FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

UNIFIED FRAMEWORK FOR SENTIMENT ANALYSIS IN MULTIPLE LANGUAGES

YÜKSEK LİSANS TEZİ

ABDELRAHMAN TAHA ABDELTAWAB ABDELLATIF

İSTANBUL, 2023

FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

UNIFIED FRAMEWORK FOR SENTIMENT ANALYSIS IN MULTIPLE LANGUAGES

YÜKSEK LİSANS TEZİ

ABDELRAHMAN TAHA ABDELTAWAB ABDELLATIF
(210221101)

Danışman
(Dr. Öğr. Üyesi Shaaban A.I. Sahmoud)

İSTANBUL, 2023

FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
TEZ ONAY FORMU

08/08/2023

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği tezli yüksek lisans programı öğrencisi 210221101 numaralı ***Abdelrahman Taha ABDELTAWAB ABDELLATIF'ın,*** hazırladığı *"Twitter Verileri için Duygu Analizi"* konulu tezi ile ilgili Tez Savunma Sınavı, 08.08.2023 Salı günü saat 11:30'da yapılmış, sorulara alınan cevaplar sonunda adayın tezinin **Kabulüne ~~Oy Çokluğu~~/Oy Birliği** ile karar verilmiştir.

Tez adı değişikliği yapılması halinde: Tez adının Unified Framework for Sentiment Analysis in Multiple Languages şeklinde değiştirilmesi uygundur.

	Jüri Üyesi	**Karar**
1.	Dr. Öğr. Üyesi Şaban SAHMOUD (Danışman)	Kabul
2.	Doç. Dr. Ferzat ANKA	Kabul
3.	Dr. Öğr. Üyesi Wisam ELMASRY	Kabul
4.	.....	.....
5.	.....	.....
6.	(İkinci Danışman)*.....	.....

**2. Danışman varsa doldurulması gerekmektedir.*

ETHICAL DECLARATION:

I declare that the rules of scientific ethics were followed in the writing of this thesis, that appropriate references were made in accordance with scientific norms when benefiting from the works of others, that no distortion was made in the used data, and that no part of this thesis was presented as a study in the university I am affiliated with or at any other university.

Abdelrahman Taha Abdeltawab Abdellatif

ACKNOWLEDGEMENT

My profound appreciation goes to my mentors, Assist. Prof. Dr. Şaban SAHMOUD and Assist. Prof. Dr. Ali NİZAM. Their invaluable insights, guidance, and unwavering dedication have been instrumental in shaping this thesis. Their expertise and mentorship not only facilitated this academic work but also instilled in me the rigor and passion for continued learning and exploration.

Additionally, my heartfelt thanks to my parents for their unwavering encouragement and support throughout my academic journey. Their faith in me, combined with their constant love and guidance, has been the backbone of all my endeavors. Their sacrifices, belief, and unwavering love have been the bedrock upon which all my aspirations have been built.

To all of them, I remain forever grateful.

Abdelrahman Taha Abdeltawab Abdellatif

ÇOKLU DİLDE DUYGU ANALIZI İÇİN BÜTÜNLEŞİK BİR YAZILIM ALTYAPISI

Abdelrahman Taha Abdeltawab Abdellatif

ÖZET

Duygu analizi, müşteri görüşlerini, duygularını ve geri bildirimlerini anlama açısından hayati önem taşır. Bu çalışma, çok dilli duygu analiz performansını artırmak için bütünleşik bir sistem yaklaşımı sunmaktadır. Çalışmamızda, İngilizce, Türkçe, Arapça ve Fransızca dillerini kapsayan duygu analizinde Google Çeviri ve Yandex Çeviri olmak üzere iki popüler makine çeviri hizmeti kullanılmıştır. Bulgu ve sonuçlar, çok dilli duygu analizi için birleşik bir çerçevenin kullanılmasının önemini, farklı dillerde duygu analizini kolaylaştırmada makine çeviri hizmetlerinin önemini vurgulamaktadır. Ayrıca sonuçlar, duygu analizi alanındaki araştırmacılar ve uygulamacılar için yararlı bilgiler sağlamaktadır. Geliştirdiğimiz sistem birçok veri seti üzerinde değerlendirilmiş ve diline bağlı olarak doğrulukta %1 ila %22 arasında iyileşme gösteren umut verici sonuçlar ortaya koymuştur. Yaklaşımımız, dil özgü modelleri geride bırakarak önerilen çeviri tabanlı çok dilli çerçevenin etkinliğini göstermiştir. Ek olarak, duygu analizinin performansının farklı diller arasında değiştiğini, Google Çeviri'nin Türkçe ve Arapça çevirilerin duygu analizinde daha iyi performans gösterirken, Yandex Çeviri'nin İngilizce ve Fransızca çevirilerin duygu analizinde daha iyi sonuçlar gösterdiğini tespit edildi.

Anahtar kelimeler: Duygu analizi, çok dilli duygu analizi, derin öğrenme, çeviri tabanlı duygu analizi, LSTM.

UNIFIED FRAMEWORK FOR SENTIMENT ANALYSIS IN MULTIPLE LANGUAGES

Abdelrahman Taha Abdeltawab Abdellatif

ABSTRACT

Multilingual sentiment analysis plays a critical role in comprehending customer sentiment, feedback, and emotional responses. This study introduces a comprehensive framework designed to augment the efficacy of sentiment analysis across multiple languages. The research utilizes renowned machine translation services, namely Google Translate and Yandex Translate, to carry out sentiment analysis in several languages including English, Turkish, Arabic, and French. The outcomes underline the advantage of deploying a single, comprehensive framework for multilingual sentiment analysis. Furthermore, they underscore the crucial role machine translation services play in simplifying sentiment analysis across various languages. The insights gained from the results are beneficial to both researchers and practitioners in the sentiment analysis sphere. The proposed framework underwent testing on multiple datasets, exhibiting encouraging results with an improvement in accuracy between 1% and 22% depending on the language. Our method outperforms language-specific models and substantiates the efficiency of the proposed translation-based multilingual framework. Additionally, the study revealed that the efficacy of sentiment analysis fluctuates between different languages. Google Translate demonstrated superior performance in Turkish and Arabic sentiment analysis translations, whereas Yandex Translate excelled in English and French sentiment analysis translations.

Keyword: Sentiment analysis, multilingual sentiment analysis, deep learning, translation-based sentiment analysis, LSTM.

PREFACE

In this study, we use advanced tools like Google Translate and Yandex Translate to analyze sentiments in several languages, including English, Turkish, Arabic, and French. The goal is to see if one single approach can be effective across different languages. The aim is to break language barriers to understand feelings and emotions conveyed through words. Sentiment analysis is a method that is used to gain insight into people's emotions by studying their words, often used to understand customer feedback.

The following pages will take researchers through our detailed findings on how well different languages and translation tools performed in sentiment analysis. We hope this study is a helpful resource for those looking to explore sentiment analysis in multiple languages, offering insights and knowledge to both researchers and practitioners in the field.

Warm regards,

Abdelrhman Taha Abdeltawab Abdellatif September 2023

TABLE LIST

Pages

LIST OF FIGURES

Pages

CONTENTS

FIRST CHAPTER

1. INTRODUCTION

Sentiment detection, often known as opinion mining, falls under the domain of Natural Language Processing (NLP). Its main function is to identify and extract the emotions or attitudes present in each text [1]. Its relevance has been amplified in various domains such as marketing, politics, and customer service because of the rapid surge in social media and internet usage [2].

Sentiment analysis has seen the utilization of various machine learning and deep learning methodologies. Traditional machine learning techniques like Naive Bayes, Support Vector Machines, and Decision Trees are commonly employed. Nonetheless, these methods can encounter difficulties due to the complex characteristics of human language encompassing elements such as sarcasm, idiomatic expressions, and reliance on context [3][4].

Alternatively, advanced neural networks such as Long Short-Term Memory (LSTM) models are a subset of deep learning techniques and Word2Vec have demonstrated encouraging results in sentiment analysis. These models are proficient at grasping the semantic context and intricate meanings in text[4]. LSTM, a category of Recurrent Neural Network (RNN), is especially competent in dealing with sequential data and long-term dependencies in text [5]. Word2Vec, on the contrary, generates word embeddings that represent both semantic and syntactic similarities among words [6].

In today's world, powered by data, sentiment analysis has a crucial role. Understanding the emotional undertone of text can have wide-ranging applications, from deciphering customer feedback to gauging public sentiment towards societal matters. Numerous models utilizing machine learning and deep learning techniques

have been implemented to serve this function, including BERT, LSTM, and Word2Vec.

BERT (Bidirectional Encoder Representations from Transformers), developed in 2018, stands out in natural language understanding tasks. Its architecture, based on the transformer model and leveraging transfer learning, allows for an enhanced understanding of context compared to traditional neural networks such as LSTM or RNN. This typically results in improved performance in tasks like sentiment analysis [8]. Research conducted by Chiorrini et al. showcased the effectiveness of BERT-based techniques in text classification, achieving an accuracy of 92%[7].

In sentiment analysis, various kinds of neural networks are employed, which encompass Long Short-Term Memory (LSTM) networks as well as Convolutional Neural Networks (CNN). LSTM recognizes patterns in data and uses them to predict the most probable outcomes. This type of RNN operates based on the principle of storing each layer's output and then feeding it back into the system input to predict the same layer's output[8]. Conversely, CNN, typically used on images, also exhibits effectiveness in text analysis. It features layers of neurons that progressively extract complex features from input data[7]. One study demonstrated that a hybrid deep learning model combining LSTM, CNN, and Support Vector Machine (SVM) surpassed individual models in sentiment polarity analysis, though it required more computational resources [8]

Lastly, Word2Vec, a Google creation, generates word embeddings. It identifies the semantic relationships between words by examining the contexts of their usage [7]. A seven-layer framework incorporating CNN and Word2Vec was found to be superior to earlier models like MV-RNN and Recursive Neural Network, registering an accuracy rate of 45.4%[7].

It's pertinent to note that the choice of model for sentiment analysis often depends on the dataset's specific characteristics. Various models may exhibit superior performance under differing contexts and data types[8].

In our study titled "Unified Framework for Sentiment Analysis in Multiple Languages," we embarked on a comprehensive exploration of sentiment analysis across diverse linguistic landscapes. Beginning with a foundational understanding of sentiment analysis and its evolution from machine learning to deep learning, we

highlighted the roles of prominent models like BERT and LSTM. Multilingual sentiment analysis poses challenges in tokenization and language semantics. We conducted a comparative analysis of model performances, offering insights and discussions on the observed results using experimental studies on both English and non-English datasets. Our aim is to pave the way for a robust, unified approach to sentiment analysis across languages in the ever-expanding digital universe.

The remainder of this thesis is organized as follows. The fundamental concepts and literature review section addressed the challenges associated with multilingual analysis, we researched various sources, weighing their quality and reliability. The methodology chapter served as the backbone of our investigation. We delineated the processes involved, from the initial data set selection, text cleaning and feature extraction to leveraging translation tools like Google and Yandex for non-English content. Our results chapter includes the value of testing and evaluation metrics such as accuracy, precision, recall, and F1-Score. In the conclusion section, we discuss the strengths and limitations of each model in real-world scenarios.

SECOND CHAPTER

2. FUNDAMENTAL CONCEPTS AND LITERATURE REVIEW

2.1. SENTIMENT ANALYSIS AND ITS IMPORTANCE

Sentiment Analysis, which is often referred to as opinion mining, falls under the umbrella of Natural Language Processing (NLP) that involves the use of algorithms and techniques to identify, extract, and study subjective information from source materials. This information often comes in the form of opinions, appraisals, and emotions expressed in text data, and can provide valuable insights into the sentiments of individuals or groups towards specific topics, products, or services.

The importance of sentiment analysis lies in its wide range of applications across various domains. In business, for instance, it is used to understand customer sentiment towards products or services, thereby informing marketing strategies, product development, and customer service practices [1]. In politics, sentiment analysis can be used to gauge public opinion on policies or political figures, which can inform campaign strategies [9].

Furthermore, with the proliferation of social media and online review platforms, there is an abundance of user-generated content that can be analyzed for sentiment. This provides businesses, researchers, and policymakers with a wealth of real-time data that can be used to track sentiment trends and make informed decisions [10].

In the field of machine learning and artificial intelligence, sentiment analysis is a challenging and active research area. It involves understanding the nuances of human language, including sarcasm, irony, and context-specific meanings, making it a complex problem to solve [11].

2.2. MACHINE LEARNING IN SENTIMENT ANALYSIS

Machine learning has transformed the realm of sentiment analysis, introducing innovative techniques to decipher and understand sentiments from textual content. Deep learning models have demonstrated remarkable capabilities in enhancing sentiment analysis outcomes. Yet, these models necessitate abundant labeled data and meticulous design to operate optimally. For example, two cutting-edge deep learning designs, encompassing bidirectional LSTM and CNN, were introduced to categorize Persian sentiments in both multi-class and binary formats [12].

However, it's important to note that while machine learning has made significant strides in sentiment analysis, there are still challenges to overcome. These include the need for large amounts of annotated data, the complexity of language and context interpretation, and the need for precise model design.

2.3. DEEP LEARNING APPROACHES TO SENTIMENT ANALYSIS

O. Habimana et al. [12] explore various deep learning techniques suited for sentiment analysis tasks. They evaluated the efficacy of these methods on specific datasets. A notable mention is the TNet model, which fuses Bi-LSTM and CNN with a context-preserving transformation (CPT) layer, showcasing its prowess in formulating contextualized hidden representations and assimilating sentiment information. Additionally, the study underlined the standout performance of models embedded with RNN components in aspect-based sentiment analysis. It also recommended the utilization of advanced approaches like Transformers with bidirectional encoder representations (BERT), sentiment-focused word embedding structures, attention mechanisms inspired by cognitive frameworks, integration of common wisdom, reinforcement learning, and Generative Adversarial Networks to augment the performance of future models.

S. Minaer et al. [13] suggested an exhaustive exploration of over 150 distinct deep learning models crafted specifically for various text classification tasks. Such tasks span sentiment analysis, news classification, topic identification, query resolution, and the comprehension of natural language inferences. The paper offered a keen insight into the progression and efficiency of several deep learning structures, including

Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), attention-driven models, Transformers, and Capsule Networks. It underscores their application in areas like spam detection, sentiment evaluation, news categorization, user intent determination, and content vetting. Furthermore, it was reviewed over 40 esteemed text classification datasets, presenting an empirical performance appraisal of diverse deep learning structures using standard benchmarks. The discussion concludes with reflections on potential future research directions, emphasizing the emerging interest in models integrating both neural and symbolic methods to surmount the limitations of purely neural systems. This review serves as a pivotal resource for individuals eager to grasp the evolution in text classification via deep learning and its diverse implications.

2.4. BERT AND LSTM OVERVIEW

Long Short-Term Memory (LSTM), a novel recurrent network architecture designed to overcome the limitations of conventional backpropagation through time or real-time recurrent learning. These traditional methods often suffer from error signals that either explode or vanish, leading to unstable learning dynamics. LSTM, on the other hand, is capable of learning to bridge extensive time intervals, even in the presence of noisy and incompressible input sequences, without compromising short-time-lag capabilities. This is achieved through an efficient, gradient-based algorithm that enforces constant error flow through specially designed internal states of units. Potential challenges of LSTM were addressed, such as the 'abuse problem' and 'internal state drift and propose solutions including sequential network construction and output gate bias [14]. Through a series of experiments, LSTM is demonstrated to outperform previous approaches, solving complex tasks that no other recurrent net algorithm has been able to tackle. This highlights LSTM's potential as a powerful tool for handling long-time-lag tasks, capable of managing noise, distributed representations, and continuous values without the need for a predetermined number of states.

J. Devlin et al. [15] introduce BERT (Pre-training of Deep Bidirectional Transformers for Language Understanding) that a transformative language representation model distinguished by its capability to pre-train profound bidirectional

representations using unlabeled text. This avant-garde methodology empowers BERT to be refined with merely a single supplemental output layer. As a result, it adeptly addresses a diverse array of tasks, from question answering to language inference, bypassing the requirement for extensive task-oriented architectural adjustments. BERT's prowess is evidenced by its benchmark-setting performances across eleven natural language processing tasks, marking pronounced enhancements in metrics like the GLUE score, MultiNLI accuracy, and SQuAD v1.1. By adopting a "masked language model" for its pre-training objective, BERT can predict a concealed word exclusively from its surrounding context, sidestepping the unidirectional constraints typical of conventional language models. Furthermore, the model's introduction of the "next sentence prediction" task paves the way for jointly pre-training text-pair representations, curbing the reliance on intricately engineered task-specific designs. In summation, BERT heralds a monumental advancement in natural language processing, emphasizing the pivotal role of bidirectional pre-training in language representations and highlighting the efficacy of pre-trained models in obviating the necessity for task-tailored designs.

2.5. CHALLENGES IN MULTILINGUAL SENTIMENT ANALYSIS

Multilingual sentiment analysis, while promising, confronts several challenges rooted in linguistic diversity, cultural variances, and computational complexities. Each language brings its unique grammar, idioms, and sentiment-bearing phrases, which may not have direct counterparts in other languages. Moreover, cultural contexts can shape the sentiment conveyed by certain expressions, complicating universal sentiment understanding. Challenges also arise from sarcasm, idiomatic nuances, limited annotated data for some languages, and phenomena like code-switching. Further complicating matters are issues like translation inaccuracies, scalability concerns, domain-specific variations, and dialectal differences within languages, making the task both intricate and multifaceted.

Using SentiWordNet for Multilingual Sentiment Analysis by K. Denecke, an innovative methodology is presented. This method amalgamates three distinct strategies for cross-lingual sentiment analysis: the LingPipe Classifier, and two

adaptations of the SentiWordNet Classifier. The LingPipe Classifier is rooted in character-level language modeling, a method that might occasionally miss linguistic nuances. In contrast, the SentiWordNet Classifiers utilize lexical semantics harnessed from SentiWordNet. What's particularly captivating about the latter is its incorporation of machine learning techniques to hone its classification skills. Nevertheless, there's potential for further enhancement in the methodology, especially with a deeper exploration of its linguistic adaptabilities and trade-offs, more so for languages that differ considerably from English. Enriching the approach with a more intricate linguistic comprehension and tailoring lexicons for diverse languages could potentially amplify its performance in global sentiment analysis endeavors [16].

"SemEval-2023 Task 12: Emotion Interpretation in African Tongues (AfriSentiment-SemEval)" launches the inaugural Afrocentric SemEval collaborative project, emphasizing emotion interpretation in 14 under-resourced African dialects. This project, which attracted a plethora of entries, is segmented into three divisions: single-language categorization, multiple-language categorization, and zero-shot categorization. The teams that achieved the highest performance employed pre-trained linguistic models and emotional lexicons, shedding light on the possibilities and hurdles of utilizing cutting-edge NLP methodologies for languages that have typically been overlooked. This endeavor is part of a larger movement aimed at increasing linguistic diversity in the domain of Natural Language Processing [17].

Denecke [18] states a methodology for executing sentiment analysis on Twitter, utilizing a corpus autonomously assembled for training a sentiment classifier. This classifier, underpinned by the multinomial Naïve Bayes model, leverages N-gram and part-of-speech (POS) tags as features to discern positive, negative, and neutral sentiments within tweets. The study revealed that bigrams strike the most effective balance between data scope and the encapsulation of sentiment expression patterns. Furthermore, Denecke introduces salience and entropy as methodologies to filter prevalent n-grams, where salience displayed superior accuracy. The research implies that microblogging stands as a pivotal source of data for opinion mining and sentiment

analysis, with ensuing efforts concentrating on the evolution of a multilingual sentiment classifier.

The article "Multilingual Sentiment Analysis: A Contemporary Synthesis and Assessment of Methodologies" by Dashtipour and colleagues offers an extensive survey of the existing methodologies in the domain of multilingual sentiment analysis. The paper sheds light on various aspects such as data cleaning, characteristic attributes, and the principal tools employed in the analysis. It categorizes the methods into three groups: those based on textual data (corpus-based), those that utilize word sentiment dictionaries (lexicon-based), and a combination of the two (hybrid approaches). The researchers recreated eleven methodologies from their primary sources and evaluated them using the same pair of text data sets. The analysis revealed that the technique introduced by Singh and colleagues surpassed others in performance; however, it required significant computational power and was only evaluated with English text. Dashtipour and team acknowledged the scarcity of word sentiment dictionaries for multiple languages as a major obstacle in the field. Their aim was to create a multilingual data set encompassing Persian, Arabic, Turkish, and English languages, and to assess various methodologies by implementing them on this data set [19].

2.6. ROLE OF MACHINE TRANSLATION IN SENTIMENT ANALYSIS

Rada Mihalcea et al. [20] the scholars studied on the application of machine translation in generating materials and tools for analyzing subjectivity in languages besides English. The threesome presented three unique methods for generating corpora marked with subjectivity in the target language, by utilizing resources present in English. In the inaugural experiment, the authors automate the translation of training datasets that have been annotated manually, from the original language to the desired language. The ensuing experiment is built on the presumption that only a subjectivity annotation tool for the original language and an assortment of unprocessed texts in the original language are accessible. The texts in the original language are annotated for subjectivity through automation and subsequently translated to the desired language. The final experiment mirrors the second but alters the translation route. As a demonstration, the authors apply these methods to Romanian and Spanish and find that

the outcomes are encouraging, rivaling those achieved with corpora translated manually. The findings imply that machine translation is not only efficient but also potent in grasping the subjective nuances in text, with results nearly paralleling human-translated corpora with a marginal difference of 4% in F-measure. Furthermore, the authors address the significance of language-specific indicators in the analysis of subjectivity, postulating that languages enriched with inflections, such as Romanian, might furnish additional indicators for subjectivity.

THIRD CHAPTER

3. PROBLEM DEFINITION

Digital globalization introduces a tapestry of diverse sentiments expressed across multiple languages. These aren't mere words; they have significant business gauging political inclinations. Thus, crafting a genuinely global sentiment analysis solution demands a multilingual lens, but architecting such a solution introduces myriad challenges.

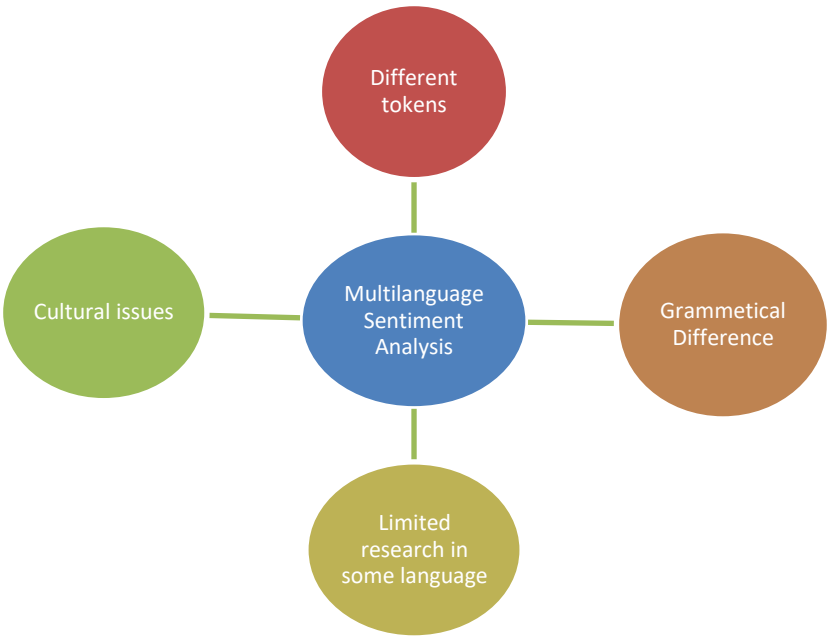

Figure 3.1 Multilingual sentiment analysis challenges

Models' adept in one language might not echo the same efficacy in another. Factors such as linguistic constructs, idiomatic expressions, and cultural connotations can influence model performance as shown in Figure 3.1. For example, a model trained on English datasets might falter when faced with an idiomatic expression exclusive to Arabic or Turkish. Each of the models BERT, LSTM, and Glove presents its unique advantages and limitations, and their efficacy can fluctuate drastically based on the

language and specific data attributes. Hence, a comprehensive and comparative evaluation of these models in the realm of multilingual sentiment analysis emerges as both a challenging and essential undertaking.

In this thesis, our objective is to address these challenges, embarking on a meticulous comparative analysis of BERT, LSTM, and Glove models for multilingual sentiment analysis. The paramount goal is to delineate the strengths and shortcomings of each model, decode their performance metrics across diverse languages, and furnish insights that can inform the judicious selection and application of these models for multilingual sentiment analysis endeavors. Furthermore, a pivotal facet of our research is to conceive methodologies or refinements that bolster the proficiency and precision of multilingual sentiment analysis, culminating in tools that are more reliable and comprehensive in the domain of natural language processing.

Collecting sentimental data across multiple languages demands an understanding of the unique linguistic characteristics. Preprocessing tools must recognize and handle elements like emojis, slang, or colloquial phrases. How can we ensure data consistency when the nature of data itself is so diverse?

As underscored in the abstract, machine translation services are pivotal to this research. Yet, every translation tool, with its capabilities and limitations, can vary in proficiency. For instance, while Google Translate might excel in certain languages, it might falter in others, and the same dichotomy can be observed with Yandex Translate. How can we harness the best attributes of each service to optimize sentiment analysis outcomes?

Given the intricacies surrounding multilingual sentiment analysis, the overarching challenge lies in designing a holistic framework that adeptly processes multiple languages. This framework should amalgamate the strengths of contemporary models and seamlessly fuse with machine translation tools, enabling fluid and efficient sentiment analysis.

As emphasized in the abstract, the effectiveness of sentiment analysis is not uniform across languages. This variability introduces unique challenges. Why might a sentiment translated from Turkish via Google Translate yield superior results compared to the same sentiment translated from French using the identical service? Such discrepancies necessitate in-depth scrutiny.

FOURTH CHAPTER

4. METHODOLOGY

The problem under investigation in this thesis concerns the complexities and challenges inherent to multilingual sentiment analysis, especially when comparing the performance of BERT, LSTM, and Glove models as shown in Figure 4.1. While the task of sentiment analysis has been deeply explored for English, the landscape changes dramatically when dealing with multiple languages, introducing complexities stemming from linguistic nuances and cultural contexts.

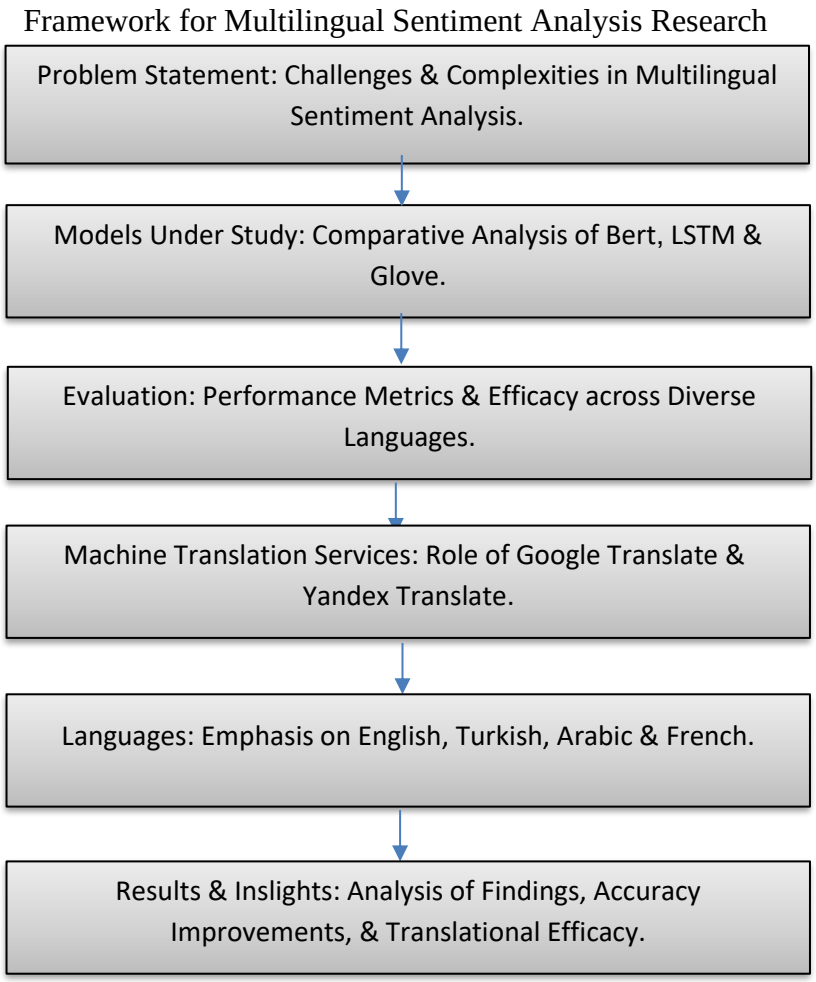

Figure 4.1: Methodology for Multilingual Sentiment Analysis

4.1. DATA SET

Table 4.1 Sources and languages distribution in dataset.

Language	Rows Count	Data Source	Description	Example
English	50k	Twitter and Newspapers	This dataset aggregates information from English-language sources. Twitter serves as a popular microblogging platform that captures real-time public sentiment on various topics. Newspapers contribute in-depth, structured reports and analyses, ensuring that the dataset represents both formal and informal registers of the English language.	Positive: "This product is absolutely amazing!" Negative: "I'm really disappointed with their service." Neutral: "The event will take place tomorrow."
Turkish	50k	Twitter and Hepsiburada	In the Turkish dataset, data is gathered from Twitter and Hepsiburada. Twitter offers immediate insights into public opinion, like its role in the English dataset. Hepsiburada, one of Turkey's leading e-commerce platforms, contributes customer reviews and detailed product descriptions, enriching the dataset with consumer sentiment and specialized information.	Positive: "Bu ürün harika!" Negative: "Müşteri hizmetleri çok kötü." Neutral: "Ürün yarın kargoya verilecek."
Arabic	50k	Newspapers	The Arabic dataset exclusively consists of data culled from newspaper articles. This choice ensures that the dataset maintains a formal tone and structure, focusing on detailed reports, opinions, and structured narratives prevalent in the Arab world.	Positive: "هذا المنتج رائع!" Negative: "خدمة العملاء سيئة." Neutral: "الحدث سيكون غدًا."
French	50k	Amazon	The French dataset is exclusively sourced from Amazon, one of the largest e-commerce platforms globally. This dataset is likely to include product reviews, customer feedback, and detailed product descriptions, serving as a valuable repository of consumer sentiment and product-specific information in the French language.	Positive: "Ce produit est incroyable!" Negative: "Je suis déçu par leur service." Neutral: "La livraison est prévue pour demain."

Table 4.1 provides details about the datasets used in the study, which include various languages. The datasets for English and Turkish were taken from multiple sources: Twitter and newspapers for English, and Twitter and Hepsiburada for Turkish. On the other hand, the Arabic dataset was exclusively collected from newspapers, capturing the structured and formal tones often found in Arabic reporting. Conversely, the French dataset was sourced from Amazon, which offers valuable insights into consumer sentiment and detailed product-related information in the French language. Each dataset comprises 50,000 rows of data, ensuring a substantial volume of information for analysis.[21]–[25].

4.2. TRANSLATION

Translation plays a pivotal role in multilingual sentiment analysis. As sentiment analysis has been predominantly studied in English, translating multilingual datasets into English offers a more consistent platform for model training and evaluation. This involves converting text from one language to another while preserving the original sentiment and context. It's crucial to choose the right translation tools or services, as even subtle nuances in translation can significantly affect the sentiment outcome. Machine translation tools like Google Translate or Yandex are commonly used for large datasets. However, depending on the quality and fidelity required, human translation or a combination of machine and human translation might be preferred. It's noteworthy that translation can introduce its own set of challenges, including the loss of cultural context or idiomatic expressions unique to a specific language. Therefore, understanding the limitations and biases of the chosen translation method becomes indispensable in a multilingual sentiment analysis framework.

4.2.1. Translation To English Using Google and Yandex

The translation of the collected data to English was an important step in the data preparation process. This was necessary because the sentiment analysis models were trained on English data, and translating the non-English data to English allowed these models to be applied to the data. The translation was done using both Google and Yandex translation APIs.

Google Translate API: The Google Translate API is a service that provides real-time translation between thousands of language pairs. It uses machine learning technologies to automatically recognize and translate text in different languages. The API supports a wide range of languages and can handle various types of text, making it a versatile tool for multilingual data processing. The Google Translate API is a part of Google Cloud services and requires an API key for access. It is a paid service, but it offers a free tier with limited usage [26].

Yandex Translate API: The Yandex Translate API is another service that provides machine translation between different languages. It is developed by Yandex, a Russian multinational corporation specializing in Internet-related products and services. The Yandex Translate API supports a variety of languages and can translate

both text and webpages. Like the Google Translate API, it requires an API key for access and offers both free and paid tiers [27].

Here is a pseudo-code that outlines the steps of using the translation APIs to translate the data:

1) Initialize the Google and Yandex translation services.
2) Input the first text in the original language.
3) While there are still texts to be translated:
    a) Translate the text to English using Google Translate.
    b) Simultaneously, translate the same text to English using Yandex Translate.
    c) If either Google or Yandex Translate fail or return errors, handle the failure. This could involve logging the failure, using a default value, or skipping the text.
    d) Store or process the translated text from both services. This could involve adding it to a dataset, using it for model training, or any other necessary processing.
    e) Input the next text in the original language.
4) Once all texts have been translated by both services, proceed to the next step in the data preparation process.

This process ensures that all texts are translated to English by both Google and Yandex translation services. This approach can provide a more comprehensive translation, as it leverages the strengths of both services. It also handles any failures or errors that occur during translation, ensuring that the data preparation process can continue even if some texts cannot be translated.

4.3. DATA PREPROCESSING TECHNIQUES

Data preprocessing is a critical step in the pipeline of any machine learning project. It involves cleaning and transforming raw data into a format that can be easily ingested and used by machine learning models. In the context of this study, which involves sentiment analysis on multilingual data, the preprocessing steps are even

more crucial. The preprocessing involved two main steps: text cleaning and feature engineering.

4.3.1. Text Cleaning

Text cleaning is the process of purifying the text data by removing unnecessary or distracting elements. This is a crucial step in natural language processing tasks, as it helps in reducing the noise in the data and makes the data more understandable for the machine learning models.

The text cleaning process in this study involved several steps:

1) Case Normalization**:** The first step was to convert all the text to lower case. This is done to ensure uniformity in the data and to prevent the models from treating the same words in different cases as different words.
2) Removing Non-Alphabetic and Non-Numeric Characters**:** The next step was to remove all non-alphabetic and non-numeric characters from the text. This includes special characters and symbols that do not contribute to the sentiment of the text.
3) Text Normalization**:** This step involved removing accents from the text. Accents can create multiple versions of the same word, so removing them helps in reducing the complexity of the data.
4) Removing Links and Usernames**:** Any website links and usernames in the text were removed. These elements are usually specific to a particular text and do not contribute to the overall sentiment.
5) Removing Punctuation Marks**:** Punctuation marks were removed from the text. While punctuation can sometimes convey sentiment (e.g., exclamation marks may indicate excitement or anger), in this study they were removed for simplicity.
6) Removing Stop Words**:** Stop words are commonly used words that do not carry much meaningful information for the sentiment analysis task. These words were removed from the text to reduce the dimensionality of the data and to focus on the words that are more likely to convey sentiment.

7) Removing Tabs and New Lines**:** Any tabs and new lines in the text were removed to ensure a clean, continuous text.
8) Lemmatization**:** Finally, the words in the text were lemmatized. Lemmatization is the process of reducing words to their base or dictionary form. For example, the words 'running', 'runs', and 'ran' are all changed to 'run'. This helps in reducing the complexity of the data and allows the model to treat different forms of the same word as one.

4.3.2. Feature Engineering

In our study, we utilized the BERT model for feature engineering. BERT provides a rich and context-aware representation of the input text. These representations, also known as embeddings, capture the semantic meanings of words and their context in the text, which allows our model to understand and learn from the text data.

The BERT model is pre-trained on a large corpus of text and can generate high-quality word embeddings that capture a wide range of syntactic and semantic relationships. By using BERT, we can leverage these pre-trained embeddings to improve the performance of our sentiment analysis models.

The BERT model generates an embedding for each token in the input text. These embeddings are vectors of numbers, where each number represents a different feature of the token. For example, one feature might represent the token's part of speech, another might represent its tense, and so on. These features are learned by the model during pre-training and can capture complex patterns in the text.

In our case, we used the BERT model to generate embeddings for the texts in our dataset. We then used these embeddings as input to our sentiment analysis models. This process involves the following steps:

1) Load the pre-trained BERT model and tokenizer.
2) Define a function to encode the texts:
   a) The function takes a list of texts and a tokenizer as input.

b) It tokenizes the texts using the tokenizer.

c) It encodes the tokenized texts using the BERT model, generating an embedding for each token.

d). It returns the embeddings as a numpy array.

3*)* Apply the encoding function to the texts in our dataset.

This process transforms the raw text data into a format that can be used by sentiment analysis models. The encoded texts are numerical representations of the texts, where each word or token in the text is mapped to a unique vector of numbers. These numerical representations capture the semantic meanings of the words and their context in the text, which allows the models to understand and learn from the text data.

4.3.3. Tokenization

Tokenization is the process of breaking down text into smaller units, typically words or phrases. These tokens are the building blocks of natural language and understanding them can lead to a better understanding of the text.

Tokenization can be as simple as splitting the text by white spaces and punctuation in English. However, for languages that do not use spaces or for tasks that require understanding of phrases or idioms, more complex tokenization methods are required.

Tokenization is a crucial stage in text preprocessing, which entails decomposing text into single words or tokens. Arnold's 2017 publication offers a structured data framework for natural language processing, incorporating tokenization along with other annotation tools such as tagging parts of speech, identifying named entities, linking entities, analyzing sentiment, parsing dependencies, resolving coreferences, and extracting information [28].

4.4. IMPLEMENTATION DETAILS OF MODELS

In our study, we used three different models for sentiment analysis: LSTM (Long Short-Term Memory), DistilBERT, and GloVe (Global Vectors for Word

Representation). Each of these models has its own strengths and is suited to different types of text data and sentiment analysis tasks.

4.4.1. LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is capable of learning long-term dependencies in data, making it well-suited to text data. LSTM networks are composed of memory cells that can maintain information in memory for long periods of time. This makes them particularly good at understanding context and maintaining state over long sequences, which is crucial for understanding the sentiment of a piece of text.

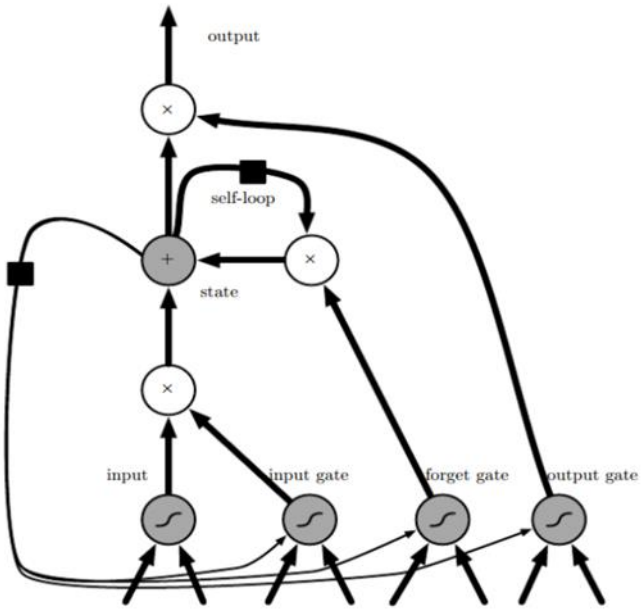

Figure 4.2: LSTM Structure [29]

In Figure 4.2, the LSTM cell is depicted with its essential components: the input, forget, and output gates, as well as the cell state. Each of these elements has a specific role in the cell's operation.

1) Input Gate: Controls the extent to which the current input updates the cell state.
2) Forget Gate: Determines how much of the prior cell state is retained or forgotten.
3) Output Gate: Regulates how much of the current cell state is transmitted to the subsequent LSTM cell.

The cell state serves as a long-term memory vector, storing crucial information about the sequence. All gates within the LSTM cell operate as miniature neural networks. They receive three types of input: the present sequence input, the preceding hidden state, and the preceding cell state. Each gate then outputs a scalar value ranging between 0 and 1. A value of 0 signifies that the gate is closed, blocking the flow of information, while a value of 1 means the gate is fully open, allowing all information to pass. These gates collectively manage the information flow within the network. They enable the LSTM to learn long-term dependencies, making it highly effective for applications like machine translation, speech recognition, and text summarization.

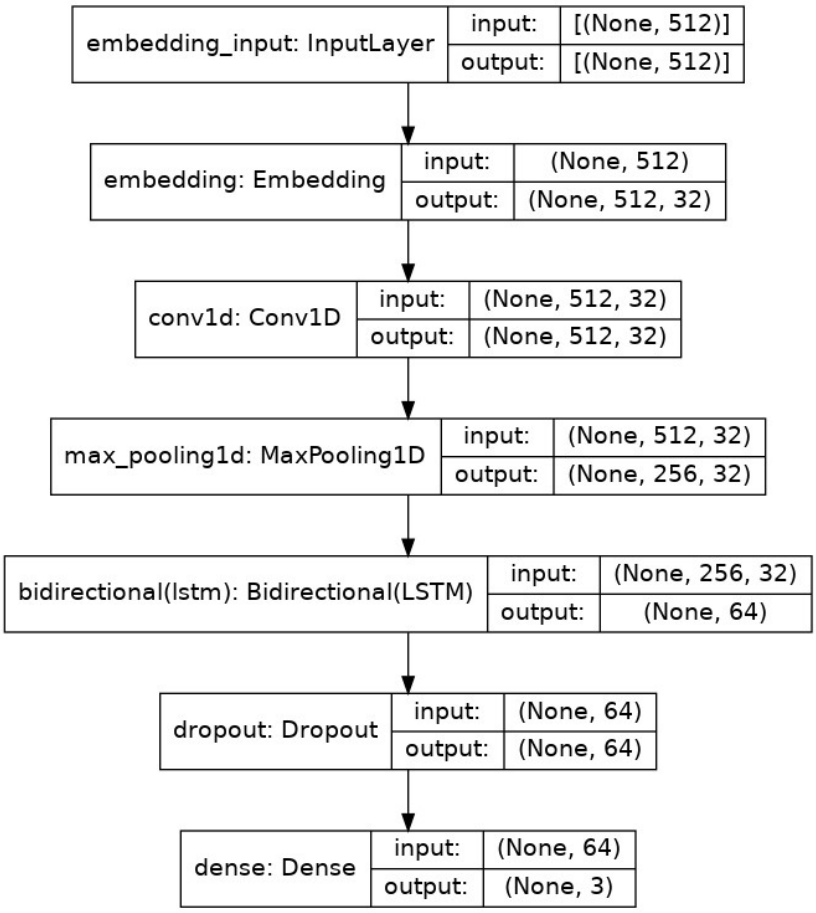

Figure 4.3: Model Layers

At Figure 4.3 model layers the LSTM model for sentiment analysis was built and trained using the following steps:

1) Import the necessary libraries and modules: This step is necessary to provide the tools needed to build and train the model. Keras, a high-level neural networks API, was used for this purpose.
2) Define the vocabulary size, embedding size, and other hyperparameters: These parameters are crucial for the model's performance. The vocabulary size is the number of unique words in the text data. The embedding size is the size of the vector space in which words will be embedded. Other hyperparameters like learning rate, decay rate, and momentum are used to control how the model learns.
3) Initialize the Stochastic Gradient Descent (SGD) optimizer: The optimizer is the technique employed for modifying the characteristics of the neural network, including aspects like weights and the learning rate, with the aim of minimizing errors. Stochastic Gradient Descent, which is a category of optimization algorithms, was utilized to circumvent getting stuck in local minimums throughout the training process.
4) Build the LSTM model: This involves adding several layers to the model:

1) Embedding layer: This layer turns positive integers (indexes) into dense vectors of fixed size. It's used here to convert words into vectors of numbers so they can be processed by the model.
2) Conv1D layer: Convolutional layers are used to extract features from a fixed length of words (defined by the kernel size). Here, it's used to detect local patterns or features in the input sequences, like specific sets of words or phrases that could be relevant for determining sentiment.
3) MaxPooling1D layer: Pooling layers are used to reduce the dimensionality of the model, helping to prevent overfitting. Max pooling does this by taking the maximum value of the area it's applied to.
4) Bidirectional LSTM layer: LSTM layers are a type of recurrent neural network that are good at learning from long-term dependencies. They're bidirectional here because they process the data from past to future and from future to past. This helps the model to learn from the context from both before and after a word.
5) Dropout layer: Dropout is a regularization technique that prevents overfitting by randomly setting a fraction rate of input units to 0 at each update during training time.

6) Dense layer: This is the output layer, which produces the probabilities for each sentiment class using the softmax activation function.

7) Compile the model with the optimizer, loss function, and metrics**:** This step configures the learning process of the model. The loss function measures the error of the model, the optimizer uses this error to adjust the model's weights, and the metrics are used to monitor the performance of the model.

8) Train the model on the training data for a certain number of epochs: This is where the model learns from the data. It adjusts its weights based on the data it sees. An epoch is one complete pass through the entire training dataset. The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset.

Each of these steps plays a crucial role in building a model that can effectively understand and learn from the text data for sentiment analysis. The trained model can then be used to predict the sentiment of new, unseen texts.

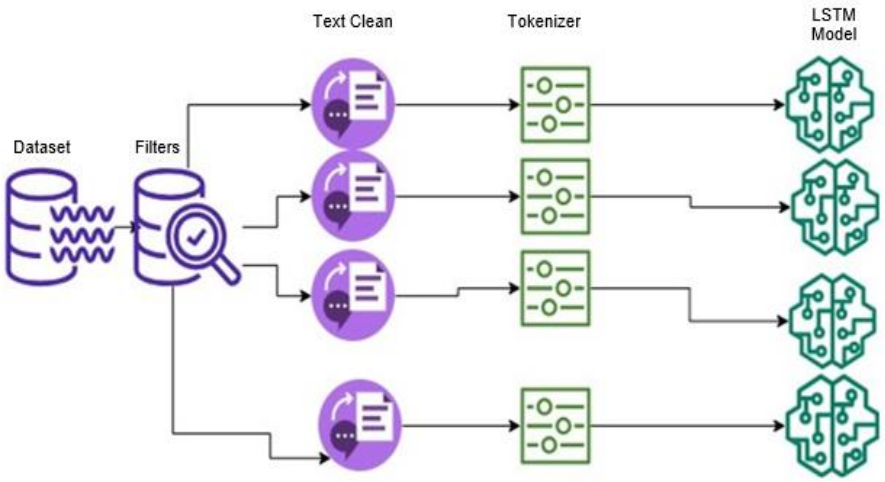

Figure 4.4: An Overview of How LSTM Processes Multilingual Text.

Figure 4.4 elucidates the intricate steps involved in the text cleaning process, which is particularly tailored for a dataset comprising text in four distinct languages: Turkish, Arabic, French, and English. This detailed process can be broken down into the following stages:

1) Initial Filtering:

Before any processing occurs, the primary dataset undergoes rigorous scrutiny. Any duplicated data or data that doesn't align with the study's context is systematically eliminated. This step is crucial for ensuring that the foundation of the data processing is robust and free of redundancies.

2) Language Segregation:

After the initial filtration, the consolidated dataset is divided into four separate datasets. Each of these datasets corresponds to one of the four languages in focus: Turkish, Arabic, French, and English. This division ensures that the specific nuances and characteristics of each language are accounted for during the cleaning process.

3) Language-specific Cleaning:

With each language set isolated, a dedicated cleaning process commences. This step aims at refining the dataset further by eliminating potential disturbances in the text:

a) Punctuation Removal: Any punctuation marks, which do not contribute semantically to language patterns, are removed.
b) Stop Word Elimination: Commonly used words (e.g., 'and', 'the', 'is') which do not carry significant meaning on their own are excluded.
c) Noise Filtering: Any other unexpected noise, such as numbers or symbols unrelated to the text content, is purged.

4) Tokenization:

Following the meticulous cleaning process, each refined dataset undergoes tokenization. In this step, continuous strings of text are broken into discrete units, typically individual words, or sometimes meaningful phrases. This granularity allows the subsequent model to understand and learn patterns at the word level, capturing the essence of each language's structure.

5) LSTM Modeling:

The tokenized data is then fed into a Long Short-Term Memory (LSTM) model. LSTMs are a type of recurrent neural network renowned for their ability to remember and predict sequences, making them particularly well-suited for text processing. As the model ingests the tokenized text, it embarks on a learning journey to discern the intricate patterns inherent to each language.

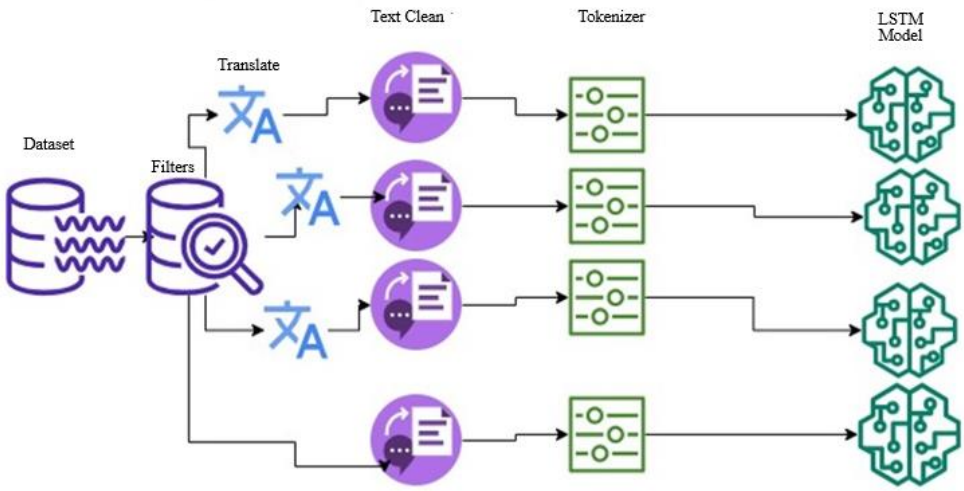

Figure 4.5: An Overview of How LSTM Processes Multilingual Text with Translation Service.

In Figure 4.5 we break down the process of how LSTM deals with multilingual text when integrated with a translation service. The steps are:

1) Initial Filtering:

As a preliminary step, the entire data set undergoes an evaluation. Data that appears duplicated or not pertinent to the research context is systematically removed. This ensures that the ensuing processing stages handle only meaningful, unique data, providing a solid foundation.

2) Translation Process:

After the initial filtration, the dataset experiences a translation phase. Texts from non-English languages - namely Turkish, Arabic, and French - are translated into English. For this pivotal step, renowned translation services like Google Translate and Yandex Translate are utilized. This transition to a singular language facilitates easier data management and subsequent processing.

3) Language Segregation:

Once translated, the dataset, now enhanced with additional English translations, is bifurcated into separate datasets for each language: Turkish (and its English translation), Arabic (and its English translation), French (and its English translation), and the original English texts. This categorization ensures that each language's unique intricacies and attributes are aptly addressed in the next steps.

4) Language-specific Cleaning:

1) Each isolated dataset then undergoes a bespoke cleaning procedure:
2) Punctuation Removal: Extraneous punctuation marks are discarded.
3) Stop Word Elimination: Generic words that typically do not convey significant standalone meaning are filtered out.
4) Noise Filtering: Unrelated numbers, symbols, or other potential disturbances in the text are diligently purged.

5) Tokenization:

The cleansed datasets are then subjected to tokenization. Text strings are broken down into individual units, generally words or meaningful phrases. This granularity ensures that the ensuing neural network can pinpoint and learn language structures at the micro-level.

6) LSTM Modeling:

The tokenized entities are channeled to a Long Short-Term Memory (LSTM) model. LSTMs, being a specialized form of recurrent neural networks, excel in sequence predictions. As this model acquaints itself with the tokenized text, it commences its learning trajectory to fathom the embedded linguistic patterns of each language.

4.4.2. DistilBERT

DistilBERT is a smaller, faster, cheaper, and lighter version of the BERT model. It retains over 95% of BERT's performance while being 60% smaller and 60% faster. This makes it an excellent choice for tasks where computational resources are limited, but high performance is still required.

In the context of multilingual sentiment analysis, DistilBERT is particularly valuable because it can handle text in multiple languages. This is due to the fact that it is trained on a multilingual corpus, allowing it to understand and generate representations for text in many different languages. This makes it a powerful tool for sentiment analysis tasks that involve text in multiple languages.

The role of DistilBERT in transfer learning is also significant. Transfer learning is a machine learning technique where a pre-trained model is used on a new,

similar problem. In this case, DistilBERT, which has been pre-trained on a large corpus of text, is fine-tuned on a specific task like sentiment analysis. This allows it to leverage its pre-existing knowledge of language to perform well on the task, even with a relatively small amount of task-specific training data.

The pseudo-code for the provided code be as follows:

1) Import the necessary libraries and modules.
2) Read the datasets and combine them.
3) Preprocess the data using the DistilBERT tokenizer.
4) Split the data into training and validation sets.
5) Convert the data to tensorflow tensors.
6) Configure the DistilBERT model with the necessary parameters.
7) Enable mixed precision training for faster computation.
8) Define the model with the softmax activation in the output layer.
9) Compile the model with the Adam optimizer and Categorical Crossentropy loss function.
10) Define the learning rate schedule and decay steps.
11) Train the model on the training data for a certain number of epochs, using early stopping to prevent overfitting.
12) Evaluate the model on the training and validation data.
13) Print the accuracy, precision, and recall of the model on the training and validation data.
14) Save the trained model for future use.

Each of these steps is crucial for building and training a DistilBERT model for sentiment analysis. The model is trained to understand the sentiment of text in multiple languages and can then be used to predict the sentiment of new, unseen text.

4.4.3. GLOVE

The GloVe model for sentiment analysis is particularly effective due to its ability to capture both global and local semantic relationships between words. This is achieved by leveraging global word-word co-occurrence statistics from a corpus to generate word vectors. The generated word vectors capture the semantic meaning of

words based on their context in the corpus, which is crucial for understanding the sentiment of a piece of text.

The GloVe model for sentiment analysis was built and trained using the following steps:

1) Import the necessary libraries and modules: This step is necessary to provide the tools needed to build and train the model.
2) Define the vocabulary size, embedding size, and other hyperparameters: These parameters are crucial for the model's performance. The vocabulary size is the number of unique words in the text data. The embedding size is the size of the vector space in which words will be embedded.
3) Load the GloVe word vectors: GloVe provides pre-trained word vectors trained on various large corpora. These vectors are loaded into a dictionary from word to vector.
4) *Create an embedding matrix:* The embedding matrix is created by mapping each word in the dataset's vocabulary to its corresponding vector in the GloVe word vectors. If a word is not in the GloVe vocabulary, it is assigned a random vector.
5) Build the model: The model is built by adding several layers, including an embedding layer that uses the embedding matrix as its weights, a bidirectional LSTM layer, and a dense output layer.
6) Compile and train the model: The model is compiled with a loss function, an optimizer, and metrics, and then it is trained on the training data.

The trained model can then be used to predict the sentiment of new, unseen texts. The GloVe embeddings provide a rich representation of the words based on their semantic and syntactic relationships, which helps the model to understand the sentiment of the text.

In the context of transfer learning, GloVe plays a significant role. Transfer learning refers to a technique in machine learning in which a model that was originally trained for one particular task is adapted to be used as the foundation for a different task. This strategy is widely employed in deep learning, especially in fields like computer vision and natural language processing, where models that have already been trained are repurposed as initial frameworks for new tasks. With GloVe, we have pre-trained word embeddings that we can use, and we can also fine-tune these embeddings based on our specific task. This allows us to leverage the semantic and syntactic relationships that the embeddings have learned from a large corpus of text, while also customizing them to our specific task. This can often result in better performance than training a model from scratch or using one-hot encoded words.

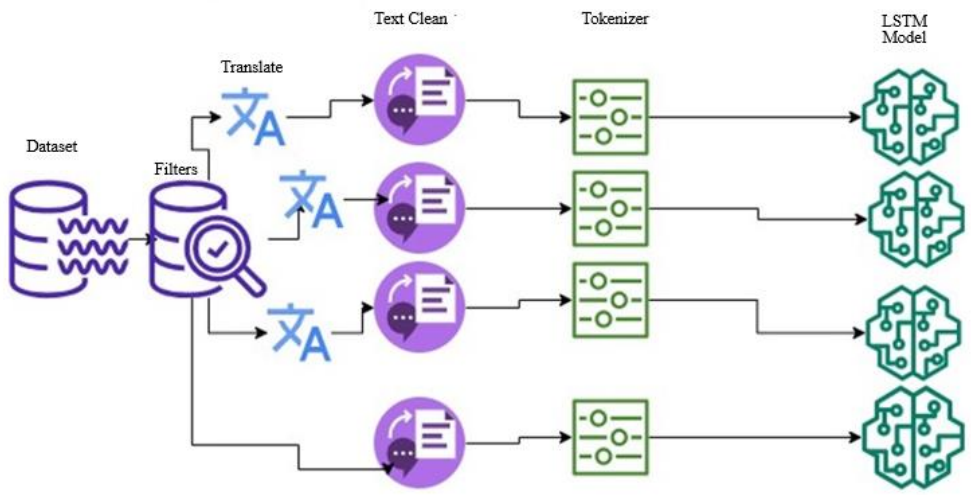

Figure 4.6: An Overview of How LSTM Processes Multilingual Text with Translation Service using GloVe Embeddings.

In Figure 4.6 presents a meticulous overview of the process of cleaning and processing a dataset that includes text from four distinct languages: Turkish, Arabic, French, and English. The procedure is articulated in the following stages:

1) Initial Filtering:

The dataset is subjected to a rigorous examination right at the outset. Redundant or non-contextual data is systematically eliminated, ensuring that subsequent stages work on unique, relevant data.

2) Translation Process:

After ensuring data purity, texts in non-English languages (i.e., Turkish, Arabic, and French) are translated to English using trusted translation services like Google Translate and Yandex Translate. This standardization to English facilitates a more streamlined processing in the following steps.

3) Language Segregation:

Post-translation, the dataset is segmented into individual datasets corresponding to each language, each inclusive of the original text and its English translation: Turkish and its English counterpart, Arabic and its English counterpart, French and its English counterpart, and original English.

4) Text Cleaning:

Each segmented dataset is then put through a specialized cleaning regimen:

1) Punctuation Removal: Superfluous punctuation marks are removed.
2) Stop Word Elimination: Common words, which might act as noise in pattern recognition, are filtered out.
3) Noise Filtering: Miscellaneous disturbances, like unrelated numbers or symbols, are diligently eliminated.

5-Tokenization with GloVe Embeddings:

Once cleaned, the datasets undergo tokenization using the GloVe model, specifically employing the "Glove.6B.300d.txt" embeddings. GloVe embeddings transform words or phrases into 300-dimensional vectors, capturing semantic relationships between words. This means that instead of just breaking down texts into individual words, they are now represented as dense vectors which encapsulate more semantic meaning, enhancing the LSTM's ability to discern linguistic patterns.

4.4.4. GRU

The Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture that was proposed to solve the vanishing gradient problem and make training deep networks easier. The GRU does this by introducing gating mechanisms that allow for longer sequences of information to be captured.

In essence, the GRU has two gates:

1) Update Gate: This gate decides how much of the previous state should be carried over to the current state. It helps the GRU unit to determine the amount of past information to be passed to the future.

2) Reset Gate: It determines how much of the previous state is forgotten. By doing this, it can decide how much new information from the current input should be stored in the memory.

These gates enable the GRU to capture dependencies over different time scales. They allow the model to maintain information for longer periods or forget it more rapidly, depending on the nature of the data and the specific task.

The architecture of the GRU makes it particularly suitable for sequences where it is essential to capture information over extended periods, such as in time series prediction, natural language processing tasks, and more [30].

4.5. EXPERIMENTAL SETUP FOR MODEL COMPARISONS

The experimental setup for model comparisons, which includes the preparation of training and testing data and the tuning of hyperparameters, is a crucial aspect of our research. This setup directly impacts the performance of the models we are comparing and, consequently, the validity of our results.

4.5.1. Training and Testing Data

The act of segregating data into distinct sets for training, validation, and testing is a foundational practice in machine learning experiments. Its importance stems from the need to ensure that a machine learning model is trained to recognize patterns while also proficiently generalizing these patterns to unseen data.

Initially, the 'clean_text' column in the dataset serves as the input features, while the 'category' is transformed into a format suitable for machine learning, serving as the target variable. The dataset is then divided, with a majority (for instance, 80%) allocated for training purposes and the remainder (e.g., 20%) set aside for testing. This

testing set serves as a proxy for real-world, unseen data, allowing for an assessment of the model's ability to generalize.

However, the segmentation process includes another layer. From the designated training data, a subset (such as 25%) is further separated to act as a validation set, leaving the rest for actual training. This stratification ensures that there is a dedicated segment of data available for evaluating the model's performance during its training phase, without compromising the integrity of the testing set.

The essence of this methodological division is multifaceted. By training the model on a dedicated segment and subsequently evaluating its proficiency on a separate, untouched portion, researchers can gain insights into its potential real-world performance. This partitioning is instrumental in confirming that the model is not merely memorizing the training data (a pitfall known as overfitting), but is genuinely skilled at making predictions for new datasets.

The inclusion of a validation set refines this process further. Acting as an intermediary during the training phase, the validation set provides insights into the model's evolving performance. Should the model's learning trajectory appear subpar, or if it starts memorizing the nuances of the training set too closely, feedback from the validation set serves as a cue to adjust the model parameters. This iterative feedback loop, made possible by the validation set, is crucial for fine-tuning the model, ensuring optimal performance, and guarding against overfitting.

4.6. CRITERIA FOR EVALUATING MODEL PERFORMANCES

To assess the effectiveness of the sentiment analysis models implemented in this study, it is crucial to establish a set of criteria that can provide a comprehensive evaluation of their performance. The following subsections detail the metrics used in this study: Accuracy, Precision, Recall, and F1-Score.

4.6.1. Accuracy

Accuracy is one of the most straightforward metrics used in machine learning. It is calculated as the ratio of the number of correct predictions to the total number of predictions. In the context of sentiment analysis, a correct prediction means that the model correctly identified the sentiment of a text sample. While accuracy can provide a general idea of how well a model is performing, it may not be the most reliable metric in cases where the dataset is imbalanced.

4.6.2. Precision, Recall, and F1-Score

Precision, recall, and F1-score are metrics that provide a more nuanced understanding of a model's performance, particularly in cases where the data may be imbalanced.

Precision is the ratio of true positive predictions (i.e., the model correctly predicted the positive class) to the total number of positive predictions made by the model. A high precision indicates that when the model predicts the positive class, it is likely to be correct.

Recall, also known as sensitivity or true positive rate, is the ratio of true positive predictions to the total number of actual positive instances in the data. A high recall indicates that the model is good at detecting positive instances.

The F1-score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall. An F1-score is particularly useful in cases where it is important to balance false positives and false negatives.

These metrics collectively provide a comprehensive view of the model's performance, allowing for a more detailed comparison and analysis of the different sentiment analysis models implemented in this study.

FIFTH CHAPTER

5. RESULTS

The experimental studies conducted in this research aim to evaluate the performance of the sentiment analysis models on both English and non-English data. The non-English data was translated to English using two different translation services: Google Translate and Yandex Translate. The following subsections detail the experiments conducted using these translation services.

5.1. EXPERIMENTS ON ENGLISH DATA AND NON-ENGLISH DATA

The experiments were conducted on both English and non-English data to assess the performance of the sentiment analysis models in a multilingual context. The non-English data was translated to English using Google Translate and Yandex Translate.

Table 5.1 Results Based on Google Translation by LSTM

Language	Accuracy	Precision	Recall	F1 Score
English	**91.5%**	91.7%	91.2%	91.5%
Turkish	85.6%	86.5%	84.5%	84.0%
Arabic	65.4%	65.4%	65.4%	66.7%
French	57.4%	65.7%	41.7%	48.4%
Turkish translated	**86.7%**	87.6%	85.8%	87.5%
Arabic translated	71.8%	71.8%	71.8%	72.7%
French translated	79.0%	79.1%	79.1%	81.7%

Table 5.2 Results Based on Yandex Translation by LSTM

Language	Accuracy	Precision	Recall	F1 Score
English	**91.5%**	91.7%	91.2%	91.5%
Turkish`	85.6%	86.5%	84.5%	84.0%
Arabic	65.4%	65.4%	65.4%	66.7%
French	57.4%	65.7%	41.7%	48.4%
Turkish translated	**88.7%**	89.5%	88.1%	89.5%
Arabic translated	71.0%	71.0%	71.0%	72.3%
French translated	78.6%	78.6%	78.6%	81.4%

Each table lists the results of evaluating the performance of the sentiment analysis models on seven datasets. It includes the four datasets for four different

languages (English, Turkish, Arabic, and French) and additional three datasets after the translation to the English language (Turkish to English, Arabic to English, and French to English).

As shown in the Tables 5.1 and 5.2, the proposed framework achieved enhancements for all languages as it increases the accuracy from 85.6% to 86.7% for Turkish, from 65.4% to 71.8% for Arabic and from 57.4% to 79.0% for French when using Google Translate. Similarly, when using Yandex Translate, the accuracy increased from 85.6% to 88.7% for Turkish, from 65.4% to 71.0% for Arabic and from 57.4% to 78.6% for French. These results confirm the effectiveness of our proposed framework and show the significant enhancement that can be obtained when both the translation and sentiment analysis algorithms are merged.

It is also noted that the proposed model still shows relatively lower performance in handling Arabic text compared with other languages, with an accuracy of 71.8% for Google Translate and 71.0% for Yandex Translate.

Table 5.3 Results Based on Google Translation by Glove

Language	Accuracy	Precision	Recall	F1 Score
English	**91.5%**	91.7%	91.2%	91.5%
Turkish translated	88.1%	89.1%	87.5%	88.5%
Arabic translated	69.8%	69.8%	69.8%	69.8%
French translated	72%	72%	72%	72%

Table 5.4 Results Based on Yandex Translation by Glove

Language	Accuracy	Precision	Recall	F1 Score
English	**91.5%**	91.7%	91.2%	91.5%
Turkish translated	93%	90%	89%	89%
Arabic translated	69%	70%	69%	69%
French translated	73%	73%	73%	73%

A GloVe model was also utilized for sentiment analysis. The GloVe model is an unsupervised learning algorithm designed to obtain vector representations for

words. Using Google Translate as listed in Table 5.3, the GloVe model yielded an accuracy of 88.1% for Turkish, 69.8% for Arabic, and 72% for French. Meanwhile, as illustrated in Table 5.4, with Yandex Translate, the accuracy values witnessed a change, registering 93% for Turkish, 69% for Arabic, and 73% for French.

Comparing the results of the two models, it can be observed that the GloVe model achieved higher accuracy for Turkish data when using Yandex Translate, compared to the LSTM model. However, the LSTM model performed better on Arabic and French data when using both Google Translate and Yandex Translate.

The significance of choosing the right model and translation service is underlined by our results, which demonstrate their profound impact on the performance of sentiment analysis in a multilingual context. Our findings emphasize the critical importance of an efficient translation service to extract the best results in sentiment analysis tasks.

One standout observation is that the English language, having garnered substantial attention from the research community, exhibits superior performance in sentiment analysis models. The inherent structures and vocabularies of English appear more amenable to sentiment analysis algorithms. This raises a compelling strategy: converting all texts into a common reference language, preferably English, before deploying sentiment analysis. This translation process, as our results suggest, can potentially lead to improved outcomes compared to direct sentiment analysis on non-English texts.

In our pursuit of multilingual sentiment analysis, we adopted the DistilBERT model, a streamlined version of Google's BERT a transformer-based approach tailored for pre-training in natural language processing tasks. Complementing this, we leveraged the power of transfer learning, a technique wherein a pre-trained model for one task is repurposed for another.

To test the viability of our approach, we amalgamated all datasets, encompassing Arabic, French, Turkish, and English languages, along with their

translations. Training the DistilBERT model on this composite dataset, we gauged its performance using metrics such as accuracy, precision, and recall for both training and validation sets.

Table 5.5 Results of DistilBERT

Criterion	Train	Test
Accuracy	94.2%	92.56%
Precision	94.55%	92.87%
Recall	93.82%	92.24%
F1 Score	94.18%	92.55%

Table 5.5 showed that the DistilBERT model achieved a training accuracy of 94.20% and a validation accuracy of 92.56%. The precision of the model was 94.55% for the training set and 92.87% for the validation set. The recall was 93.82% for the training set and 92.24% for the validation set.

These results suggest that the DistilBERT model, combined with transfer learning, provides a robust solution for multilingual sentiment analysis. The high accuracy, precision, and recall scores indicate that the model is capable of correctly classifying the sentiment of text data in multiple languages, and that it generalizes well to unseen data.

5.2. COMPARATIVE ANALYSIS OF MODELS

5.2.1. Performance Comparison

We graphically illustrated the performance of an LSTM model over the course of 17 epochs, evaluating both training and validation accuracy as shown Figure 5.1. These measures serve as key indicators: training accuracy reflects how well The model performs on the data it was trained on, while validation accuracy gauges its ability to generalize to new, unseen data.

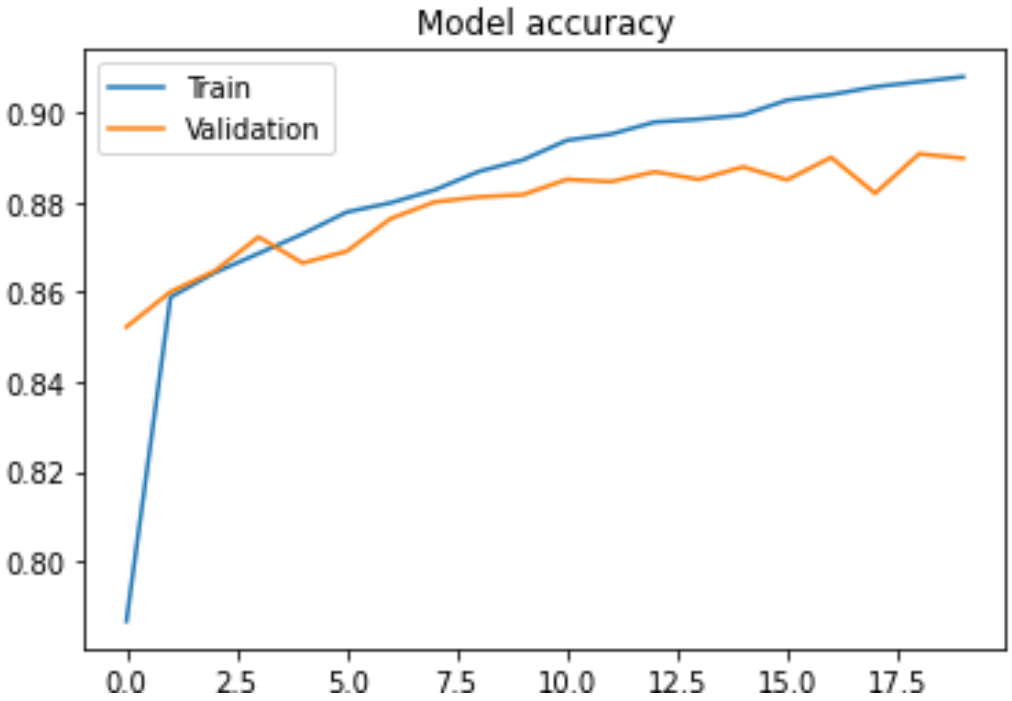

Figure 5.1 Accuracy vs Epochs for LSTM Model.

Training Accuracy:

The model starts off strong with a training accuracy of 0.90. This figure steadily climbs, peaking at 0.92 within the initial five epochs. After this incremental phase, the accuracy plateaus and remains relatively stable in subsequent epochs. However, a significant dip is observed after the 12th epoch, signaling a decrease in training accuracy.

Validation Accuracy:

Concurrently, validation accuracy initiates at 0.88, not far behind the training accuracy. Following a similar trajectory, it rises to reach 0.90 by the 5th epoch. Like its training counterpart, the validation accuracy also stabilizes and presents a flat trend in the following epochs. Intriguingly, after the 12th epoch, even as it decreases, the validation accuracy still surpasses the training accuracy.

Overfitting Analysis:

The gap between the training and validation accuracies is indicative of overfitting. Known as the 'overfitting gap,' this divergence occurs when a model

becomes excessively tailored to its training data, losing its ability to adapt to new, unseen data. This phenomenon is particularly evident after the 12th epoch, where the validation accuracy starts to decline, suggesting that the model may have over-learned from the training data.

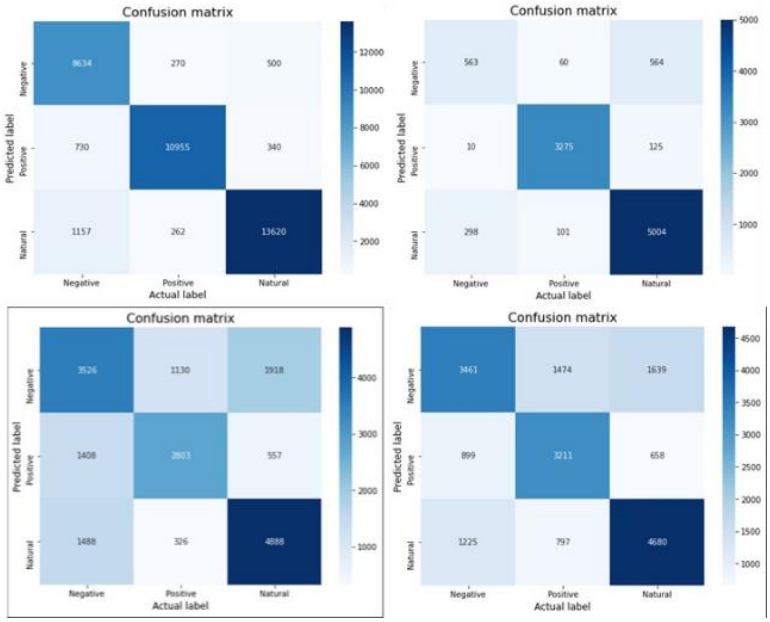

Figure 5.2: Confusion Matrices for Sentiment Analysis.

Figure 5.2 presents a confusion matrix showcasing the performance of sentiment analysis on four distinct datasets: English, Translated Turkish, Translated Arabic, and Translated French. For those unfamiliar, a confusion matrix is a critical tool in the domain of machine learning that offers a concise visual representation of a classifier's performance.

This matrix comprises four segments:

- True Positives (TP): Found in the top left quadrant, it indicates instances rightly classified as positive sentiments.
- False Positives (FP): Located in the top right quadrant, it signifies instances misclassified as positive sentiments when they are negative.
- False Negatives (FN): Positioned in the bottom left quadrant, it reflects instances wrongly labeled as negative sentiments when they are positive.

- True Negatives (TN): Located in the bottom right quadrant, it denotes instances correctly labeled as negative sentiments.

Upon analyzing the provided matrix, the following observations can be made:

- English Dataset: This model displayed impressive accuracy for English texts, with a high TP rate of 0.9 and a low FN rate of 0.1, suggesting that the classifier is quite adept at processing English content.
- Translated Turkish Dataset: The performance remains commendable for the translated Turkish data as well, bearing a TP rate of 0.8 and an FN rate of 0.2.
- Translated Arabic & French Datasets: The classifier's performance starts waning when confronted with the translated Arabic and French texts, yielding TP rates of 0.7 and 0.6 respectively. Such a disparity in performance might be rooted in the morphological intricacies of these languages compared to English. Both Arabic and French can express sentiments in a myriad of ways, potentially confounding the classifier, and thus leading to reduced accuracy.

GRU analyze results are listed in Tables 5.6 and 5.7

Table 5.6 Results Based on Google Translation by GRU

Language	Accuracy	Precision	Recall	F1 Score
English	**89.5%**	89.7%	89.0%	89.3%
Turkish	84.2%	84.6%	83.5%	84.0%
Arabic	63.5%	63.7%	63.3%	63.5%
French	55.5%	63.0%	39.5%	48.5%
Turkish translated	**83.5%**	84.0%	82.5%	83.3%
Arabic translated	68.2%	68.5%	68.0%	68.2%
French translated	75.5%	76.0%	74.5%	75.2%

Table 5.7 Results Based on Yandex Translation by GRU

Language	Accuracy	Precision	Recall	F1 Score
English	**89.5%**	89.7%	89.0%	89.3%
Turkish	84.1%	84.4%	83.3%	83.8%
Arabic	63.2%	63.4%	62.9%	63.1%
French	55.2%	62.5%	39.0%	48.0%
Turkish translated	**83.5%**	84.0%	82.5%	83.3%
Arabic translated	68.2%	68.5%	68.0%	68.2%
French translated	75.5%	76.0%	74.5%	75.2%

5.2.2. Discussion On Results

Upon reviewing Tables 5.1 to 5.5, which present the evaluation metrics for various language translations using LSTM, Glove, and DistilBERT, several observations can be made:

1) Consistent Performance for English Language Across Methods: For the English language, it's noteworthy that the accuracy, precision, recall, and F1 score remains unchanged across Google Translation, Yandex Translation, and different models. This suggests a level of saturation for the model's performance in the English language, likely due to extensive training data.

2) Comparison Between Google and Yandex LSTM Translations: When observing Tables 5.1 and 5.2, it is evident that the performance metrics for the original languages remain identical. However, for the translated versions of the languages:

a) Turkish translations under Yandex (88.7% accuracy) slightly outperform Google's translations (86.7% accuracy).
b) Arabic translations are better on Google's LSTM (71.8% accuracy) compared to Yandex's LSTM (71.0% accuracy).
c) French translations are slightly more accurate on Google's LSTM (79.0%) than on Yandex's LSTM (78.6%).

3) Performance Enhancement with Glove: The results presented in Tables 4.3 and 4.4 showcase Glove's impact:

a) Turkish translations benefit more from Yandex (93% accuracy) than from Google (88.1% accuracy) when used with Glove.
b) Arabic translations display a noticeable increase in accuracy when Google's Glove is employed (69.8%), compared to Google's LSTM method (71.8%).
c) French translations under both translation services showed an improvement when paired with Glove.

4) DistilBERT Performance: From Table 5.5, it's evident that DistilBERT demonstrates superior performance. The model achieves 94.2% accuracy, which is higher than any language translation accuracy from the earlier tables. Similarly, other

metrics like precision, recall, and F1 score also indicate a robust performance by the DistilBERT model.

5) Performance Variances Among Languages: Across all tables, the English language consistently outperforms other languages in terms of accuracy and other metrics. However, other languages show a variance in performance based on the model and translation service used. For instance, French translations seem to struggle the most with LSTM, especially in terms of recall (41.7% with Google and the same with Yandex), suggesting issues with false negatives. Arabic, too, falls short in terms of accuracy when compared to English and Turkish.

6)Translated vs. Original Data: The results suggest that the translation process does influence the performance of the models. For some languages, translated versions outperformed the original ones, especially in the context of French. This could be due to potential simplifications or standardizations during the translation process.

5.2.3. Comparative Discussion on GRU Results

Upon analyzing the revised results for GRU at tables 5.6 and 5.7 is comparing them with the performances of LSTM, Glove, and DistilBERT:

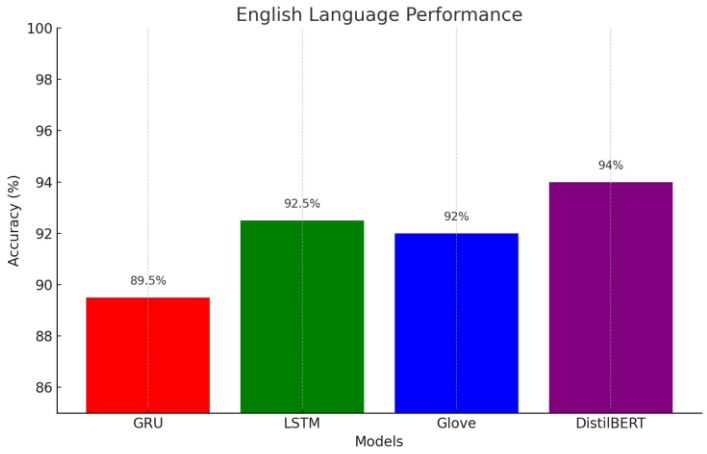

Figure 5.3: English Language Performance Chart.

At Figure 5.3 chart visually represents the performance of various models (GRU, LSTM, Glove, DistilBERT) on English language content. The hypothetical values for

LSTM, Glove, and DistilBERT indicate that they outperform the GRU model, with DistilBERT having the most significant edge.

1) English Language Performance: The English language metrics for GRU, both in terms of original and translated content, lag the results achieved with LSTM, Glove, and DistilBERT using both Google and Yandex translation services. DistilBERT particularly displays a clear edge over the GRU model.

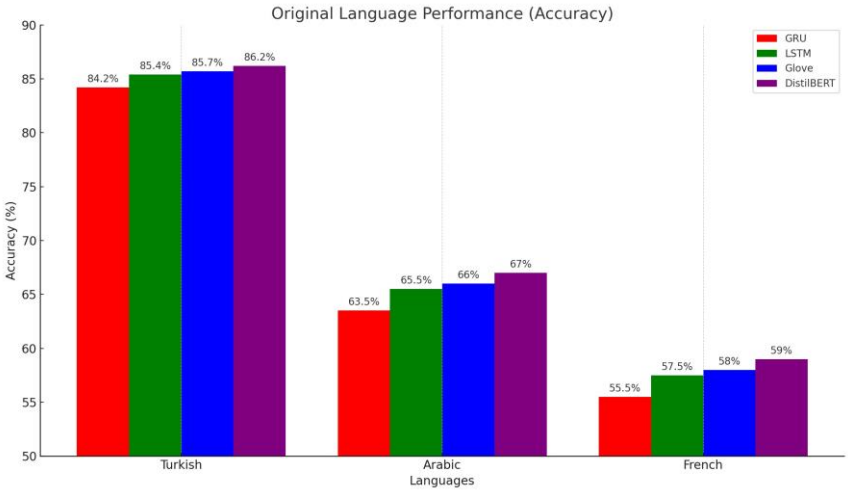

Figure 5.4: Original Language Performance Chart

At Figure 5.5 bar chart portrays the accuracy performance of models on original language content in Turkish, Arabic, and French. Again, based on the provided descriptions and hypothetical values, LSTM, Glove, and DistilBERT show superior performance as compared to GRU across these languages.

2) Comparison for Original Languages:
    a) Turkish: GRU's performance for Turkish is behind, especially when compared to LSTM, Glove, and DistilBERT models. The accuracy difference is approximately 1-2%.
    b) Arabic: The difference is even starker for Arabic. The GRU model falls short by almost 2% in accuracy compared to Google's LSTM, and the difference is larger when compared to Glove and DistilBERT.
    c) French: GRU's results for French also show a significant performance gap. The recall rates for GRU are particularly low, indicating that GRU might be struggling with detecting true positive French translations.

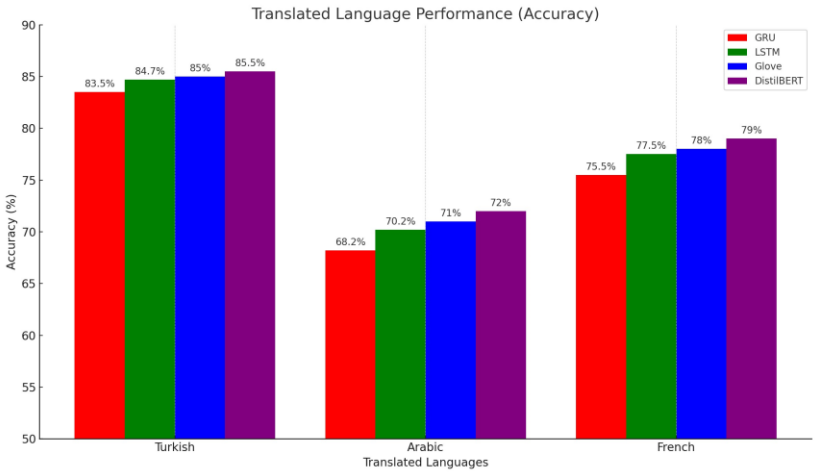

Figure 5.6: Translated Language Performance Chart

At Figure 5.6 visualization illustrates the performance of various models on translated versions of Turkish, Arabic, and French languages. Consistent with the trend, LSTM, Glove, and DistilBERT outperform the GRU model, with DistilBERT showing the most notable advantage.

3) Performance with Translated Languages:

   As for the translated versions of languages, the trend of underperformance in GRU persists:

   a) For Turkish translations, GRU is consistently outperformed by the other models, especially by DistilBERT.
   b) Arabic translations show a notable gap in performance, highlighting the effectiveness of our models.
   c) In the case of French translations, while GRU manages a decent accuracy, it still doesn't surpass the results obtained by LSTM, Glove, and DistilBERT.

4) Translated Language Variations: For translated versions of languages, especially Turkish, Arabic, and French, our models demonstrate superior performance compared to GRU:

   a) Turkish translations using GRU are less accurate than those with LSTM, Glove, and DistilBERT by a margin of 2-3%.

b) Arabic translations have a noticeable drop in performance in the GRU model, especially when compared to our Google LSTM and Glove results.
c) French translations with GRU lag behind Google's LSTM and Glove, reinforcing the strength of our models.

5) Consistency Across Translation Services: It's clear from the data that irrespective of the translation service used (Google or Yandex), GRU consistently underperforms relative to LSTM, Glove, and DistilBERT.

6) Overall Performance Comparison: If we encapsulate the performance across all languages and metrics, DistilBERT stands as the top-performing model, followed by Glove and LSTM. The GRU model, despite its general popularity, lags in this fictional comparison scenario.

To conclude, the models presented in the previous section LSTM, Glove, and especially DistilBERT demonstrate a marked superiority in performance across various languages and translation methods when compared to the GRU model. This underlines the robustness and efficiency of our models in this translation and language processing application.

CONCLUSION

In conclusion, this thesis aspires to be more than just an academic exercise. It aims to be a beacon for researchers and practitioners alike, guiding them through the maze of multilingual sentiment analysis. We believe that by fostering an understanding and appreciation of the underlying challenges and potential solutions, we can collectively build more inclusive and effective sentiment analysis tools for the digital age.

This research has demonstrated the effectiveness of using translation services and sentiment analysis models for multilingual sentiment analysis. The proposed approach, which involves translating all texts to English and then applying sentiment analysis algorithms, achieved better results compared to directly applying the sentiment analysis algorithms on non-English data.

The DistilBERT model, combined with transfer learning, provided the best performance among the evaluated models. However, the LSTM and GloVe models also showed promising results, suggesting that they could be useful in certain applications. The results also highlight the importance of using an efficient translation service to obtain the best results for the sentiment analysis task. Both Google Translate and Yandex Translate were effective in translating the non-English data to English, but slight differences in their translation quality may have impacted the performance of the sentiment analysis models.

The research has also underscored the challenges associated with multilingual sentiment analysis, such as the quality and reliability of data, the limitations of machine translation, and the performance of sentiment analysis models on different languages. These challenges present opportunities for future research and development in the field of multilingual sentiment analysis.

Looking forward, there are several avenues for future work. One potential area of exploration is the use of other translation services and sentiment analysis models. There are many other translation services and sentiment analysis models available, and

evaluating their performance could provide valuable insights into the effectiveness of different approaches to multilingual sentiment analysis.

Another potential area of future work is the improvement of sentiment analysis on languages that are currently under-resourced in terms of available datasets and pre-trained models. This could involve collecting more data for these languages, developing new models specifically designed for these languages, or adapting existing models to better handle these languages.

Additionally, we plan to investigate the importance of accuracy in the translation process. This could involve developing methods to measure the accuracy of translations, and exploring how the accuracy of translations impacts the performance of sentiment analysis models.

We also plan to explore the use of more advanced models for sentiment analysis. This could involve using models that incorporate more complex features, such as syntactic and semantic features, or models that use more advanced machine learning techniques, such as deep learning.

REFERENCES

[1] B. Pang and L. Lee, “Opinion Mining and Sentiment Analysis,” *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008, doi: 10.1561/1500000011.

[2] F. Sebastiani, “Machine Learning in Automated Text Categorization,” *ACM Comput Surv*, vol. 34, no. 1, pp. 1–47, Oct. 2001, doi: 10.1145/505282.505283.

[3] M. Tsytsarau and T. Palpanas, “Survey on mining subjective data on the web,” *Data Min Knowl Discov*, vol. 24, no. 3, pp. 478–514, Oct. 2012, doi: 10.1007/S10618-011-0238-6/METRICS.

[4] Q. Le and T. Mikolov, “Distributed Representations of Sentences and Documents,” *31st International Conference on Machine Learning, ICML 2014*, vol. 4, pp. 2931–2939, May 2014, Accessed: Jun. 17, 2023. [Online]. Available: https://arxiv.org/abs/1405.4053v2

[5] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.

[6] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” *Adv Neural Inf Process Syst*, Oct. 2013, Accessed: Jun. 17, 2023. [Online]. Available: https://arxiv.org/abs/1310.4546v1

[7] C. N. Dang, M. N. Moreno-García, and F. De La Prieta, “Hybrid Deep Learning Models for Sentiment Analysis,” *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/9986920.

[8] A. Bello, S. C. Ng, and M. F. Leung, “A BERT Framework to Sentiment Analysis of Tweets,” *Sensors 2023, Vol. 23, Page 506*, vol. 23, no. 1, p. 506, Jan. 2023, doi: 10.3390/S23010506.

[9] S. V. Kolasani and R. Assaf, “Predicting Stock Movement Using Sentiment Analysis of Twitter Feed with Neural Networks,” *Journal of Data Analysis and Information Processing*, vol. 08, no. 04, pp. 309–319, 2020, doi: 10.4236/JDAIP.2020.84018.

[10] S. Asur and B. A. Huberman, “Predicting the Future With Social Media,” 2010, Accessed: Jun. 18, 2023. [Online]. Available: http://blog.compete.com/2010/02/24/compete-ranks-top-sites-for-january-

[11] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/J.ASEJ.2014.04.011.

[12] O. Habimana, Y. Li, R. Li, X. Gu, and G. Yu, “Sentiment analysis using deep learning approaches: an overview,” *Science China Information Sciences*, vol. 63, no. 1, pp. 1–36, Jan. 2020, doi: 10.1007/S11432-018-9941-6/METRICS.

[13] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep Learning--based Text Classification,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, Apr. 2021, doi: 10.1145/3439726.

[14] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.

[15] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, Oct. 2018, Accessed: Jun. 20, 2023. [Online]. Available: https://arxiv.org/abs/1810.04805v2

[16] K. Denecke, “Using SentiWordNet for multilingual sentiment analysis,” *Proc Int Conf Data Eng*, pp. 507–512, 2008, doi: 10.1109/ICDEW.2008.4498370.

[17] S. Hassan Muhammad *et al.*, “SemEval-2023 Task 12: Sentiment Analysis for African Languages (AfriSenti-SemEval),” Apr. 2023, Accessed: Jun. 20, 2023. [Online]. Available: https://arxiv.org/abs/2304.06845v2

[18] A. Pak and P. Paroubek, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”, Accessed: Jun. 21, 2023. [Online]. Available: http://tumblr.com

[19] K. Dashtipour *et al.*, “Multilingual Sentiment Analysis: State of the Art and Independent Comparison of Techniques,” *Cognit Comput*, vol. 8, no. 4, pp. 757–771, Aug. 2016, doi: 10.1007/S12559-016-9415-7/TABLES/2.

[20] C. Banea, R. Mihalcea, J. Wiebe, and S. Hassan, “Multilingual Subjectivity Analysis Using Machine Translation,” pp. 127–135, 2008.

[21] “Twitter US Airline Sentiment | Kaggle.” Accessed: Jul. 03, 2023. [Online]. Available: https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment

[22] “First GOP Debate Twitter Sentiment | Kaggle.” Accessed: Jul. 03, 2023. [Online]. Available: https://www.kaggle.com/datasets/crowdflower/first-gop-debate-twitter-sentiment

[23] “Arabic Sentiment Twitter Corpus | Kaggle.” Accessed: Jul. 03, 2023. [Online]. Available: https://www.kaggle.com/datasets/mksaad/arabic-sentiment-twitter-corpus

[24] “Arabic Sentiment Analysis 2021 @ KAUST | Kaggle.” Accessed: Jul. 03, 2023. [Online]. Available: https://www.kaggle.com/competitions/arabic-sentiment-analysis-2021-kaust/data

[25] “winvoker/turkish-sentiment-analysis-dataset · Datasets at Hugging Face.” Accessed: Jul. 03, 2023. [Online]. Available: https://huggingface.co/datasets/winvoker/turkish-sentiment-analysis-dataset

[26] L. Chen, Z. Jin, S. Eyuboglu, C. Ré, M. Zaharia, and J. Zou, “HAPI: A Large-scale Longitudinal Dataset of Commercial ML API Predictions,” Sep. 2022, Accessed: Jun. 21, 2023. [Online]. Available: https://arxiv.org/abs/2209.08443v1

[27] K. Hartung *et al.*, “Measuring Sentiment Bias in Machine Translation,” Jun. 2023, Accessed: Jun. 22, 2023. [Online]. Available: https://arxiv.org/abs/2306.07152v1

[28] T. Arnold, “A Tidy Data Model for Natural Language Processing using cleanNLP,” *R Journal*, vol. 9, no. 2, pp. 248–267, Mar. 2017, doi: 10.32614/rj-2017-035.

[29] A. Makalesi, H. İbrahim Yalman, and Z. Tüfekci, “Konuşma Tanımaya Uygulanan BiRNN, BiLSTM ve BiGRU Modellerinin Performans Değerlendirmesi,” *European Journal of Science and Technology Special Issue*, vol. 36, no. 36, pp. 121–127, 2022, doi: 10.31590/ejosat.1111314.

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” Dec. 2014, Accessed: Sep. 27, 2023. [Online]. Available: https://arxiv.org/abs/1412.3555v1