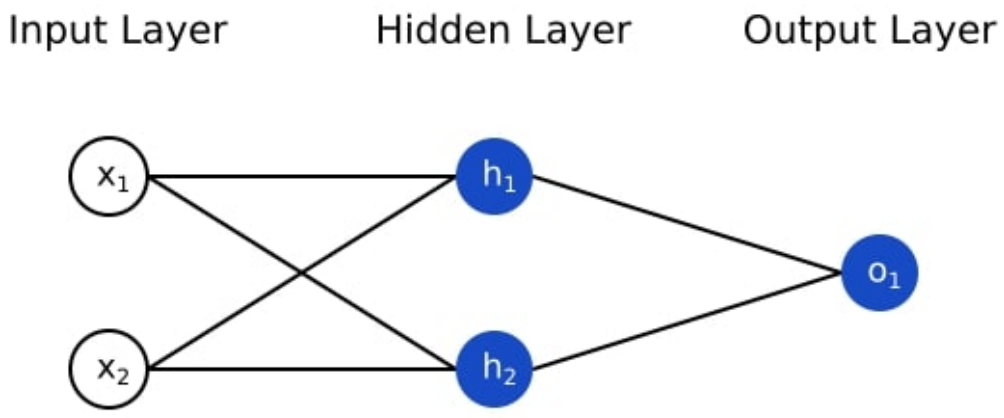


Figure 2.1 Simple Neural Network infographic

2.3.4.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are series-based inclusive forecasting models based on Neural Network infrastructure. RNN works by performing a "repetition" operation on Neural Networks, as can be understood from the meaning of the word "recurrent". RNNs are artificial neural networks that produce a new output by taking both the input and the output of the previous step at each step. With the help of such neural networks, instead of a one-way prediction or weighting method, a mechanism that repeats in the correct order and time within a memory depending on the repeated observation of the attributes is based. Recurrence makes the model sensitive not only to the given input but also to the context of the input.

While modelling with RNNs, the Sequential Modelling method has been created for practical use and on a sequence basis. Sequential modelling is a machine learning method used to model sequential data. Sequential data is data that shows a regularity over time or space. For example, data such as text, audio, video and stock prices can be given as examples of sequential data.

The success of RNN models built using such sequential data also enables to gather very significant results depending on the quality of the data and its applicability according to the recurrence logic. Schematically, RNN models are represented as shown in Figure 2.2 below.

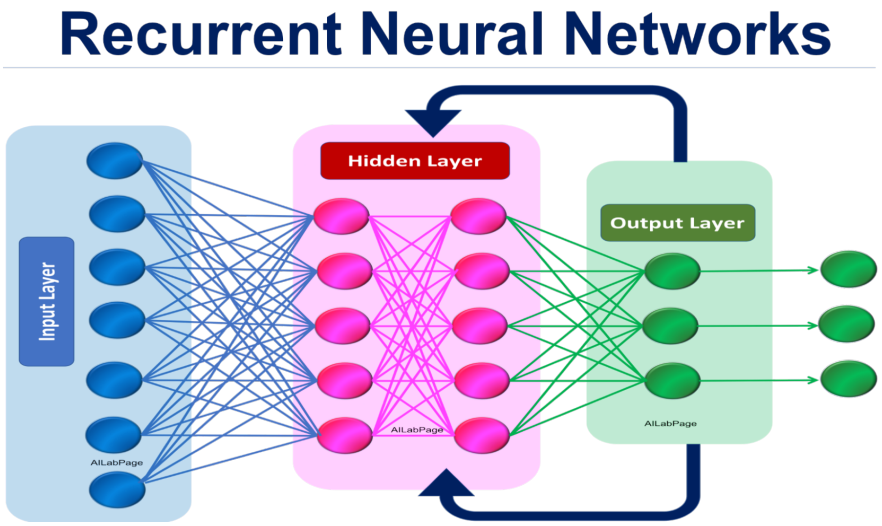


Figure 2.2 RNN Architecture
[14]

To better understand the concept of Recurrence and the Sequential Modelling method, a simple example can be given as follows. Let's say we have an RNN model that is intended to generate text. In each step, our model generates a new

character and outputs it as output. With the help of the recurrence mechanism of this model, it can remember the characters it has generated before and generate new characters accordingly. For example, if our model wants to generate the word "Hello", it can work as follows:

- First step: The model randomly generates the character "H" and outputs it.
- Second step: The model remembers the character "H" and generates the character "e" accordingly and outputs it.
- Third step: The model remembers the character "He" and accordingly generates the character "l" and outputs it.
- Step four: The model remembers the character "Hel" and accordingly generates the character "l" again and outputs it.
- Step five: The model remembers the "Hell" characters and accordingly generates the character "o" and outputs it.

In this way, the model generates new characters in a logical way by following the full text. As explained in the image given below, the characters "h,e,l,l" are represented as the "Input Layer" vectorizing the characters and weighting the characters that may come after them as "e,l,l,l,o" respectively with the help of the model.

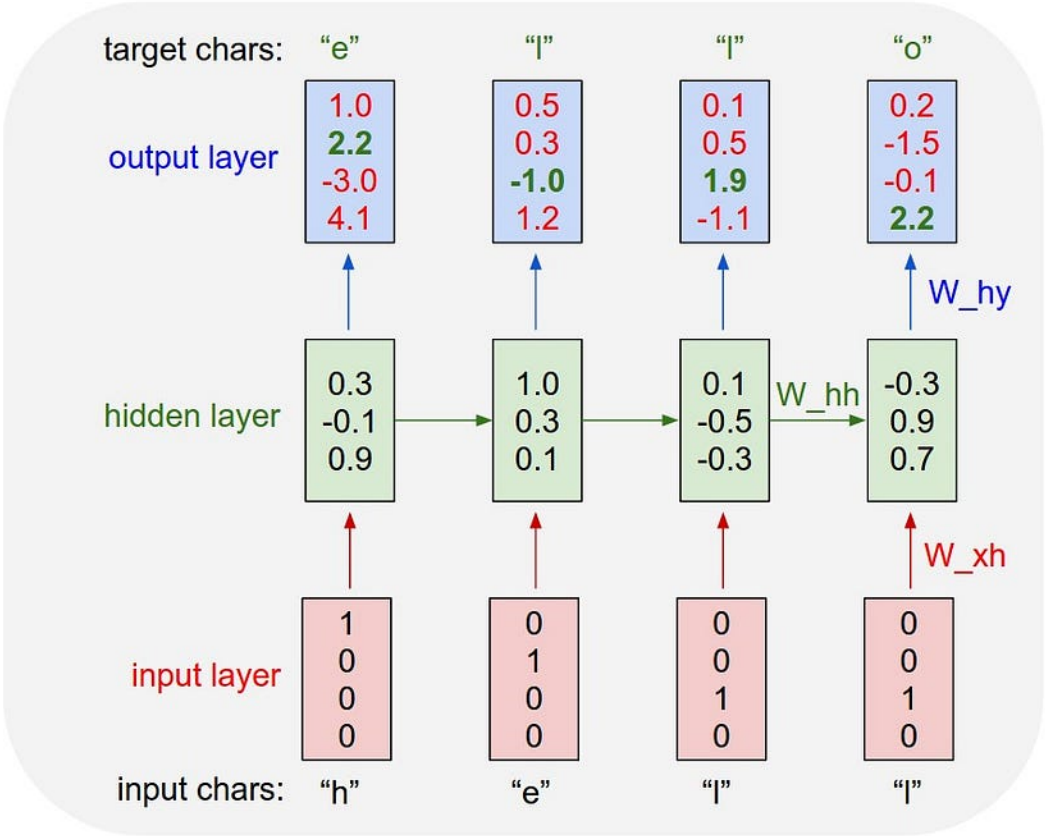


Figure 2.3 RNN "Hello" example
[15]

An example of RNN with 4-dimensional input and output layers, and a hidden layer of 3 units (neurons) is figured in Figure 2.3. This diagram shows the activations in the forward pass when the RNN is fed the character "hell" as input. The output layer contains confidences the RNN assigns for the next character (vocabulary is "h,e,l,o")

2.3.4.3 Transfer Learning Methods

Transfer learning is a sophisticated approach that encompasses two main steps: pre-training models on extensive datasets and subsequently fine-tuning them to suit specific natural language processing (NLP) tasks. By utilizing this technique, researchers and practitioners are able to exploit the acquired knowledge from one task or domain and effectively apply it to another closely related task or domain. This process has proven to be highly advantageous in the field of NLP, particularly in the context of large-scale text corpora. Notably, one widely recognized model that has significantly contributed to the success of transfer learning is BERT, an acronym for Bidirectional Encoder Representations from Transformers. This pre-trained model differs from traditional language models by using a different

approach to capture the context of the entire input sentence, rather than processing the input sentence from left to right or right to left, which is called Transformer in the literature. The Transformer method is developed and based on the "Attention is All You Need" research paper [16].

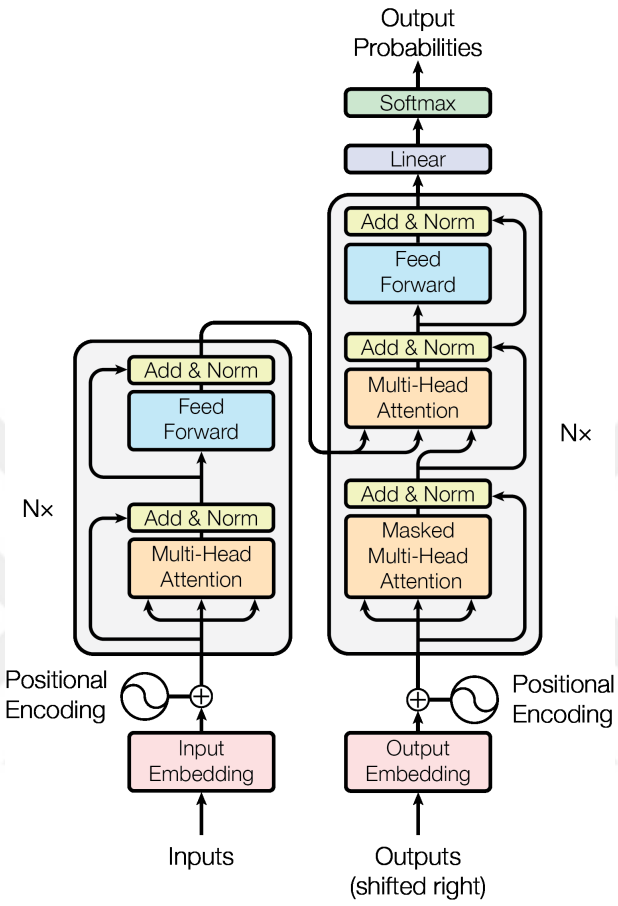


Figure 2.4 Transformers architecture example

This method is a network architecture that is entirely dependent on attention processes, with no repetition or convolutions. It computes representations of its input and output via a self-attention technique rather than sequence-aligned RNNs or convolutional layers. The model's self-attention mechanism enables it to attend to different locations in the input sequence in order to calculate a representation of that sequence. This enables the model to better recognize long-term dependencies between input and output sequences than typical RNNs or convolutional layers. The Transformer has been demonstrated to be more parallelizable and require much less time to train than other sequence transduction models. By capitalizing on the pre-training of BERT on vast amounts of text data, it has emerged as a pivotal starting point for various NLP tasks, offering notable enhancements in both efficiency and accuracy across the board.

2.4 NLP Pipeline

The pipeline of Natural Language Processing includes a series of unique computational stages or modules that are used to process and analyze natural language input. These pipelines serve as a conceptual roadmap, guiding the preparation of raw textual information into meaningful representations that can be understandable by computers for various language-related tasks, such as text classification, information extraction, sentiment analysis, machine translation, and question-answering. Despite the fact that each language-related problem requires a different and specific approach, NLP pipelines are useful for approaching these problems effectively from the very start. By systematically applying these processing steps, NLP pipelines empower researchers and developers to unlock insights, automate language-related tasks, and create sophisticated language-driven applications.

2.4.1 Sentence Segmentation

The process of identifying and separating individual sentences within a given text or document is known as sentence segmentation. This task is critical for a variety of NLP applications because sentences serve as fundamental units of meaning and context. Accurate sentence segmentation is critical for tasks such as machine translation, text summarization, sentiment analysis, and information extraction, as it allows NLP systems to analyze and understand text at the sentence level, allowing for more nuanced and accurate language processing.

Sentence segmentation is demonstrated in the following paragraph.

"Türk Hava Yolları, Türkiye'nin ulusal bayrak taşıyıcı hava yolu şirketidir. Türk Hava Yolları'nın merkezi İstanbul'dadır. Türk Hava Yolları'nın uçuş ağı Avrupa, Orta Doğu, Uzak Doğu, Kuzey Afrika, Orta Afrika, Güney Afrika, Kuzey ve Güney Amerika'ya kadar uzanmaktadır. 2015 yılında 61,2 milyon, 2018 yılında 75,2 milyon, 2019 yılında 74,3 milyon, 2020 yılında 28,0 milyon, 2021 yılında 44,8 milyon yolcu taşımıştır." [17]

After using sentence segmentation, we get the following result:

- "Türk Hava Yolları, Türkiye'nin ulusal bayrak taşıyıcı hava yolu şirketidir."
- "Türk Hava Yolları'nın merkezi İstanbul'dadır. Türk Hava Yolları'nın uçuş ağı Avrupa, Orta Doğu, Uzak Doğu, Kuzey Afrika, Orta Afrika, Güney Afrika, Kuzey ve Güney Amerika'ya kadar uzanmaktadır."

- "2015 yılında 61,2 milyon, 2018 yılında 75,2 milyon, 2019 yılında 74,3 milyon, 2020 yılında 28,0 milyon, 2021 yılında 44,8 milyon yolcu taşımıştır."

2.4.2 Word tokenization

Word tokenization is the process of dividing a continuous stream of text, paragraphs, and sentences into individual units, known as tokens, where each token typically represents a word. These units are usually words, but they can also include punctuation marks, numbers, or other linguistic elements, depending on the specific requirements of the NLP task at hand. By breaking down the text into tokens, it enables the application of further techniques, such as part-of-speech tagging, named entity recognition, and syntactic parsing, which heavily rely on the accurate identification of individual words within a sentence or document.

For instance, the sentence 'Fly with the Airline that flies to more countries than any other airline.' would be tokenized as the list "Fly", "with", "the", "Airline", "that", "flies", "to", "more", "countries", "than", "any", "other", "airline", ".".

Besides, with the latest development of new tokenization methods that are used by GPT-3 and GPT-4 models owned by OpenAI, the same sentence can also be tokenized as shared in Figure 2.5 below.

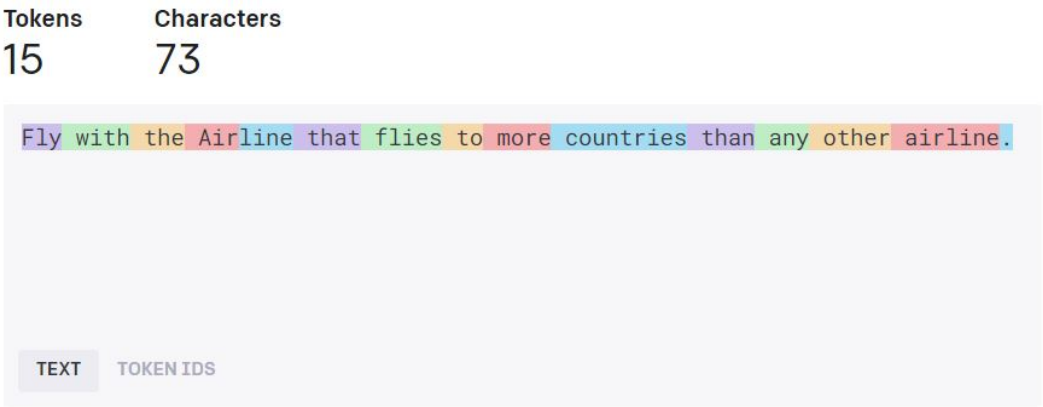


Figure 2.5 Tokenization example from OpenAI Tokenizer [18]

Word tokenization step serves as an essential preprocessing step of NLP pipeline that sets the way for more in-depth textual data analysis and comprehension. With a deeper understanding of a context, the NLP models can understand the statistical relationships between these tokens for building advanced NLP models.

2.4.3 Stemming

Stemming is a key technique for reducing words to their base or root form. It is an essential component of text preprocessing and is used in a variety of applications such as search engines, translation algorithms, and sentiment analysis.

The goal of stemming is to make textual data analysis easier by transforming words into their common morphological or linguistic variants. Stemming algorithms, for example, can convert words like "running," "runs," and "ran" into their base form, "run." By doing so, stemming enables more efficient and accurate information retrieval, particularly when variations of a word have the same or similar meanings.

However, depending on the algorithm and the language being analyzed, stemming can occasionally result in two potential issues: overstemming and understemming, both of which have the potential to change the original meaning of words. Overstemming occurs when the stemming algorithm is overly aggressive, resulting in the reduction of multiple words to the same base form. This can result in the loss of important semantic nuances and word distinctions, potentially impairing the accuracy and precision of language processing tasks. Understemming, on the other hand, occurs when the algorithm fails to adequately reduce words to their base form, resulting in related words being treated as separate entities. This can result in lower recall in information retrieval systems, where relevant documents may be missed due to word form variations.

Despite these drawbacks, stemming is still a useful technique in NLP tasks and information retrieval. Stemming allows for more efficient indexing and retrieval processes by reducing the complexity of textual data. It improves search recall and precision by capturing a broader range of related words and enables a more thorough understanding of the context in which a word is used.

2.4.4 Lemmatization

Lemmatization is a linguistic process in which words are reduced to their base or root form, known as the lemma. It is a technique used in NLP and computational linguistics to standardize and normalize textual data. Lemmatization improves text analysis, classification, and comprehension by converting words to their base form. It reduces the various inflected forms of a word to a common representation.

In contrast to stemming, which also aims to reduce words to their root form, lemmatization takes into account the context and meaning of words. It considers a language's grammatical rules, such as verb conjugations, noun plurals, and adjective

comparisons. When these factors are considered, lemmatization produces more meaningful and linguistically accurate results than simple stemming algorithms.

In NLP applications, lemmatization has several advantages. It reduces the dimensionality of text data, making analysis more efficient and effective. It can help with tasks like information retrieval, text mining, sentiment analysis, and language translation by computers. Moreover, by treating different inflected forms of a word as a single entity, lemmatized text improves the accuracy of search engines, information extraction systems, and text-to-speech synthesizers.

Lemmatization, on the other hand, is a computationally intensive process that necessitates access to extensive linguistic resources such as morphological databases or language-specific dictionaries. It is also more time-consuming than simpler techniques such as stemming. Nonetheless, because of its ability to preserve word linguistic integrity, lemmatization is a valuable tool for researchers and developers working on NLP applications that require greater language understanding and precision.

2.4.5 Stop word analysis

Stop word analysis is an important method in NLP that includes identifying and removing frequently used words known as "stop words" from a given text. Articles (e.g., "a," "an," "the"), prepositions (e.g., "on," "in," "at"), and conjunctions (e.g., "and," "but," "or") are examples of stop words.

The goal of stop-word analysis is to remove words that have little semantic meaning and contribute little to the overall understanding of the text. By eliminating these words, the emphasis shifts to more meaningful and important terms, improving the efficiency and accuracy of various NLP tasks such as text classification, information retrieval, and sentiment analysis.

Stop word analysis usually involves creating a predefined list of stop words based on the target language and NLP application requirements. The stop words in the input text are then removed using this list. The choice of stop words, on the other hand, can vary depending on the context and the task at hand. Certain stop words, for example, may be relevant in one domain but not in another. As a result, it is critical to tailor the stop word list to the specific application of stop word analysis.

2.4.6 Dependency parsing

Dependency parsing step in NLP is to analyze the syntactic structure of a sentence by determining the grammatical relationships between words. It aims to discover the dependencies or relationships that exist between words, which are typically represented as directed edges connecting the words in a sentence. These relationships reveal the sentence's hierarchical structure as well as the underlying grammatical rules.

The dependency parsing process begins with tokenizing and POS tagging the input sentence, which assigns each word a unique POS tag based on its grammatical function. Once the words have been tagged, the parser analyzes their relationships and builds a dependency tree that represents the sentence's grammatical structure. The root of the tree usually corresponds to the sentence's main verb or predicate.

Except for the root, each word in the sentence is connected to another by a directed edge, indicating their dependency relationship. The edge bears a label that describes the dependent word's grammatical role in relation to the headword. The label could, for example, indicate whether a word is a subject, object, adjective modifier, or adverbial modifier.

Dependency parsing has a wide range of applications in NLP tasks such as machine translation, sentiment analysis, information extraction, question answering, and text summarization. It provides valuable insights into sentence syntactic structure, allowing for more accurate and robust language understanding.

2.4.7 Part-of-speech (POS) tagging

Part-of-speech (POS) tagging is an important step in natural language processing (NLP) in which grammatical labels or tags are assigned to individual words in a sentence. POS tagging aims to identify each word's syntactic category or part of speech, such as noun, verb, adjective, adverb, pronoun, preposition, conjunction, or interjection. This process is critical in many NLP applications, such as machine translation, information retrieval, sentiment analysis, and text-to-speech synthesis.

For example, words of given two sentences with their corresponding POS tags are listed below.

- "Book the flight." is represented as: [("Book", "Verb"), ("the", "Det"), ("flight", "Noun")]
- "I like to travel" is represented as: [("I", "Preposition"), ("like", "Verb"),

(“to”, “To”), (“travel”, “Verb”)]

By assigning appropriate tags to words based on their contextual usage, POS tagging allows computers to understand the structure and meaning of sentences. The ambiguity caused by homonyms (words with multiple meanings) and the context-dependent nature of word usage make accurate POS tagging difficult. For example, "bank" can be either a noun (financial institution) or a verb (to tilt or incline). Contextual cues and statistical models are frequently used to resolve such cases. These models are trained on large annotated corpora that have been manually labeled with the corresponding parts of speech by human linguists.

2.5 BERT Model

The BERT (Bidirectional Encoder Representations from Transformers) model is a deep learning language model developed by researchers at the Google AI Language unit in 2018 and shared as open source. The BERT model has shown remarkably successful performance in various NLP tasks such as Sentiment Analysis, Question Answering, Text Prediction, Text Generation and Summarisation.

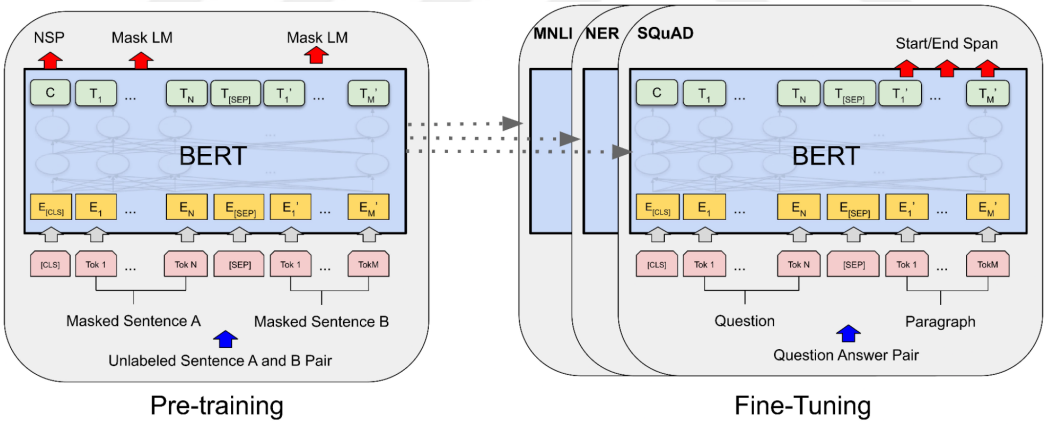


Figure 2.6 BERT Model Infographic
[19]

In addition to the impact of the Transformer approach in the field of NLP, the large amount of training data feeding the BERT neural network architecture and its training as a Masked Language Model (MLM) differentiates BERT from other NLP models. A large dataset of 3.3 billion words from Wikipedia (2.5 billion words) and Google’s BooksCorpus (800 million words) was prepared for training BERT. With the help of this large training dataset, BERT is aimed to have a deep understanding of the English language and to represent everyday spoken language. BERT’s

Masked Language Model is a critical component that enables bidirectional learning of the text. MLM works by masking a word in a sentence and forcing BERT to use the surrounding words to bidirectionally predict the missing word. This approach is a breakthrough in NLP and enables BERT to gain a deeper understanding of language. For example, in the sentence "This weekend I promised my little brother to eat [mask] at Eminönü Coast in Istanbul." we can guess what might come in the blank part because Eminönü is historically and culturally a "fish" eating area.

With the same understanding, the BERT model is able to predict the missing word "fish" by using contextual clues and prior knowledge used during training. This approach, thanks to MLM, differentiates the BERT model from other NLP models created to fill the gaps and understand the language more deeply. The ability to capture the contextual meaning of words and expressions required for sentiment analysis, with the particular advantage of being trained on large amounts of text data, results in very successful results. Therefore, BERT-based language models have been shown to outperform traditional machine learning-based approaches in sentiment analysis tasks, especially on text-based cases.

2.5.1 BERT Model performance measurements

BERT models have achieved remarkable success over 11 popular performance measurement methods widely known in the field of natural language processing (NLP). BERT models have outperformed previous popular NLP models and have even achieved better accuracy rates than human performance on some tasks. Let's try to explain how a few pre-selected performance measurement methods, listed below, measure the success of the BERT model.

- SQuAD

  "Stanford Question Answering Dataset" [20] is a reading comprehension dataset containing approximately 108,000 question sentences that can be answered with the help of paragraphs gathered from open-source Wikipedia content. This dataset was used to measure the performance of BERT. BERT's performance on the SQuAD evaluation method is a remarkable achievement as it exceeds previous best technology models. The table given below shows the performance of the BERT model on SQuAD test.

BERT's Performance - SQuAD1.1 Leaderboard

Rank	Model	EM	F1
1 **[Oct 05, 2018]**	**BERT (ensemble)** Google AI Language arrive.org/abs/180.04805	**87.433**	**93.16**
-	**Human Performance** Stanford University (Rajpurkar et al. '16)	82.304	91.221
2 **[Sep 09, 2018]**	**nlnet (ensemble)** Microsoft Research Asia	85.356	91.202
3 **[Jul 11, 2018]**	**QANet (ensemble)** Google Brain & SMU	84.454	90.490



Figure 2.7 BERT Performance on SQUAD benchmark

- SWAG "Situations With Adversarial Generations" [21] evaluation method is used to test the ability of an NLP model to make common sense inference on a large-scale data set. This dataset consists of 113,000 multiple-choice questions generated from video & film scenes based on common sense situations. SWAG is another method used to evaluate the performance of BERT, and BERT performed well in this evaluation method, surpassing the previous best models and human-level performance. The performance of BERT and other models can be observed in the table below.

BERT's Performance - SWAG (Situations With Adversarial Generations)

System	Dev	Test
BERT$_{LARGE}$	**86.6**	**86.3**
Human (expert)	-	85.0
OpenAI GPT	-	78
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2



Figure 2.8 BERT Performance on SWAG benchmark

- GLUE "General Language Understanding Evaluation" [22] benchmark is a

collection of resources for training, measuring and benchmarking language models. GLUE consists of nine challenging tasks, which are specifically designed to test the understanding of an NLP model. While some of these tasks may seem straightforward, they are highly useful in demonstrating which models are best suited for particular NLP applications. The table given below shows the performance of the BERT model on the GLUE benchmark.

BERT's Performance on GLUE:

Task	Average	Grammatical	Sentiment Analysis	Similarity	Paraphrase	Question Similarity	Contradiction	Answerable	Entail
BERT$_{LARGE}$	**82.1**	**60.5**	**94.9**	**86.5**	**89.3**	**72.1**	**86.7/85.9**	**92.7**	**70.1**
BERT$_{BASE}$	79.6	52.1	93.5	85.8	88.9	71.2	84.6/83.4	90.5	66.4
OpenAI GPT	75.1	45.4	91.3	80.0	82.3	70.3	82.1/81.4	87.4	56.0
Pre-OpenAI SOTA	74.0	35.0	93.2	81.0	86.0	66.1	80.6/80.1	82.3	61.7
BiLSTM+ELMo+Attn	71.0	36.0	90.4	73.3	84.9	64.8	76.4/76.1	79.8	56.8

Figure 2.9 BERT Performance on GLUE benchmark

2.6 Turkish BERT Model - BERTurk

BERTurk is a collection of community-driven models designed specifically for Turkish language on multiple NLP tasks. Some of the developed models are BERT, DistilBERT, ELECTRA, and ConvBERT, which are all intended to improve natural language processing tasks in Turkish. The name "BERTurk" was chosen to honor the vibrant Turkish NLP community, which provided valuable datasets for both pretraining and evaluation.

An extensive corpus, including a filtered and segmented version of the Turkish OSCAR corpus, a recent dump of the Wikipedia dataset, a variety of OPUS corpora, and a specifically curated corpus by Kemal Oflazer, was used to train the current iteration of the models used in these models. This sizeable training corpus is 35GB in size and contains an astounding 4,404,976,662 tokens in total.

Here is a list of trained both cased and uncased models from BERTurk model collection as detailed on Table 2.1.

Table 2.1 Table of BERTurk models

Model name	Pre-training corpus size
ELECTRA Small (cased)	35GB
ELECTRA Base (cased)	35GB
ELECTRA Base mC4 (cased)	242GB
ELECTRA Base mC4 (uncased)	242GB
BERTurk (cased, 32k)	35GB
BERTurk (uncased, 32k)	35GB
BERTurk (cased, 128k)	35GB
BERTurk (uncased, 128k)	35GB
DistilBERTurk (cased)	35GB
ConvBERTurk (cased)	35GB
ConvBERTurk mC4 (cased)	242GB
ConvBERTurk mC4 (uncased)	242GB

The training of both cased and uncased models on a TPU v3-8 has been facilitated by the use of Google's TensorFlow Research Cloud (TFRC). Additionally, it introduced as cased and uncased models with a 128k vocabulary size, providing greater flexibility and a wider range of options to meet different requirements for natural language processing.

3
APPLICATION

3.1 Data Set

The dataset used in this study is THY_SA_Dataset, an open-source dataset available on the Turkish Language Depository (TDD) website [23]. The TDD portal is a platform for sharing data resources and software such as datasets, language models, corpora and natural language processing tools for Turkish Natural Language Processing and computational linguistics.

Content	Date	Language	Sentiment	Tags
hiçbir müzik platformunda olmayan placebo mtv unplugged türk hava yolları arşivinde var.	03/31/2021	tr	positive	mim, Reporting, TR, Ucak ici Eglence Sistemi (IFE), Yorum
@TK_HelpDesk Tamamdir super, dediginiz gibi yapinca basariyla oldu. Tesekkur ederim.	07/09/2021	tr	positive	mim, TR, Webform, Yorum
Yeni Delhi ye uçuşlarınız varmı	03/03/2021	tr	neutral	Destinasyon, Kontrol Edildi, mim, Mim Post Yorum, TR
İki valizimi kırdılar şijayet ettim tık yok.	01/15/2021	tr	negative	Hasarli Bagaj, mim, Mim Post Yorum, TR

Figure 3.1 Dataset preview on TDD.ai website

The variables in the data set used are as follows.

- "Content" - tweets quoting @TK_HelpDesk account (in character format)
- "Date" - The date the Tweet was posted (in DD/AA/YYYYY date format)
- "Language" - the language of the tweet in ’tr’ Turkish and ’en’ English (in character format)
- "Sentiment" - predefined sentiment label of the tweet (in character format as "positive", "negative", "neutral")
- "Tags" - The tags are based on the information presented in the tweet.

Since the "tweet" variable, which will be taken as the basis of the study, has the form of a sentence structure, it is necessary to examine the characters it contains. Before including the "tweet" variable in a machine learning model, it needs to be transformed into a meaningful format by going through data cleaning stages. Through these processes, "tweets" will become more useful and understandable by the machine learning model to be built. Below are the steps for cleaning "tweets" in order.

Right before the modeling of the data, the semantic integrity of the dataset should be verified. One of the most important steps in machine learning modeling is to transform the data set to be processed into quality attributes with the help of algorithms, so that the outputs of the model are expected to be robust. This consistency is expected to reduce the error values and increase the success metrics of the models and approaches. Our models were built in R Studio environment with the help of Tensorflow and Keras libraries using R programming language.

3.2 Data Preprocessing

During the data preprocessing step, a number of operations must be performed. A text-based response variable designated as "tweet" must be cleaned for such a provided text-based modeling task. The following is a list of activities that were performed on our data set.

1. All characters in a tweet must be lowercase. This is done to standardize the text and eliminate variations in capitalization.
2. Remove special characters and punctuation: This includes all characters that are not letters or numbers, such as hashtags, mentions and emoticons.
3. Removal of Unicode characters: Any character that is not a letter in the Turkish or English alphabet but has the status of a unicode character should be removed.
4. Removing URLs and email addresses: These are often irrelevant to sentiment analysis work and are recommended to be removed.
5. Remove stop words: These are common conjunctions such as "and", "this", "can", "how" in English and "bu", "şu", "bir", "hiç", "birkaç" in Turkish, which do not affect the sentiment of the sentence and do not convey meaning.
6. Stemming or lemmatization: This is the process of reducing words to reduce variation and increase the success of the model.

7. Cleaning of digits: This process can be applied depending on the scope of the study and the nature of the numbers.

8. Removing double or more spaces: This should be done to remove unnecessary spaces between words.

In addition to these processes, emojis, which are symbols indicating emotional states and are frequently used on social media channels today, should also be taken into consideration. Although emojis constitute a unity of meaning within themselves, they are not included in the objectives of this study, so it was decided to clean them up. After this cleaning and simplification process, it is expected to increase the performance of the machine learning models. Similarly, it was also considered appropriate to delete all digits in the tweets and they were cleaned. The main purpose here is that although there is a time difference between, for example, a passenger's complaint that his/her flight was 20 minutes late and 45 minutes late, it basically tells us that a service was disrupted and the model we want to build investigates the quality of word wholes rather than the quantity of numbers. It is possible to gather that, tweets that are likely to be shared in the case of both events in the previous example can be labeled as "negative" regardless of time length.

As a result of this dataset cleaning, the total number of characters in each tweet is reduced. Based on the number of characters in the remaining tweets, tweets containing few meaningful words were discarded from the dataset. At the end of these data cleaning operations, 20689 rows of variables remained from the initial 51695 rows of data.

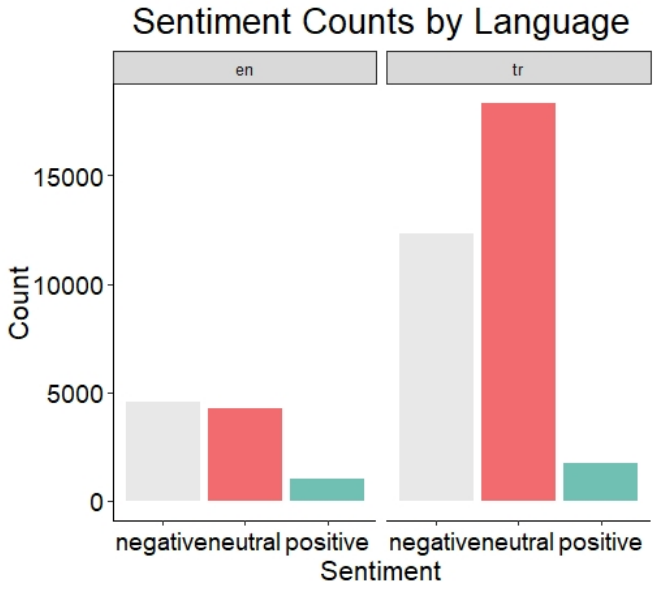


Figure 3.2 Sentiment labels grouped by languages

After the data cleaning process, the first step of the study is to separate Turkish tweets from English tweets and to set up separate machine learning models. Due to the grammatical structure and the special features of Turkish as a suffixed language, such as suffix/stem distinctions, two separate models that can make sense of the two languages should be built. The "language" variable in the dataset has already been used to categorize the tweet as a language. When only tweets in "en" and "en" languages were filtered using this variable, it was observed that some of the remaining tweets were classified as "az" (Azerbaijani) or "uz" (Uzbek) while their content was Turkish. These languages, which belong to the same language family as Turkish, were mostly miscategorized by Twitter's algorithm. Instead of discarding these misclassified tweets from the database and causing data loss, we used Google's Compact Language Detector [24], a model developed by Google that outputs which language a text belong to. Thanks to this process, the language of a tweet was tried to be understood in two steps, and fewer data was removed from the study because it was not of the same quality. As a result of these steps, a distribution as shown in Figure 3.2 was obtained.

3.3 Data Modeling

In this study, the estimator sequential models were trained separately in two different languages, Turkish (TR) and English (EN).

The training procedure starts after the model architecture for both languages has been determined. For both sentiment analysis models, the dataset is partitioned into 80% training and 20% testing. The model architectures of the trained sequential models are identical and are constructed as Figure 3.3. When given the labeled data, the model learns to map the input text to the corresponding sentiment labels through an iterative optimization process. The parameters of the model are changed to reduce the discrepancy between the predicted sentiments and the given labels in techniques like backpropagation and gradient descent.

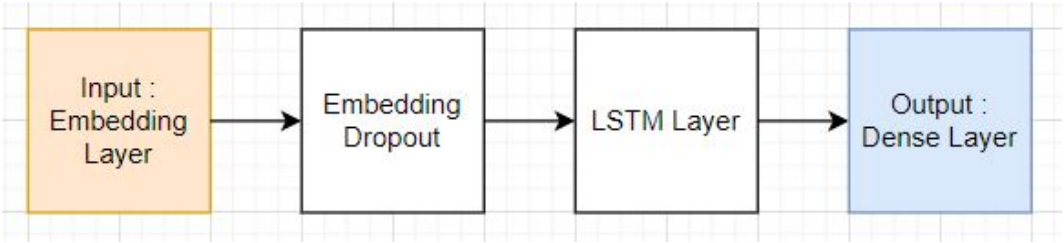


Figure 3.3 Flow Chart of Sequential Model Architecture

To improve the generalization and avoid overfitting during training, it is common

practice to use methods like mini-batch training, regularization, and dropout. In order to make sure the model is learning properly and not overfitting the training data, its performance also gets evaluated using test data.

When working with large datasets, training a sequential sentiment analysis model frequently takes a significant amount of computational time. High-performance hardware, such as GPUs or TPUs, is commonly used to accelerate the training process and handle the computational demands of deep learning algorithms. In this work, regular personal computer hardware was enough to train both models with nearly 1 million parameters, respectively.

Once the model training is complete, the final model is evaluated using the testing set. This evaluation measures the sentiment analysis model's accuracy and precision metrics at hand. The results help assess each model's performance and provide insights into potential areas for improvement, if needed.

3.4 Turkish Sequential Model

As the input variable of the Turkish sequential model, the tweets in the dataset were converted into vector values by the tokenization method. A maximum parameter of 49 words, which is the number of elements of the longest sentence vector within the vectors, was set. In this way, the words in each vectorized tweet can be fully represented by the model. In order to increase the prediction success of the model, the epoch training value of the model was set to 20 and the batch_size value was set to 512. In the output layer of the model, there are tags belonging to 3 different emotional states, and the vector equivalents of these tags are created during the data preparation phase. As a result of the tokenization process, the input and output values found as text were converted into numeric vectors that the model can understand and the consistency of the model was ensured. The number of input parameters of the Turkish sequential model is 1,061,022 as shown in the 3.1 image below. With such a high number of trained specific parameters, the model was aimed to be successful.

Table 3.1 Turkish Sequential Model Architecture

Model : "Turkish sequential"		
Layer (type)	Output Shape	Param #
input (Embedding)	(None, 49, 32)	1044800
embedding_dropout (Dropout)	(None, 49, 32)	0
lstm (LSTM)	(None, 49)	16072
output (Dense)	(None, 3)	150
Total params: 1,061,022		
Trainable params: 1,061,022		
Non-trainable params: 0		

After this modeling process, Turkish sequential sentiment analysis model was trained for nearly 8-10 minutes.

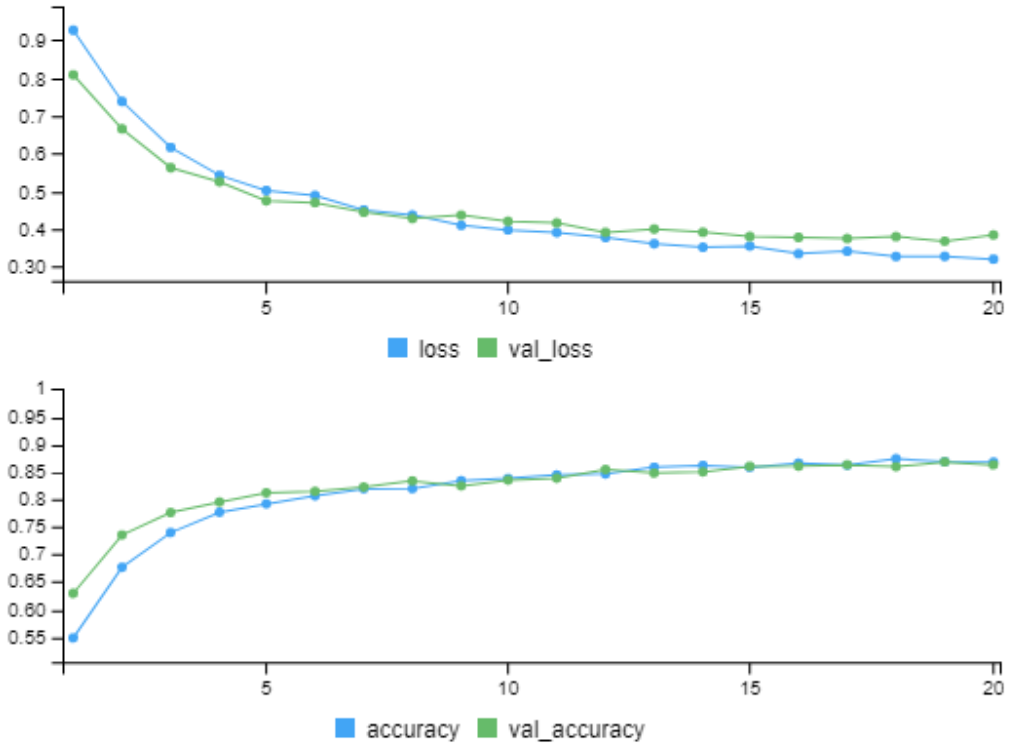


Figure 3.4 Turkish Sequential Model Training Results

As a result of the predictions made with the test set, which was kept separate from the trained data during the model construction phase, the following success values were obtained with the Sequential Turkish model.

Table 3.2 Turkish Sequential Model Confusion Matrix Table

Confusion matrix			
	Reference		
Prediction	neutral	negative	positive
neutral	151	28	6
negative	13	570	4
positive	1	4	163
Statistics by class			
	Class: Neutral	Class: negative	Class:Positive
Sensitivity	0.9152	0.9468	0.9422
Specificity	0.9561	0.9497	0.9935
Pos Pred Value	0.8162	0.9710	0.9702
Neg Pred Value	0.9815	0.9093	0.9870
Prevalence	0.1755	0.6404	0.1840
Detection Rate	0.1606	0.6064	0.1734
Detection Prevalence	0.1968	0.6245	0.1787

The overall prediction success of our Turkish sentiment analysis model, which includes 3 sentiment classes, was 94.04%. According to this model, which has very high Sensitivity (neutral: 0.91, negative: 0.94, positive: 0.94) and Pos Pred Value (neutral: 0.81, negative: 0.97, positive: 0.97) values, the detection of "neutral" tweets will occur less frequently than other categories. Based on this, there may be a weakness in obtaining the "neutral" classification when predicting a new tweet. In addition, by looking at the Detection Rate (neutral: 0.16, negative: 0.6, positive: 0.17) obtained, it can be seen that the weight of the "negative" classification in the model is similar to the distribution of the classes in the dataset that our model is fed.

3.5 English Sequential Model

The sequential model in English is designed similarly to the Turkish model. In this model, there are only differences in terms of parameters. For example, the longest sentence in English tweets is 64 words long. In addition, there are 295,907 specific vectorized words. The reason why this number is quite low compared to the Turkish model can be explained by the small number of English tweets in the dataset. The English sequential model was also trained with an epoch of 20 and a batch_size of 512. The detailed architectural visualization of the model 3.3 is shown below.

Table 3.3 English Sequential Model Architecture

Model : "English sequential"		
Layer (type)	Output Shape	Param #
input (Embedding)	(None, 64, 32)	270880
embedding_dropout (Dropout)	(None, 64, 32)	0
lstm (LSTM)	(None, 64)	24832
output (Dense)	(None, 3)	195
Total params: 295,907		
Trainable params: 295,907		
Non-trainable params: 0		

During the sequential model training in English, epoch-wise loss and accuracy values were calculated and plotted as Figure 3.5 below.

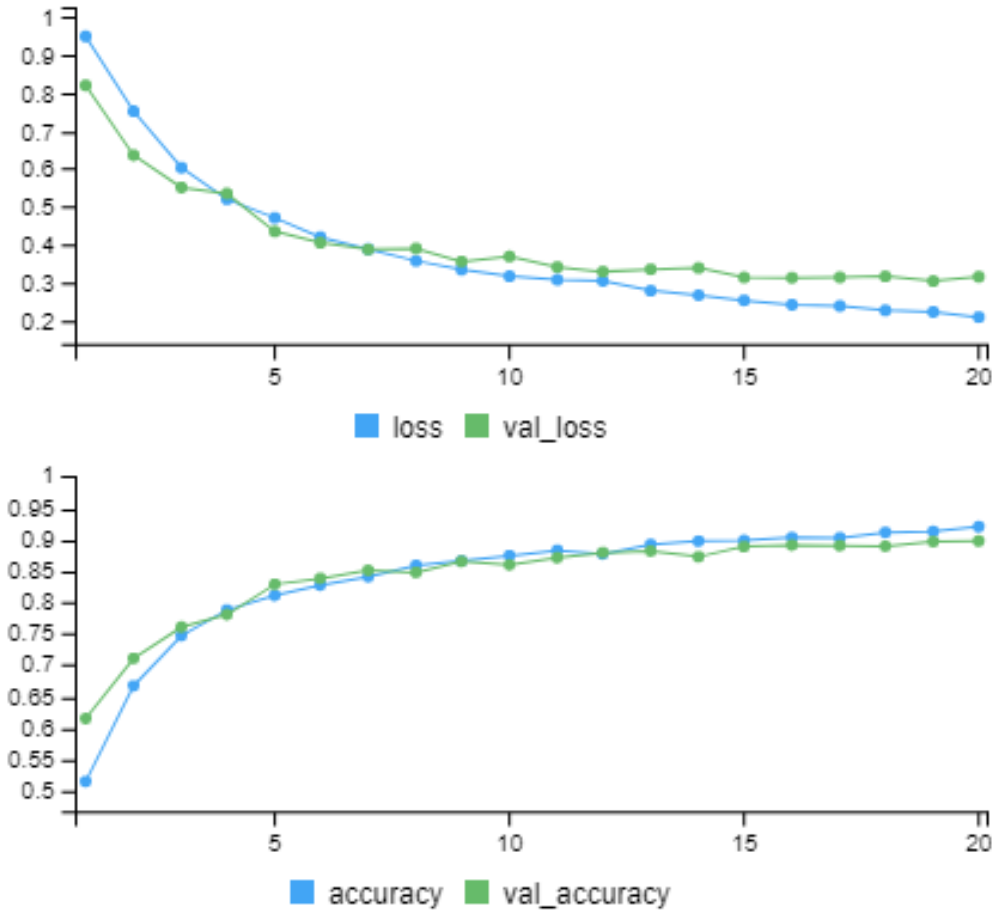


Figure 3.5 English Sequential Model Training Results

As a result of the predictions made on the test set with the model built on the tweets parsed in English, the following success values were obtained with the Sequential English model.

Table 3.4 English Sequential Model Confusion Matrix Table

Confusion matrix			
	Reference		
Prediction	neutral	negative	positive
neutral	201	20	2
negative	11	303	3
positive	4	0	90
Statistics by class			
	Class: Neutral	Class: negative	Class:Positive
Sensitivity	0.9306	0.9381	0.9474
Specificity	0.9474	0.9550	0.9926
Pos Pred Value	0.9013	0.9558	0.9574
Neg Pred Value	0.9635	0.9369	0.9907
Prevalence	0.3407	0.5095	0.1498
Detection Rate	0.3170	0.4779	0.1420
Detection Prevalence	0.3517	0.5000	0.1483

The overall prediction success of our English sentiment analysis model was 93.69%. Sensitivity (neutral: 0.930, negative: 0.938, positive: 0.947) values of our model were very close to each other and similar sensitivity values were calculated on the prediction classes. Based on the Table 3.4 values such as Specificity (neutral: 0.947, negative: 0.955, positive: 0.992) we can observe that the tokenizer vector in the "positive" classes is more inclusive than the other two classes. As in the Turkish model, similar to the distribution of the classes in the dataset fed to the model, the weight of the "negative" emotion classification was also found in the Detection Rate (neutral: 0.317, negative: 0.477, positive: 0.142) in the English sentiment analysis model. However, unlike the Turkish model, the weight of the "negative" class was found to be less. After the Sequential Turkish and Sequential English sentiment analysis models, the performance of the two BERT models, which were previously trained in this area, were measured on similar data sets.

3.6 BERT and BERTurk models

The BERT model trained in English was more inclusive in the testing phase of the model as it included all 3 emotion classes in the dataset in the study. Since "neutral" was not included in the emotion classes during the training of the BERTurk model, it could not fully cover the tested dataset.

The results obtained with the BERT English model are as shown in Table 3.5.

Table 3.5 BERT testing results table

	predicted			
truth	NEGATIVE	NEUTRAL	POSITIVE	TOTAL
NEGATIVE	2680	377	177	3234
NEUTRAL	856	1004	301	2161
POSITIVE	176	145	637	958
TOTAL	3712	1526	1115	

As can be seen from this table, predicting a "negative" tweet using the predefined BERT library was 83% successful in accuracy. One of the reasons for this is the number of "negative" classes in the distribution of classes in the test dataset used is high and more homogeneous. For the "positive" classes, which is a similar example, the success rate was relatively low at 66%. The "neutral" class, which is evaluated separately from these two classes, was again estimated as "neutral" with a very low success rate of 46%. As expected, predicting tweets marked as "neutral" as "negative" is at least as successful as predicting its own class. Predictably, the BERT model was not as successful in this case, as it has its own tokenizer, model architecture and methods for making sense of sentences. This is mainly due to the diversity of the dataset on which the BERT model was trained during its construction. When the models work on new data that it has not seen before, it is possible to get results like this when there are tokens that it cannot identify.

Similarly, the results of the Turkish tweets filtered from the main data and tested with the BERTurk model were generally similar. The only difference here is that the BERTurk model does not have the "neutral" class, so the success percentages and the distributions of these percentages differ. Table 3.6 shows the results of the BERTurk study.

Table 3.6 BERTurk testing results table

	predicted		
truth	NEGATIVE	POSITIVE	TOTAL
NEGATIVE	5469	555	6024
NEUTRAL	1030	625	1655
POSITIVE	486	1244	1730
TOTAL	6985	2424	

According to the results in BERTurk Table, predicting the "negative" classes as "negative" has the most successful percentage. This high prediction rate may be misleading due to the high number of "negative" variables in the test data. However, in a similar way, it is possible to evaluate this success rate as the performance of the BERTurk model. Unlike the English BERT model, in the test of this model, when the "neutral" classes were predicted, it was observed where the resulting class was clustered. As can be seen from the table, 62% of the predictions were "negative". Again, this can be interpreted as the class distribution of the data set used for the test. Finally, by looking at the number of tweets classified as "positive", it can be said that it is generally successful, although not as successful as in the "negative" example.

4
CONCLUSION

4.1 Reliability of the data and generalizability of the study

As the dataset used throughout the study is gathered from an internet source, it's likely to contain some labeling errors by preparation approach. There are many factors such as the way the labeling process is done, the perspective of the person doing it, and the quality of the data labeling tools, if used, that may reduce the context reliability of the data set. At this point, as can be seen in Table 4.1 below, there are tweets that would normally be labeled differently, but were correctly predicted according to the Turkish Sequential model result. A similar example also exists in the English dataset used in the model.

Table 4.1 BERTurk Table

cleanedtweet	**label**	**predicted_label**	**model_confidence_score**
büyük resimden küçüğe doğru	positive	neutral	0.90
thynin altıncı a tipi uçağı istanbul via	positive	neutral	0.97
yapilamiyor sikinti orada anlatamiyor muyum	negative	neutral	0.95
normalde dolar açıklıyor kampanyalarını artık kullanılabilir fiyatlar	negative	neutral	0.97
belarus planlama yapacakmısınız sayın başkan iptal uçuşlar haricinde	negative	neutral	0.76

As can be also seen in Table 4.2 for English tweets, the tweet that was placed in the first row is previously labeled as "negative" with being considered as a refund process. However, testing with the English sentiment model predicted the sentiment of this text as "neutral". It can be easily said that model failed in its prediction based on a labeling error that has previously done at the data frame labeling step.

Table 4.2 BERT English Table

cleanedtweet	label	predicted_label	model_confidence_score
i need the cost of my flight to be refunded	negative	neutral	0.90
there is no response to my support tickets please help tk	positive	neutral	0.82
anyone able to assist me recover this luggage i keep being tossed between turkish luggage days later	neutral	negative	0.99
i need to change my flight but i can t with your website need some help please	negative	neutral	0.75
i d like to accept the changes without calling in is that possible the schedule change is less than mins thanks	negative	neutral	0.98

According to these results, it is understandable that the accuracy of sentiment analysis models will increase with the implementation of meaningful labeling and strict data control in terms of reliability and generalizability. On the other hand, one of the biggest obstacles to the generalizability of BERT, BERTurk, and all other pre-trained machine learning models is whether or not the language models are trained with data on the intended topic, as seen in the testing phase. Moreover, the cased and uncased pre-trained model's knowledge on a specific topic may not cover all the related logical connections to generalize its efficiency.

In both sequential models trained and tested in this study, accuracy rates were high for such a sentiment analysis model with the help of the creation of individual tokenizers on a specific case. While the accuracy rates of pre-trained BERT models are not as high as sequential models, it can be inferred as their case/uncase states. In addition, in terms of model architectures, parameters such as the number of layers in the BERT architecture, the number of trained steps, batch size and other hyperparameter values can also directly affect these accuracy rates.

REFERENCES

[1] E. Constantinides, N. Holleschovsky, "Impact of online product reviews on purchasing decisions," English, in *Proceedings of the 12th International Conference on Web Information Systems and Technologies*, T. Majchrzak, P. Traverso, V. Monfort, K.-H. Krempels, Eds., 12th International Conference on Web Information Systems and Technologies, WEBIST 2016, WEBIST ; Conference date: 23-04-2016 Through 25-04-2016, SCITEPRESS, Apr. 2016, pp. 271–278, isbn: 978-989-758-186-1. doi: 10.5220/0005861002710278.

[2] R. Bose, R. Dey, S. Roy, D. Sarddar, "Sentiment analysis on online product reviews," *Information and Communication Technology for Sustainable Development*, 2019.

[3] R. Suchdev, P. Kotkar, R. M. Ravindran, S. Swamy, "Twitter sentiment analysis using machine learning and knowledge-based approach," *International Journal of Computer Applications*, vol. 103, pp. 36–40, 2014.

[4] F. Sternberg, K. H. Pedersen, N. K. Ryelund, R. R. Mukkamala, R. Vatrapu, "Analysing customer engagement of turkish airlines using big social data," *2018 IEEE International Congress on Big Data (BigData Congress)*, pp. 74–81, 2018.

[5] E. Adalı, "Doğal dil İşleme," *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, vol. 5, no. 2, 2012, issn: 1305-8991.

[6] E. M. Mercha, H. Benbrahim, "Machine learning and deep learning for sentiment analysis across languages: A survey," *Neurocomputing*, vol. 531, pp. 195–216, 2023, issn: 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.02.015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231223001546.

[7] B. Pang, L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008, issn: 1554-0669. doi: 10.1561/1500000011. [Online]. Available: http://dx.doi.org/10.1561/1500000011.

[8] A. PATEL, P. OZA, S. AGRAWAL, "Sentiment analysis of customer feedback and reviews for airline services using language representation model," *Procedia Computer Science*, vol. 218, pp. 2459–2467, 2023, International Conference on Machine Learning and Data Engineering, issn: 1877-0509. doi: https://doi.org/10.1016/j.procs.2023.01.221. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050923002211.

[9] S. Zhai, Z. Zhang, "Semisupervised autoencoder for sentiment analysis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[10] A. Severyn, A. Moschitti, "Unitn: Training deep convolutional neural network for twitter sentiment classification," in *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 2015, pp. 464–469.

[11] A. Onan, M. A. Toçoğlu, "Satire identification in turkish news articles based on ensemble of classifiers," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 28, no. 2, pp. 1086–1106, 2020.

[12] J. Devlin, M. Chang, K. Lee, K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. arXiv: 1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805.

[13] S. Schweter, "Berturk - bert models for turkish," version 1.0.0, Apr. 2020. doi: 10.5281/zenodo.3770924. [Online]. Available: https://doi.org/10.5281/zenodo.3770924.

[14] *Yinelemeli Sinir Ağları (RNN) nedir? — elektrikport.com*, [Accessed 31-May-2023]. [Online]. Available: https://www.elektrikport.com/makale-detay/yinelemeli-sinir-aglari-(rnn)-nedir/23277#ad-image-0.

[15] A. Karpathy, *The Unreasonable Effectiveness of Recurrent Neural Networks — karpathy.github.io*, [Accessed 31-May-2023]. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[16] A. Vaswani *et al.*, *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL].

[17] Wikipedia, *Türk Hava Yolları — Wikipedia, the free encyclopedia*, http://tr.wikipedia.org/w/index.php?title=T%C3%BCrk%20Hava%20Yollar%C4%B1&oldid=29651962, [Online; accessed 31-May-2023], 2023.

[18] *OpenAI Tokenizer — platform.openai.com*, https://platform.openai.com/tokenizer, [Accessed 31-May-2023].

[19] *BERT 101 - State Of The Art NLP Model Explained — huggingface.co*, https://huggingface.co/blog/bert-101, [Accessed 31-May-2023].

[20] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, "Squad: 100, 000+ questions for machine comprehension of text," *CoRR*, vol. abs/1606.05250, 2016. arXiv: 1606.05250. [Online]. Available: http://arxiv.org/abs/1606.05250.

[21] R. Zellers, Y. Bisk, R. Schwartz, Y. Choi, "SWAG: A large-scale adversarial dataset for grounded commonsense inference," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 93–104. doi: 10.18653/v1/D18-1009. [Online]. Available: https://aclanthology.org/D18-1009.

[22] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. R. Bowman, *Glue: A multi-task benchmark and analysis platform for natural language understanding*, 2019. arXiv: 1804.07461 [cs.CL].

[23] *Türkçe Dil Deposu (TDD)*, https://tdd.ai/, 2022.

[24] J. Ooms, *Cld3: Google's compact language detector 3*, R package version 1.5.0, 2023. [Online]. Available: https://docs.ropensci.org/cld3/%20https://github.com/ropensci/cld3%20https://ropensci.r-universe.dev/cld3.

PUBLICATIONS FROM THE THESIS

Conference Papers

1. Kurtuluş T. ve Cene E., “Analysis of Customer Feedback Replied to The Turkish Airlines Support Account via Twitter with Natural Language Processing Methods”, 5th International Conference on Data Science and Applications (ICONDATA’22), September 7-11, 2022, Fethiye, TURKEY - ISBN: 978-605-72045-2-3