Arab American University

Faculty of Graduate Studies

On the Combination of NLP and Extrinsic Semantic Resources for Developing an Arabic-English Sentiment Analyzer

By

Enas Mesqali

Supervisor

Dr. Mohammed A. M. Maree

Co- Supervisor

Dr. Mujahed Eleyat

This thesis was submitted in partial fulfillment of the requirements for the Master`s degree in

Computer Science

June / 2024

I

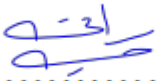
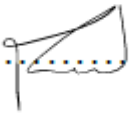
Thesis Approval

On the Combination of NLP and Extrinsic Semantic Resources for Developing an Arabic-English Sentiment Analyzer

By

Enas Mesqali

This thesis was defended successfully on 23/06/2024 and approved by:

Committee members	Signature
1. Dr. Mohammed A. M. Maree: Supervisor	
2. Dr. Mujahed Eleyat: Co- Supervisor	
3. Dr. Rami Hodrob: Internal Examiner	
4. Prof. Labib Arafeh: External Examiner	

Declaration

This is to declare that the thesis entitled “On the Combination of NLP and Extrinsic Semantic Resources for Developing an Arabic-English Sentiment Analyzer” under the supervision of Dr. Mohammed A. M. Maree and Dr. Mujahed Eleyat is my own work and does not contain any unacknowledged work or material previously published or written by another person, except where due reference is made in the text of the document.

Student Name: Enas Mesqali

Student ID:201812750

Signature: Enas Mesqali

Date: 09/11/2024

Acknowledgment

To our noble martyrs... Those who watered the soil of Palestine with their blood, you remain forever in our hearts and memories.

To our brave prisoners... Whose freedom was taken for the sake of liberating our homeland, Palestine. In your patience and steadfastness, we find inspiration to hold onto our just cause.

To my esteemed supervisors, Dr. Mohammad Maree and Dr. Mujahid Eleyat... It has been a great honor to learn from you and benefit from your valuable guidance throughout my research journey.

To my beloved father, may his soul rest in peace... My role model and guide, your memory will forever stay in my heart and accompany me in every step I take.

To my dear mother... My source of inspiration, my strength, and my unwavering supporter throughout the journey.

To my beloved brothers and sisters... You were the fuel to my spirit, sharing every moment of joy and challenge with me.

To my dear husband... My life partner, who shared with me the fruits of success on this journey.

To My daughter, the apple of my eye... who has all my heart, and all my love

To my family... Who have always stood by my side with their limitless support.

To my colleagues... Who shared with me the moments of research and study.

To those who loved me and were as warm as the sun during the hardest times... This thesis is dedicated to you. Thank you for your priceless love and support.

Abstract

The recent advancements in Natural Language Processing (NLP) technologies have significantly enhanced the capabilities of processing, analyzing, and understanding sentiments expressed in user-generated reviews across various products and services. This surge of interest in sentiment analysis has spurred considerable research efforts. In this study, we explore sentiment analysis with a specific focus on Arabic language. Leveraging both traditional pre-processing techniques and machine learning algorithms, we propose a comprehensive sentiment analysis model consisting of four stages.

The primary objective of our model is to harness English language resources and techniques to gauge their impact on classifier accuracy when applied to Arabic sentences. Through a series of experiments conducted on Arabic datasets and their English translations, we assess the effectiveness of various pre-processing methods and machine learning classifiers: Logistic Regression (LR), Random Forest (RF), Naïve Bayes (NB), and Support Vector Machines (SVM). Notably, SVM classifier consistently outperformed others, exhibiting the highest accuracy across most scenarios especially when combining Lemmatization and Stemming.

Furthermore, we explore the influence of translating datasets and incorporating synonyms on sentiment analysis accuracy. While the translation of datasets from Arabic to English and vice versa did not yield significant changes in accuracy, the inclusion of synonyms from English datasets in Arabic sentiment analysis experiments produced mixed results. This underscores the intricacies of language-specific nuances and the challenges in effectively capturing sentiment across different languages.

When comparing our study with previous research that used the ASTD dataset, several key differences and similarities emerge. Previous studies explored a range of classifiers, including SVM, NB, LR, CNN, and RNTN, with accuracy results varying between 85% and 90% for traditional features like n-grams, TF-IDF, and word embeddings like Word2Vec. However, the RNTN algorithm showed a lower accuracy rate of 58.5%, and the SVM algorithm achieved 51.7%. Other research focused on deep learning models like CNN and LSTM, which yielded accuracy rates of 64.3% and 64.75%, respectively. In contrast, our study highlighted the importance of specific pre-processing techniques, demonstrating that methods such as lemmatization and stemming could significantly enhance the performance of machine learning classifiers like SVM, achieving accuracy results of up to 80%.

Overall, our study showcases the evolving landscape of sentiment analysis research, highlighting the adaptability of techniques to address language-specific challenges and nuances. These findings contribute to the broader understanding of sentiment analysis methodologies and underscore the importance of considering linguistic differences in sentiment analysis tasks. Finally, recommendations for future research include expanding the Arabic dataset and exploring advanced deep learning models to capture more complex patterns. Additionally, refining linguistic tools specific to Arabic could further enhance sentiment analysis accuracy. These steps aim to better address the intricacies of language-specific challenges and contribute to more effective sentiment analysis methodologies.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AJGT: Arabic Jordanian General Tweets dataset

ANN: Artificial Neural Networks

Acc: Accuracy

API: Application Programming Interfaces

ArSenTD: Arabic Sentiment Twitter Dataset

ASTD: Arabic Sentiment Twitter Dataset

AWN: Arabic WordNet

BNB: Bernoulli Naive Bayes

CBOW: Continuous Bag of Words

CNN: Convolutional Neural Network

DA: Dialects Arabic

DBOW: Distributed Bag of Words

DT: Decision Tree

DM: Distributed Memory

FN: False Negative

FP: False Positive

GloVe: Global Vectors for word representations

ID3: Iterative Dichotomiser 3

IG: Information Gain

IMDB: Internet Movie Database

KNN: k-Nearest Neighbors

LABR: Large-scale Arabic Book Reviews dataset

LR: Logistic Regression

LSTM: Long Short-Term Memory

ML: Machine Learning

MLP: Multilayer Perceptron

mRMR: minimum Redundancy Maximum Relevance

MSA: Modern Standard Arabic

NB : Naïve Bayes

NEUNET/NN: Neural Network

NLP: Natural Language Processing

NLTK: Natural Language Toolkit

POS :Part-Of-Speech

RBF SVM: Radial Basis Function SVM

RF: Random Forest

RS: Rough Set

RLU: Rectified Linear Unit

RNTN: Recursive Neural Tensor Network

SA: Sentiment Analysis

SG: Skip-gram

SO: Semantic Orientation

SVMs: Support Vector Machines

TF: Term Frequency

TF-IDF: Term Frequency Inverse Document Frequency

TN: True Negative

TP: True Positive

VSMs: Vector Space Models

CHAPTER 1:
INTRODUCTION

Chapter 1 - Introduction

1.1.Background

The burgeoning volume of social media data has propelled sentiment analysis to the forefront of research, garnering considerable attention for its role in discerning emotions conveyed through natural language text. This attention is pivotal for aiding decision-making processes in product development, service provision, and comprehending consumer perceptions across various industries. As businesses increasingly rely on social media insights, sentiment analysis has emerged as a crucial tool for extracting actionable information from user-generated content.

At its core, text sentiment analysis aims to identify and articulate sentiments expressed in textual content, discerning whether opinions are positive or negative towards products and services across diverse sectors. This analytical process has gained traction across multiple languages, with a particular focus on Arabic—a language still in its developmental stages in terms of sentiment analysis. This surge of interest is propelled by several factors, including the scarcity of available Arabic language semantic resources, the intricate nuances of Arabic vocabulary, and its varied forms of expression. Moreover, the presence of numerous Arabic dialects, in addition to the standard language, further complicates sentiment analysis efforts in Arabic texts.

Sentiment analysis (Abu Farha and Magdy 2021), as discussed in (Alrefai, Faris et al. 2018), encompasses four levels: document, sentence, aspect, and word levels. Within this framework, three primary approaches are employed for text sentiment analysis: lexicon-based, machine learning, and hybrid algorithms (Boudad, Faizi et al. 2017, Alrefai, Faris et al. 2018).

The lexicon-based approach to sentiment analysis involves utilizing lexicons that assign polarity values to words based on their sentiment orientation. Words with positive connotations are assigned values greater than zero, while those with negative connotations are assigned values less than zero. Words with neutral sentiments typically have values close to zero. In contrast, machine learning approaches involve training a model on a dataset, enabling the machine to learn patterns and make predictions based on the learned features. This method allows the machine to discern sentiment based on contextual cues and linguistic patterns present in the data.

Machine learning, as defined by Arthur Samuel, gives computers the ability to learn without explicit programming. IBM[1] defines machine learning as a branch of artificial intelligence and computer science that focuses on using data and algorithms to imitate human learning, gradually improving accuracy. Supervised and unsupervised techniques are employed in machine learning. Supervised learning involves pre-defined and labeled input and output data for constructing a model capable of sentiment analysis and predictions on new texts. Unsupervised learning operates on unlabeled data, uncovering patterns by analyzing words and their polarity, distinguishing them as positive, negative, or neutral. Supervised techniques include decision trees, linear, probabilistic, and rule-based classifiers, along with deep learning techniques (Boudad, Faizi et al. 2017, Alrefai, Faris et al. 2018), while K-Means is an example of an unsupervised technique.

The hybrid approach combines the strengths of both lexicon-based and machine learning approaches, extracting features from a lexicon or a collection of words. Machine learning

[1] "What is Machine Learning?". IBM. Retrieved 2023-12-19.

classifiers are then applied to leverage these features, proving to enhance accuracy, as seen in previous studies (Alrefai, Faris et al. 2018, Maree, Eleyat et al. 2023).

The methodology for text sentiment analysis models involves collecting datasets, pre-processing the datasets, identifying features, applying sentiment analysis techniques based on machine learning algorithms or lexicons, or a combination of both, and evaluating the results (El-Masri, Berardinelli et al. 2017).

The primary objective of this research is to advance the field of Sentiment Analysis in the Arabic language. Given the unique linguistic and contextual challenges posed by Arabic, the research aims to develop and promote specialized techniques that cater to the intricacies of sentiment expression in Arabic texts. This includes addressing issues related to grammar, morphology, and the diversity of Arabic dialects, with the goal of enhancing the accuracy and reliability of sentiment analysis outcomes.

In addition to focusing on Arabic, the research seeks to create a Bilingual Sentiment Analysis model that can effectively analyze sentiments in both Arabic and English texts. This involves employing sophisticated NLP techniques to understand the nuances of language and sentiment expression in diverse linguistic settings.

To enrich the sentiment analysis process, the research aims to integrate NLP methods with multiple extrinsic semantic resources. By leveraging semantic information and linguistic insights, the objective is to develop a more comprehensive and context-aware sentiment analysis framework. This integration is expected to enhance the model's ability to accurately discern sentiments in varying linguistic and cultural contexts.

A notable aspect of the research involves investigating the re-use of existing English language-based sentiment analysis resources in the Arabic NLP-based pipeline. This exploration seeks to determine the feasibility and effectiveness of incorporating resources developed for English sentiment analysis into the Arabic language framework. The goal is to assess whether such cross-linguistic resource utilization contributes positively to the accuracy and efficiency of sentiment analysis in Arabic.

The final objective revolves around impact assessment, where the research aims to quantitatively evaluate the effects of integrating English resources on sentiment analysis outcomes in Arabic. Through rigorous evaluation measures, the research seeks to provide insights into the benefits and potential challenges associated with the cross-linguistic application of sentiment analysis resources. Ultimately, the research aims to contribute valuable knowledge to the broader field of natural language processing and sentiment analysis, with implications for both Arabic and bilingual sentiment analysis applications.

1.2.Problem Statement and Research Questions

Sentiment Analysis encounters numerous challenges when applied to natural language texts, particularly in the context of Arabic. These challenges stem from the grammatical and morphological intricacies, diverse writing styles, and linguistic characteristics of Arabic. Some of the key challenges include:

- Grammatical and Morphological Complexity (Hamdi, Shaban et al. 2016): Arabic has complex grammatical and morphological rules that affect the formation and structure of words. This complexity poses challenges in accurately identifying and analyzing sentiment-bearing elements.

- Synonyms and Polysemy: Arabic, like any language, exhibits synonyms and polysemy (multiple meanings for a single word). Identifying the intended meaning of a word in a given context becomes challenging, impacting the precision of sentiment analysis.
- Semantic Heterogeneity: Sentiment analysis faces difficulties in handling semantic heterogeneity in Arabic texts, where words may have different meanings based on the context in which they are used.
- Diversity of Dialects (Hamdi, Shaban et al. 2016): The Arabic language encompasses various dialects, and the diversity in the way people express sentiments across these dialects can affect the analysis of textual content.
- Lack of Arabic Language Processing Tools: The scarcity of dedicated software and tools for processing Arabic texts hinders the development of effective sentiment analysis models. This includes tools for tokenization, stemming, and part-of-speech tagging tailored to the Arabic language.
- Structural Complexity of Arabic Words: Arabic words exhibit complexity in terms of derivation, morphology, and other linguistic features. Analyzing sentiments becomes challenging due to the intricate structure of words.
- Ambiguity in the Absence of Diacritics: Omitting diacritics, which are marks indicating vowel sounds and other linguistic features, can introduce ambiguity to the text. This ambiguity complicates sentiment determination and polarity assignment.
- Writing Style Variations: Differences in the way words are written, such as the addition or repetition of letters for emphasis, can impact the interpretation of sentiment, making it challenging to achieve high efficiency and accuracy.

- Limited Studies in Arabic (Nassif, Elnagar et al. 2020): The availability of studies and resources for sentiment analysis in Arabic is relatively limited compared to languages like English. This scarcity affects the development and benchmarking of sentiment analysis models in Arabic.

Addressing these challenges requires the development of specialized tools, resources, and methodologies tailored to the unique linguistic characteristics of Arabic, fostering more accurate and efficient sentiment analysis in Arabic texts.

In this research, we focus on the success of building sentiment analysis systems for the Arabic language. We attempt to address the following research questions:

1. How accurate is sentiment analysis in texts in both Arabic and English?
2. What factors affect the performance of sentiment analysis when used in Arabic texts?
3. What are the machine learning techniques used for Arabic Language Sentiment Analysis?
4. How efficient is the reuse of existing English-based resources in processing Arabic natural languages?

The research endeavors to assess the accuracy of sentiment analysis, particularly in the context of Arabic language processing. There are two hypotheses:

- The first hypothesis posits that the utilization of multiple manually grouped datasets in Arabic will have a discernible impact on the quality of existing sentiment prediction techniques. By systematically analyzing and categorizing diverse datasets, the research aims to uncover patterns and correlations that contribute to the refinement of sentiment analysis models for the Arabic language.

- The second hypothesis builds on the expansion of techniques for the determination of polarity, suggesting that such process will significantly influence the overall quality of the improved sentiment analyzer. This hypothesis acknowledges the need for advanced techniques to handle the unique linguistic features of Arabic, including grammar, morphology, and dialectal variations. The research seeks to innovate and enhance existing sentiment analysis methods by incorporating sophisticated approaches that go beyond conventional language processing.

These hypotheses collectively guide the research in its pursuit of advancing sentiment analysis in Arabic and contribute valuable insights into the nuanced challenges posed by this language. Through systematic experimentation and analysis, the research aims to validate or refine these hypotheses, ultimately contributing to the development of more accurate and effective sentiment analysis tools tailored for Arabic texts.

1.3.Thesis Organization

The subsequent chapters of this thesis are organized as follows. In Chapter 2, we provide an extensive review of the literature, focusing on existing research in sentiment analysis for both Arabic and English languages. We delve into the techniques employed in these studies and highlight key findings and methodologies.

In Chapter 3, we present the theoretical framework of our proposed methodology. This includes a detailed discussion of the pre-processing steps involved in preparing the data for sentiment analysis, as well as an overview of the machine learning classifiers utilized in our approach.

Moving on to Chapter 4, we outline the experimental setup employed in our study. This encompasses a comprehensive overview of the datasets used, as well as specific procedures for evaluating the performance of our sentiment analysis framework. We discuss the experimental results obtained and analyze their implications, particularly focusing on the impact of pre-processing techniques on accuracy for both Arabic and English languages.

Finally, in Chapter 5, we present our conclusions and discuss future directions for research. We highlight potential extensions of our proposed work and identify areas for further exploration and improvement in the field of sentiment analysis.

CHAPTER 2:

Literature Review

Chapter 2 - Literature Review

2.1 Background

Sentiment analysis, also known as opinion mining, is a natural language processing (NLP) technique used to determine the sentiment expressed in a piece of text. It involves analyzing and interpreting subjective information in order to identify the sentiment, opinions, emotions, or attitudes of a speaker or writer towards a particular topic or the overall context. Researchers have recently displayed keen interest in sentiment analysis as a prominent area of study.

Examining sentiments and grasping the contextual nuances of textual content present formidable challenges, particularly when sentiments manifest across various languages, including English and Arabic. In the following sections of this chapter, we delve into the principal research methodologies employed to analyze sentiment in these languages.

2.2 Sentiment Analysis for English Language Texts

Many research endeavors have focused on unraveling the intricacies of sentiment analysis in English text. These investigations delve into the methods used for preprocessing textual data and the various techniques applied to English datasets. The ultimate objective is to empower decision-makers with valuable insights into products or services by deciphering the sentiments conveyed in the analyzed text.

Overall, sentiment analysis for English language texts is a rich and evolving field, with a wide array of approaches and techniques that continue to be refined and adapted to address the complexities of language and sentiment expression.

In (Başarslan and Kayaalp 2021), the researchers employed various classifiers, including machine learning algorithms like NB, SVM, and ANN. These algorithms were applied to two

distinct datasets: the first comprising 4,500 tweets gathered through an API, and the second consisting of movie reviews from IMDB, aggregated via the Kotzias method.

Similarly, in (A. Al Shamsi, Bayari et al. 2021), researchers utilized a range of machine learning algorithms, including NB, ID3, K-NN, DT, RF, and Random Tree. This study focused on balanced and unbalanced datasets, incorporating over 14,000 Kaggle tweets related to six US airlines (United, Delta, Southwest, Virgin America, US Airways, and American).

In (Ali, Hamid et al. 2019), a different approach was taken as researchers applied deep learning algorithms, specifically MLP, LSTM, and CNN. The target was to analyze emotions within the IMDB dataset. Additionally, the authors proposed a hybrid model combining LSTM and CNN for enhanced performance.

In (A. Al Shamsi, Bayari et al. 2021, Başarslan and Kayaalp 2021), text preprocessing played a pivotal role. This involved several steps such as punctuation removal, stop word elimination, word root derivation, converting uppercase to lowercase, tokenization, and feature extraction using TF-IDF and Word2vec. TF-IDF focused on discerning significant words by assigning weights based on their frequency and importance, while Word2vec aimed to represent words as vectors, with similar words having proximity in coordinates. Two distinct representations, CBOW and SG, were explored. CBOW represented the target word based on neighboring words, while SG represented adjacent words based on the target word.

Furthermore, in (A. Al Shamsi, Bayari et al. 2021), dataset separation was conducted, allocating 66% for training and 34% for testing purposes. These comprehensive methodologies showcase the diversity of techniques applied in sentiment analysis, encompassing both

traditional machine learning and advanced deep learning algorithms, coupled with meticulous text preprocessing strategies.

In (Ali, Hamid et al. 2019), the authors utilized a hybrid model combining LSTM and CNN model for sentiment classification. Text processing is restricted to 500 words, and word embedding using word2vec is employed. The CNN employs weights and biases to train multiple layers within each neuron during the training phase. Each neuron's output transformation is activated using RLU. The Maxpooling layer in this hybrid model selects word weights of the highest value, applying a Maxpooling mask sequentially to reduce the text length from 500 words to 250 words.

The evaluation in (Başarslan and Kayaalp 2021) involved the use of a confusion matrix for the IMDB dataset and Twitter datasets related to health data. The outcomes from the 4,500 Twitter sentiment dataset delineated that 1,220 tweets were categorized as positive, 1,600 as negative, and 1,680 as neutral. Experiments employing NB, SVM, and ANN techniques demonstrated that ANN surpassed others in both datasets when utilizing TF-IDF and word2vec features.

In (A. Al Shamsi, Bayari et al. 2021), experiments were conducted using various classifier on both balanced and unbalanced datasets related to tweets about six US airlines. The accuracy varied among different airlines, in the unbalanced dataset, classifiers like DT, RF, and NB performed inconsistently, with lower accuracy rates, particularly for smaller datasets such as Virgin America, where accuracy ranged between 31-39%, while K-NN and DT showed improved performance up to 82.72%. In contrast, classifiers on the balanced dataset showed more consistent results. For example, NB dramatically improved in performance, achieving over 97% accuracy for Virgin America, Southwest, and American Airlines in the balanced dataset, whereas it had much lower accuracy in the unbalanced case.

Overall, the balanced dataset generally produced higher and more consistent results across classifiers, indicating that addressing class imbalances can significantly improve model performance. This is particularly evident with NB and ID3, where the balanced dataset led to substantially higher accuracy.

In (Ali, Hamid et al. 2019), the experimental work applied to the IMDB dataset of 50,000 movie reviews using the proposed LSTM and CNN hybrid approach resulted in the highest accuracy of 89.20%. This surpassed the performance of both traditional machine learning techniques (SVM, NB, RNTN) and other deep learning models (MLP, CNN, LSTM). However, when applying the RNTN model to another dataset consisting of 11,855 English movie reviews, accuracy decreased to 80.70%. Additionally, SVM and NB models were employed on a dataset comprising 2,053 reviews, yielding accuracy rates of 82.90% and 81%, respectively.

These results underscore the efficacy of hybrid models, highlighting the influence of preprocessing techniques and word embedding on sentiment analysis outcomes. Moreover, they demonstrate the fluctuating performance of classifiers when applied to diverse datasets and analytical approaches.

2.3 Sentiment Analysis for Arabic Language Texts

Despite the growing interest among researchers in sentiment analysis, the study of sentiment in the Arabic language is still in its early stages. This is primarily due to the scarcity of resources available in Arabic compared to English. Arabic presents unique challenges for sentiment analysis, as it involves complex writing styles, including Standard Arabic used in formal contexts and dialectal Arabic used in daily communication. The language exhibits

variations in word forms, negation phrases, and the presence of stop words that can significantly impact sentiment classification.

Writing Arabic text presents several complexities, including variations in word forms and the presence of negation phrases. For instance, Arabic words can be written in multiple forms, such as with or without the Ta' marbootah (ة) at the end, as seen in examples like "المقدمة" and "المقدمه". Additionally, negation phrases containing stop words are often removed during text preprocessing, impacting sentiment classification. Removing these stop words can alter the sentiment of a sentence, potentially changing it from negative to positive. Furthermore, certain phrases convey negative emotions without explicit negation words, like "حسبي الله ونعم الوكيل". Verb conjugation in Arabic also varies based on factors such as subject plurality, gender, and formality. For instance, "هي تحب المطر" (She loves rain) and "هو يحب المطر" (He loves rain) demonstrate this variation. Moreover, there are instances where nouns lacking inherent sentiment are written similarly to adjectives conveying emotions, as illustrated by the noun "جميلة" and the adjective "جميلة" (Alrefai, Faris et al. 2018).

Various methods have been employed by researchers to analyze sentiment in Arabic. In (Mohammad, Salameh et al. 2016), a lexicon-based approach was used, combining ancient and modern lexicons. The researchers compared the performance of each and explained the process of creating Arabic sentiment lexicons using remotely supervised techniques or machine translation from English via Google Translate. Experiments using the SVM classifier on different datasets achieved an accuracy of 63% using manual dictionaries.

In (Soliman, Eissa et al. 2017), researchers utilized the open-source technology AraVec to represent Arabic text words from sources like Twitter, Wikipedia, and web pages. Text preprocessing for Arabic text involves several steps to enhance the quality of the data. Initially,

non-Arabic text, particularly from languages sharing letters with Arabic like Persian and Urdu, is identified and removed. Subsequently, normalization is applied, which includes removing diacritical marks and standardizing certain letters. For instance, the letters "أ،آ،إ" are replaced with "ا", while "ة" becomes "ه" and "ى" becomes "ي". Moreover, repeated letters within words are reduced to a single instance, such as converting "سلاااااام" to "سلام". Additionally, emojis and URLs are replaced with textual representations for clarity and consistency. These preprocessing steps ensure that the Arabic text is standardized and ready for further analysis, which included employing VSMs and word embedding techniques like Word2Vec for representation in continuous space.

In (Fouad, Mahany et al. 2020), the ArWordVec model was proposed for sentiment analysis, employing CBOW, SG, and GloVe methods. Evaluation was conducted on Twitter datasets, ASTD and AraSenti, utilizing a corpus of 55 million pre-processed tweets. Despite limited Arabic language resources, the ArWordVec model demonstrated promising results, particularly in word similarity tasks.

The construction of the ArWordVec model involved several steps. Initially, data collection encompassed 55 million tweets across diverse domains. Pre-processing steps included removing hashtags, symbols, non-Arabic letters, punctuation, stop words, repeated texts, diacritics, spaces, and normalizing letters. Handling duplicate and lengthy characters further refined the dataset for subsequent processing within the ArWordVec model.

Utilizing the word2vec toolkit, the text was classified into input and output word lists, facilitating the creation of a neural network model for supervised learning. CBOW analyzed the text to identify the target word from its context, while SG defined a set of context words for each target word.

Evaluation of the word embedding model involved word similarity tasks for English language text. However, due to the absence of resources for Arabic word similarity tasks, a seed group of words comprising positive and negative terms was utilized. This seed group served as a benchmark for evaluating the model's performance in retrieving similar words from the tweet corpus. Additionally, the seed word vectors were compared with the similar word matrix during classifier training, utilizing algorithms such as SVM or NB.

This methodology showcased the application of word embedding techniques to sentiment analysis in Arabic texts, even in the absence of extensive language resources. The use of seed words allowed for the evaluation of the model's performance in capturing sentiments within the given datasets.

A different approach was presented in (Barhoumi, Aloulou et al. 2018), where sentiment analysis in Arabic involved comparing sentiments in Arabic text with machine-translated English versions. The study employed two classifiers, LR, and MLP, using vector embeddings derived from documents. The embeddings consisted of two vectors: DM and DBOW. The LABR dataset was used for experimentation.

The evaluation of experiments involved assessing the error rates, with the baseline error rate for Arabic text recorded at 25.37%, while the translated English text exhibited a lower error rate of 23.70%. This improvement was attributed to the elimination of irrelevant or redundant words during translation. These words, often untranslated into English, included proper names, terms with duplicate letters, colloquial dialects, or non-Arabic characters. Examples included terms like "رفيو" (derived from "review") and "بروتكشن" (from "protection"). The presence of such words in the original text was deemed potentially confusing for polarity detection.

Upon repeating the experiment with the original Arabic text but excluding misleading words, the error rate increased to 26.86%, underscoring the significance of the omitted words in the Arabic text and their absence in the machine-translated version.

Further experiments on the original Arabic text involved implementing various enhancements. For instance, incorporating light stemming as a preprocessing technique resulted in an error rate of 23.31%, which closely approached the performance of machine-translated text and outperformed the original text alone.

These findings indicated that leveraging machine translation as a statistical tool or employing light stemming as a linguistic technique yielded comparable outcomes. This discovery suggested the potential for using machine translation as a viable alternative to stemming in the development of effective sentiment analysis systems.

In (Shoukry and Rafea 2012), researchers concentrated on sentence-level sentiment classification in Arabic text, employing ML and SO approaches. In the supervised ML approach, a dataset with positive and negative classifications was represented by a feature vector, and SVM was utilized as the ML technique. The researchers achieved promising results, with an accuracy of up to 90%. Some other studies also utilized SVM and NB technologies, with SVM exhibiting superior performance. In a different study, KNN was employed to analyze Twitter comments by considering features such as hashtags and smiley faces, and SVM was used with abstract features.

The researchers proposed machine learning techniques for sentiment classification in Arabic text at the sentence level, particularly in the Egyptian dialect. They conducted preprocessing after collecting Twitter data, incorporating stop words specific to the Egyptian dialect. The

classifiers used were NB and SVM, with features extracted from unigrams and bigrams to achieve the highest accuracy.

The sentiment analysis process for Arabic text involved collecting tweets through Twitter's API, cleaning the tweets of non-Arabic words, user names, images, and URLs. Each tweet was then categorized as positive or negative through sentiment evaluators. Features were extracted and represented as vectors consisting of words and their frequencies, with words extracted from unigrams and bigrams. The classifiers were trained using these vectors with the Weka Suite, and experiments were conducted using NB and SVM classifiers. Negative words were excluded during experiments, and the results showed that using only unigrams yielded better accuracy in the SVM classifier 0.721, while NB had an accuracy of 0.654. Removing stop words slightly improved accuracy for both SVM and NB. However, combining unigrams and bigrams did not lead to improved accuracy, attributed to the limited number of iterations for bigrams.

Several challenges were encountered during the training of the data, impacting classification accuracy. These challenges included the omission of negation words, increased word frequency when sentences were repeated, the presence of sentences expressing insincere opinions, and statements with multiple opinions that conveyed vague sentiments. Addressing these challenges is crucial for enhancing the accuracy of sentiment classification in Arabic text.

2.4 Sentiment Analysis for Bilingual (English - Arabic)

Indeed, sentiment analysis research has predominantly focused on English-language texts due to the abundance of resources and datasets available in English. This poses a challenge for languages like Arabic, where resources are comparatively scarce, despite a significant online presence of Arabic

speakers. To overcome this challenge, researchers often resort to knowledge transfer from resource-rich languages, such as English, to languages with limited resources like Arabic.

In (El-Awady, Barakat et al. 2015), the authors addressed the challenge of sentiment analysis for resource-scarce languages like Arabic by creating the Senti-Word lexicon for Arabic vocabulary. They employed machine learning algorithms, including DT, NB, and SVM, to classify sentiments in both Arabic and English texts. The experiments involved datasets from various categories such as Movies, DVD, Books, and Electronics, collected from Amazon, each consisting of 1000 instances with an equal number of positive and negative samples.

The study explored several sentiment analysis features selected through various methods, including IG, Unigram, RS, mRMR, and a Hybrid approach. IG focused on determining data repetition, word importance percentages, and arranging features in descending order. The RS method, when used with IG, aimed to identify optimal features with reduced time and effort by eliminating redundant and unnecessary data. The researchers employed a confusion matrix to evaluate the performance of the classifiers.

For the English language dataset, experiments using the four specific features, especially those identified by IG and mRMR, showed increased accuracy. Combining RS with IG or mRMR further improved accuracy by 1.5% and 4.2%, respectively.

In the Arabic language experiments, researchers utilized a lexicon containing words along with their respective polarities. This lexicon was then applied to a dataset sourced from YouTube, which comprised 214 Arabic films. These films encompassed both native Arabic texts and translations, encompassing texts originating from Arabic as well as those translated from other languages. Approximately 25% of the dataset consisted of translations into Arabic, while 3% were translations from French. To prepare the dataset for analysis, preprocessing steps were undertaken. These steps involved removing redundant and non-Arabic texts, eliminating stop words, tokenizing the text, and applying stemming using the Arabic Stemmer Khoja.

Two classification experiments were conducted: one without feature selection and another with feature selection using IG. In the first experiment, SVM and NB achieved accuracy rates of 83.96% and 89.34%, respectively. In the second experiment with IG feature selection, NB outperformed other technologies with an accuracy rate of 91%.

Additionally, experiments involving the Senti-Word Lexicon and various preprocessing factors showed varying accuracy rates. The best accuracy of 95.9% was achieved by combining normalization, removing stop words, and using weights. The study revealed that the NB technology performed exceptionally well, outshining other technologies and achieving an overall accuracy rate of 94.5%. The researchers compared their results with other studies, showcasing the effectiveness of their proposed approach in sentiment analysis for Arabic texts using machine learning techniques and the Senti-Word Lexicon.

In their study (Abo, Shah et al. 2018), researchers conducted sentiment analysis using the NB and DT algorithms on three distinct datasets. The first dataset, gathered from the Facebook developer API, comprised 658 comments discussing a football match in English. The second dataset involved reviews of books in modern Arabic, totaling 63,000 reviews collected from Goodreads within a month, with 2,648 reviews selected for analysis. The third dataset, consisting of 409 customer reviews in Arabic dialects, was manually collected from JEERAN. The researchers pre-processed the English dataset, utilized RapidMiner software for sentiment analysis, and categorized the text into positive and negative sentiments.

The sentiment analysis process was replicated for two datasets representing MSA and DA. The researchers employed NB and DT algorithms, evaluating their performance in terms of accuracy and runtime. The results of classification experiments indicated that the DT algorithm achieved the highest accuracy of 97% for the MSA dataset, while NB attained 89.50%. Conversely, the DA dataset exhibited lower accuracy, with both DT and NB reaching 54.4% and 50.8%, respectively. In the English language dataset, DT achieved an accuracy of 83.87%, and NB achieved 84.25%. Additionally, when assessing running time, the DT algorithm consumed more time for the MSA dataset compared to the DA dataset. These findings underscored the variability in sentiment analysis outcomes across different languages

and datasets, shedding light on the nuances of applying sentiment analysis algorithms to diverse linguistic contexts.

In (Almaghrabi and Chetty 2020), researchers employed a deep learning methodology to conduct sentiment analysis on both Arabic and English texts. They utilized the MLP model, a type of Neural Network (NN) that leverages word vectors for sentiment prediction. Additionally, the model considered font types commonly used in word processing, such as Times Roman or Times New Roman. The Arabic dataset comprised 1,524 movie reviews, while the English dataset included a larger corpus of 515,000 reviews collected from 1,493 hotels.

The experiments focused on evaluating the MLP model's performance in text prediction and assessing its accuracy using metrics such as Accuracy, Precision, Recall, and F1-Score for both datasets. The results revealed that the MLP model achieved an accuracy of 87% for the Arabic dataset and an impressive 96% for the English dataset. These findings indicate that the predictions generated by the MLP model demonstrated favorable outcomes compared to experiments conducted in other studies using the Word2Vec model.

In those experiments, the classification outcomes were negative for all reviews containing both negative and positive sentiments. Conversely, the Word2Vec model performed well on the English dataset. These results underscore the enhanced accuracy achieved on Arabic datasets through the application of the MLP model, showcasing its effectiveness in sentiment analysis across diverse languages.

2.5 Summary

Sentiment analysis in Arabic faces challenges due to the complex nature of the language. Researchers have explored lexicon-based methods, machine learning approaches, and word embedding techniques to overcome these challenges and enhance accuracy in sentiment classification. The limited resources for Arabic sentiment analysis underscore the need for further research in this area.

Several studies have attempted to bridge this gap by developing lexicons, employing machine learning algorithms, and applying deep learning models to sentiment analysis tasks in Arabic to achieving accurate sentiment analysis.

By leveraging knowledge from well-established sentiment analysis approaches in English and adapting them to the specific linguistic characteristics of Arabic, researchers aim to enhance sentiment analysis capabilities for languages with limited resources. These efforts contribute to the development of effective methodologies for understanding sentiments expressed in Arabic texts on the internet.

CHAPTER 3:

PROPOSED METHODOLOGY

AND

THEORETICAL FRAMEWORK

Chapter 3 - Proposed Methodology and Theoretical

3.1 Background

The sentiment analysis process encompasses various stages, each contributing to the overall accuracy and effectiveness of the analysis. These stages include data collection, pre-processing, feature extraction, sentiment classification, and evaluation, as illustrated in Figure 1.

Figure 1:The general outline of the stages of sentiment analysis

3.1.1 Data Collection:

This initial stage involves gathering the text data from various sources, such as social media platforms, customer reviews, or other text repositories. The quality and quantity of the collected data significantly impact the subsequent analysis.

There are two types of datasets in machine learning: balanced and unbalanced. Balanced datasets have an approximately equal number of instances for each class, making it easier for models to learn and perform well across all classes. Unbalanced datasets, on the other hand, have a significant disparity in the number of instances between classes, which is common in many real-world applications. In these cases, some outcomes are rare but crucial.

Unbalanced datasets can lead to models that are biased toward the majority class, often overlooking the minority class, which may be the class of greatest interest. To address

this issue, converting an unbalanced dataset into a balanced one can significantly improve the model's performance, particularly for the minority class. Techniques such as oversampling the minority class, undersampling the majority class, leading to more reliable and accurate predictions.

We used two Arabic datasets, ASTD[2] (Nabil, Aly et al. 2015) and AJGT[3] (Alomari, ElSherif et al. 2017), providing a comprehensive foundation for conducting sentiment analysis in Arabic text. The ASTD dataset, "Arabic Sentiment Tweets Dataset," encompasses a collection of 10,006 tweets written in both Modern Standard Arabic (MSA) and the Egyptian dialect. These tweets are categorized into four sentiment classes: positive, negative, neutral, and objective, with numerical representations (1, -1, 0, and -2) respectively. The distribution of tweets across these categories is as follows: 799 tweets classified as positive, 1,684 as negative, 832 as neutral, and the majority, 6,691, categorized as objective, as shown in the Figure 2. This distribution reflects the diverse nature of sentiment expressions present in Arabic tweets, covering a wide range of emotions and opinions.

[2] https://github.com/dahouabdelghani/DE-CNN/blob/master/datasets/ASTD.csv
[3] https://github.com/komari6/Arabic-twitter-corpus-AJGT

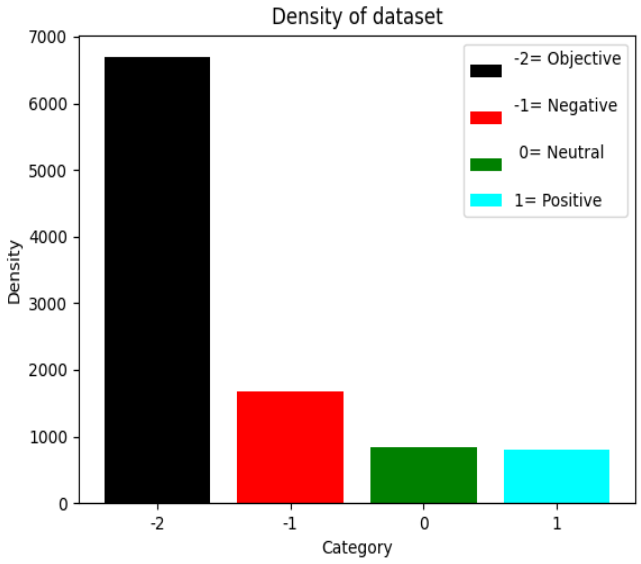


Figure 2: Distributed ASTD dataset

The second dataset is the AJGT, "Arabic Jordanian General Tweets." This dataset comprises 1,800 tweets written in MSA and the Jordanian dialect. The tweets are categorized into two sentiment classes: positive and negative. The dataset is evenly distributed between these two categories, with 900 tweets classified as positive and an equal number of 900 tweets classified as negative, as shown in the Figure 3. This balanced distribution ensures that the dataset adequately represents both positive and negative sentiments expressed in Arabic tweets. The AJGT dataset offers insights into sentiment expressions across different linguistic variations, enriching the understanding of sentiment analysis in Arabic text.

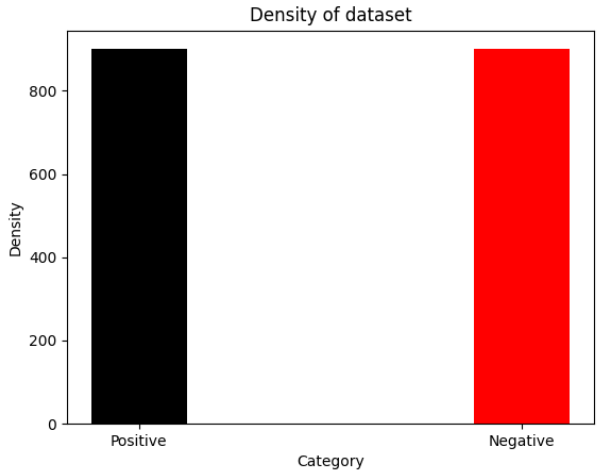


Figure 3: Distributed AJGT dataset

3.1.2 Preprocessing in sentiment analysis

Preprocessing is a crucial step that involves cleaning and preparing text data before feeding it into machine learning models or algorithms used for classification. The purpose of preprocessing is to format it in a way that it can be manipulated to improve the quality of the data and make it more suitable for analysis. Here are some pre-processing steps used in sentiment analysis:

1. Tokenization) Oueslati, Cambria et al. 2020, Oussous, Benjelloun et al. 2020) : tokenization can be performed at both the word level and the sentence level, depending on the specific requirements of the task. It involves dividing the text into individual words or tokens. This step helps break down the text into manageable units for analysis. Sentence Tokenization involves segmenting a text into individual sentences. Each sentence is treated as a separate unit for analysis.

2. Removing Special Characters and Punctuation (Almansor and Al-Ani , El-Masri, Berardinelli et al. 2017): involves removing non-Arabic letters and symbols that are irrelevant to the analysis process. These non-Arabic characters may include symbols, numbers, or punctuation marks from other languages that do not contribute to the sentiment expressed in the text. Removing such characters helps streamline the text and eliminates distractions that could interfere with the accuracy of sentiment analysis.
3. Removing Stop Words (El-Masri, Berardinelli et al. 2017, Oussous, Benjelloun et al. 2019): common words that occur frequently in the language but usually do not carry important meaning or feelings are removed, such as definite articles, prepositions, and conjunctions. Examples of stop words in Arabic include "من", "في", "و", and "إلى". By removing stop words from the text data before analysis, we can focus on the words that carry more sentiment and meaning, which are essential for accurately determining the sentiment expressed in the text. This process helps reduce noise and improve the efficiency of sentiment analysis algorithms, leading to more accurate results.
4. Stemming (Duwairi 2014, Oueslati, Cambria et al. 2020): involves reducing words to their root or base form by removing affixes such as prefixes, suffixes, and infixes. This process aims to transform different inflected forms of a word into a common base form, known as the word stem. In Arabic, stemming helps in reducing variations caused by different conjugations, declensions, and morphological changes. Examples of stemming algorithms for Arabic include the Khoja (Elhassan and Ahmed 2016), ISRI (Taghva, Elkhoury et al. 2005),

and Light (Abainia, Ouamour et al. 2016) stemmers. Examples of original words in Arabic are "يتحدث, تتحدث, تحدثون, تحدثوا" have the same stem which is "حدث".

5. Lemmatization (Anoual and Zeroual 2021, Birjali, Kasri et al. 2021): involves identifying the canonical or dictionary form of a word, known as the lemma. Unlike stemming, lemmatization considers the morphological analysis of words and maps them to their dictionary entries, preserving their semantic meaning. In Arabic, lemmatization is particularly useful due to the rich morphology of the language and the presence of root-and-pattern morphology. The lemma of the words "كتب" and "كتيبات" and "كتابان" is "كتاب".

   Both stemming and lemmatization help in reducing the vocabulary size, standardizing word forms, and improving the accuracy of sentiment analysis by treating morphologically related words as equivalents. These techniques contribute to better feature extraction and classification, leading to more effective sentiment analysis results in Arabic text.

6. N-gram Generation (Duwairi 2014, Gamal, Alfonse et al. 2019): involves creating sequences of n consecutive tokens from the text. Unigram, bigram, and trigram refer to different types of n-grams, which are contiguous sequences of n items (words, characters, etc.) in a text. A unigram is a single word occurring in a text, bigram is a sequence of two consecutive words and trigram is a sequence of three consecutive words.

These preprocessing steps help standardize and clean the text data, making it more conducive to accurate sentiment analysis. The specific preprocessing techniques

applied may vary depending on the characteristics of the text data and the requirements of the sentiment analysis task. In Table 1, we show how to apply pre-processing techniques for Arabic text. These techniques help clean and prepare the text data before further analysis or processing (For more details on apply pre-processing techniques according to the Arabic dataset used using Python, see the appendix).

Table 1: pre-processing for Arabic text

Pre-processing Technique	**Example**
Original text	اربد فيها جامعات اكثر من عمان. اريد ان ادرس فيها.
Tokenization	"اربد", "فيها", "جامعات", "اكثر", "من", "عمان.", "اريد", "ان", "ادرس", "فيها."
Removing Special Characters and Punctuation	اربد فيها جامعات اكثر من عمان اريد ان ادرس فيها
Removing Stop Words	اربد جامعات عمان. اريد ادرس
Stemming	اربد جامع عمان اراد ادرس
Lemmatization	اربد جامعة عمان اريد درس
N-gram Generation (Bigram)	"اربد فيها", "فيها جامعات", "جامعات اكثر", "اكثر من", "من عمان", "عمان اريد", " اريد ان", "ان ادرس", "ادرس فيها"

3.1.3 Feature Extraction

Feature Extraction (Anoual and Zeroual 2021, Birjali, Kasri et al. 2021) involves converting raw text data into a format that machine learning algorithms can understand and process effectively. It aims to represent text data in a numerical format by extracting relevant features that capture meaningful information about the text. Bag of

Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word Embeddings are common techniques used for feature extraction in NLP.

3.1.4 Classification techniques

Classification techniques (Abdullah and Hadzikadic 2017, Sayed, Elgeldawi et al. 2020) are algorithms used to categorize data into predefined classes or categories based on input features. In sentiment analysis, these techniques are applied to classify text data into sentiment categories such as positive, negative, or neutral. Machine learning algorithms are commonly used for this task, where a dataset is split into training and testing sets. Here are some commonly used classification techniques in sentiment analysis:

1) **Logistic Regression (LR)** (Zou, Hu et al. 2019) serves as a fundamental and extensively utilized statistical technique tailored for binary classification tasks. Despite its nomenclature, logistic regression operates as a classification algorithm rather than a regression one. The term "regression" is applied due to its utilization of a technique akin to linear regression for parameter estimation, yet it primarily predicts the probability of an event's occurrence by fitting data to a logistic curve.

   In LR, the output variable is a binary categorical variable (e.g., 0 or 1, True or False). The goal is to model the probability that an input belongs to one of the two classes based on one or more predictor variables. LR accomplishes this by applying a logistic function (sigmoid function) to a linear combination of the predictor variables. The logistic function maps any real-valued input into the range [0, 1], which can be interpreted as the probability of the positive class.

During the training phase, logistic regression estimates the coefficients (weights) of the model using optimization techniques such as maximum likelihood estimation or gradient descent. Once trained, the model can be used to predict the probability of the positive class for new input data, and a threshold can be applied to convert these probabilities into class labels.

LR is preferred when the relationship between the input variables and the output is assumed to be linear, the classes are linearly separable, and the emphasis is on understanding the influence of each predictor variable on the outcome.

2) **Random Forest (RF)** (Al-Amrani, Lazaar et al. 2018, Umarani, Julian et al. 2021) stands out as a versatile and potent machine learning algorithm within the ensemble learning domain. Renowned for its efficacy in both classification and regression tasks, RF excels in delivering robust and accurate predictions across diverse domains. The algorithm's modus operandi involve the creation of multiple decision trees during the training phase. Each tree is crafted using a subset of the training data and a random assortment of features, thereby enhancing the model's resilience and predictive capabilities.

   One of the key strengths of RF lies in its ability to mitigate overfitting, a common challenge in machine learning, especially with complex models. By generating multiple decision trees and averaging their predictions, RF reduces the variance of the model, making it more robust to noise and outliers in the data. Additionally, the random selection of features at each node split ensures that the individual trees are diverse, which further enhances the generalization ability of the ensemble.

Another advantage of RF is its capability to handle high-dimensional datasets with mixed data types. It can efficiently handle thousands of features and large training datasets without requiring extensive preprocessing or feature engineering. Moreover, RF provides a measure of feature importance, allowing users to identify the most relevant features for prediction. This insight can aid in feature selection and model interpretation, making the algorithm valuable for both predictive modeling and exploratory data analysis.

3) **Naive Bayes (NB)** (Birjali, Kasri et al. 2021) emerges as a straightforward yet effective probabilistic classifier rooted in Bayes' theorem, operating under the assumption of feature independence. Renowned for its utility in text classification endeavors, including sentiment analysis, NB calculates the posterior probability of each class given the observed features. This computation leverages the prior probabilities of classes and the likelihood of features given each class, as depicted in equation 1(Al-Batah, Mrayyen et al. 2019). Its "naive" assumption allows it to operate efficiently, especially in high-dimensional spaces common in text data, by treating features as independent. While it may struggle with highly correlated features and cannot capture complex relationships, Naive Bayes remains popular due to its computational efficiency, minimal training data requirements, and robustness to noisy data. With variants like Multinomial, Gaussian, and Bernoulli Naive Bayes (Abo, Idris et al. 2021).

$$\rho(C/X) = \frac{\rho(X/C).\rho(X)}{\rho(X)} \quad \textbf{(1)}$$

where:

- $\rho(C/X)$ represents the posterior probability of class C given evidence X.
- $\rho(X/C)$ denotes the likelihood of observing evidence X given class C.
- ρ(C) signifies the prior probability of class C.
- ρ(X) stands for the probability of observing evidence X (constant across all classes).

4) **Support Vector Machines (SVM)** (Elnagar, Lulu et al. 2018, Abo, Idris et al. 2021) are robust supervised learning models utilized for classification and regression tasks. While particularly adept at binary classification, SVMs can also be extended to accommodate multi-class classification challenges. The primary aim of SVM is to identify the hyperplane that most effectively segregates classes within the feature space, while simultaneously maximizing the margin between these classes. This hyperplane selection is optimized to maximize the distance between the nearest data points from each class, referred to as support vectors.

   SVM works by transforming the input data into a higher-dimensional space using a kernel function, which allows for nonlinear decision boundaries. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels. By mapping the data into a higher-dimensional space, SVM can find a hyperplane that separates the classes even if they are not linearly separable in the original feature space.

   One of the key strengths of SVM is its ability to handle high-dimensional data and effectively classify data points even when the number of features exceeds the number of samples. SVMs are also robust to overfitting, especially when using a regularization parameter to control the complexity of the model.

3.1.5 Evaluation

Evaluation (Touahri and Mazroui 2021) is a critical phase in the machine learning pipeline, where the performance of a model is assessed to determine its effectiveness in solving a particular task. It involves measuring various metrics to gauge how well the model is performing and to identify areas for improvement. In the context of sentiment analysis, evaluation helps determine the accuracy and reliability of the sentiment predictions made by the model.

Several metrics are commonly used to evaluate the performance of sentiment analysis models:

- Accuracy measures the proportion of correctly classified instances out of the total instances. It provides an overall indication of how well the model is performing in terms of correctly predicting sentiment labels.

The accuracy (ACC) is calculated using the following equation (2) (Başarslan and Kayaalp 2021, Chola, Heyat et al. 2021, Muaad, Kumar et al. 2022):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad \textbf{(2)}$$

Where:

- TP (True Positives): refer to the instances that are correctly predicted as positive.
- TN (True Negatives): denote the instances that are correctly predicted as negative.
- FP (False Positives): represent the instances that are incorrectly predicted as positive.
- FN (False Negatives): indicate the instances that are incorrectly predicted as negative.

- Precision assesses the accuracy of positive predictions by measuring the proportion of true positive predictions among all positive predictions made by the model. It helps evaluate the model's capability to avoid false positives.

Precision (P) is computed as in Equation (3):

$$P = \frac{TP}{TP + FP} \tag{3}$$

- Recall evaluates the model's ability to identify all positive instances by measuring the proportion of true positive predictions among all actual positive instances in the dataset. It helps assess the model's capability to avoid false negatives.

Recall (R) is computed as in Equation (4):

$$R = \frac{TP}{TP + FN} \tag{4}$$

- F1 score, a balanced measure of a model's performance, is the harmonic mean of precision and recall. It considers both false positives and false negatives, making it particularly beneficial for evaluating models on imbalanced datasets.

F1_Score is calculated as in Equation (5):

$$\text{F1\_Score} = \frac{2 * P * R}{P + R} \tag{5}$$

A confusion matrix is a tabular representation that summarizes the performance of a classification model. It provides insight into the number of true positive, true negative, false positive, and false negative predictions made by the model.

Evaluation also involves comparing the performance of different models or variations of the same model to identify the most effective approach. This may include conducting cross-validation experiments, where the dataset is divided into training and testing sets multiple times to obtain robust performance estimates.

Overall, evaluation is a crucial step in the machine learning workflow that helps validate the effectiveness of sentiment analysis models and guide decision-making regarding model selection and optimization.

We frequently used accuracy as a metric to measure the performance of our sentiment analysis model because it provides a straightforward and easily interpretable indication of the model's effectiveness. As a simple metric, accuracy represents the percentage of correct predictions out of the total predictions made by the model, making it easy to understand and offering a quick overview of its performance. It offers a general sense of how well the model classifies sentiment across different classes, such as positive, negative, and neutral. When the dataset is balanced, accuracy serves as a reliable indicator of overall model performance. Additionally, accuracy is often employed as a baseline metric for comparing different models or algorithms, where a higher accuracy typically suggests better performance. In scenarios where the dataset has a relatively balanced distribution of classes, accuracy can meaningfully capture the model's performance across all classes.

3.2 The Proposed Sentiment Analysis framework

The sentiment analysis model is depicted within a comprehensive framework that encompasses various paths and components. These paths represent different stages and processes involved in analyzing sentiment in text data, as explained in Figure 4.

The proposed sentiment analysis model consists of a four-stage pipeline aimed at analyzing sentiment in Arabic text and exploring the impact of translation and synonym enrichment on sentiment analysis.

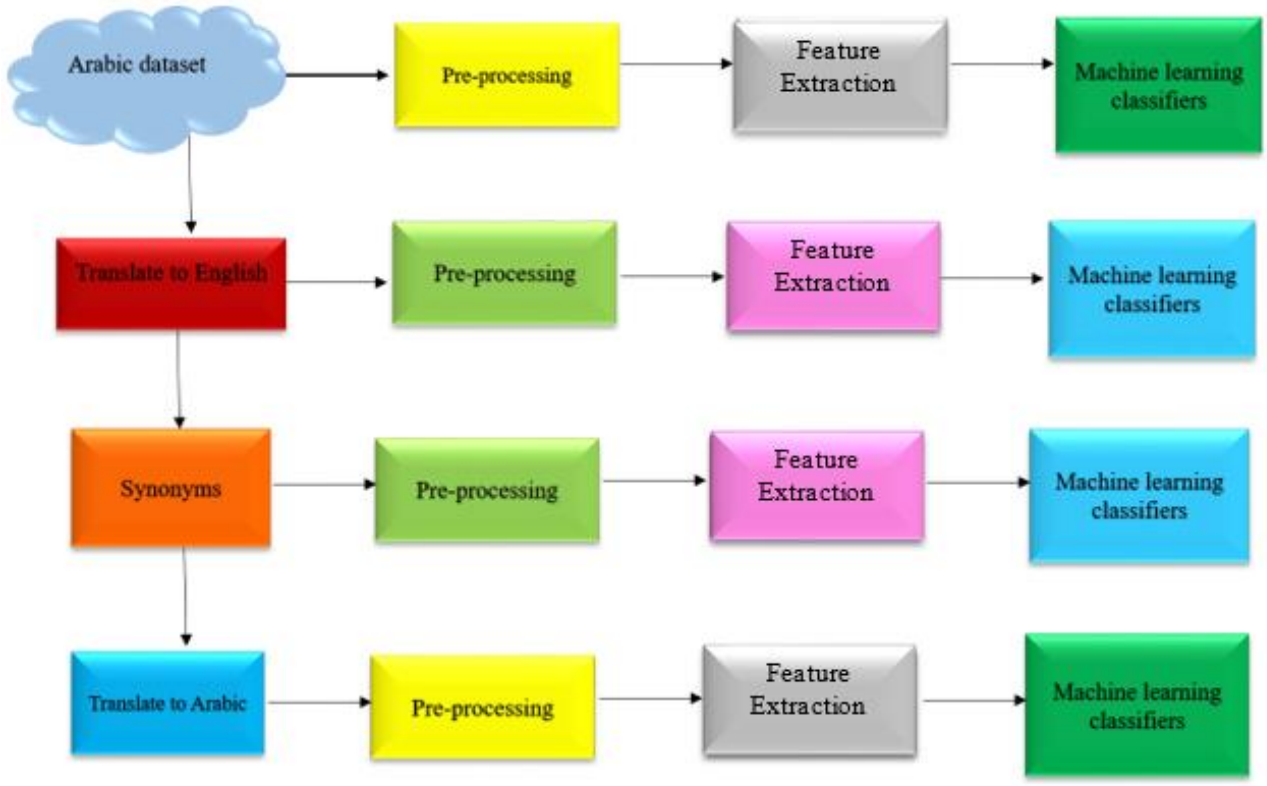


Figure 4: The framework of the proposed approach

3.2.1 The first stage of the proposed model

The sentiment analysis process in the first stage proposed begins with the utilization of Arabic origin datasets, which are accessed through an Excel file using Python software. Subsequently, the data undergoes pre-processing (Fouad, Mahany et al. 2020), a crucial step involving tokenization, removal of stop words, elimination of punctuation marks, and application of stemming and lemmatization techniques (Oussous, Benjelloun et al. 2019). Following this pre-processing phase, relevant features are extracted from the data. Machine learning algorithms are LR, RF, NB, and SVM then employed to evaluate the performance of the sentiment analysis model. The dataset is divided into a training set comprising 70% of the data and a test set containing the remaining 30%. Additional experiments were conducted by dividing the data into 80% for training and 20% for testing, as well as 65% for training and 35% for testing. It was found that there was no significant difference in the results between these different splits. However, the best results were consistently achieved using the 70% training and 30% testing division. This division facilitates effective training and validation of the model, ensuring robust performance on unseen data.

3.2.2 The second stage of the proposed model

In the second stage of the proposed model, the Arabic datasets undergo translation into English utilizing the Google Translate API. Following translation, the English text is subjected to pre-processing steps tailored to the specific characteristics of the English language. This pre-processing phase involves techniques such as tokenization, removal of stop words, punctuation removal, stemming, and lemmatization. Once pre-processed, relevant features are identified from the English text. Subsequently, machine learning classifiers are applied to the feature-extracted data. Finally, the performance of the sentiment analysis model is evaluated using

established evaluation metrics. This stage enables the assessment of sentiment analysis accuracy and effectiveness when applied to translated English text originating from Arabic datasets.

Translation from Arabic to English, especially when dealing with colloquial dialects, can significantly impact the text and the accuracy of sentiment analysis classifiers. Colloquial Arabic dialects often contain cultural references, idiomatic expressions, and informal language that may not have direct equivalents in English. This can lead to a loss of nuance and meaning during translation, which in turn affects the accuracy of sentiment classification. Sentiment expressed in Arabic, particularly in dialects, may not translate clearly into English, leading to potential misinterpretation. Positive or negative connotations in Arabic could be misrepresented or neutralized in the translation process, resulting in incorrect sentiment labeling by classifiers.

The translation process may also introduce errors, ambiguities, or simplifications that degrade the quality of the text used for training and testing classifiers. As a result, the model may not fully capture the original sentiment intended in the Arabic text, leading to lower accuracy. Additionally, the significant variability among Arabic dialects can further complicate the translation. A single English translation might not accurately reflect the differences between regional dialects, leading to inconsistencies in the translated text and further challenging the sentiment analysis process.

Cultural context plays a crucial role in sentiment analysis, and certain sentiments and expressions are deeply rooted in cultural nuances that do not easily transfer between languages. The loss of this context during translation can diminish the effectiveness of sentiment classifiers trained on translated text. Overall, the translation from Arabic to English can

introduce noise and distort the data, potentially reducing the performance of sentiment analysis classifiers, particularly when dealing with colloquial or region-specific dialects.

3.2.3 The third stage of the proposed model

In the third stage of the proposed model, the focus shifts to the extraction of synonyms for English language datasets. This involves identifying synonyms for words and phrases within the English text, expanding the vocabulary and enhancing the context for sentiment analysis. Once synonyms are extracted, the English language datasets undergo pre-processing steps similar to those applied in the previous stages, including tokenization, stop words removal, punctuation removal, stemming, and lemmatization. After pre-processing, relevant features are determined from the text containing synonyms. Subsequently, machine learning classifiers are employed to analyze the sentiment of the text based on the extracted features. Finally, the performance of the sentiment analysis model is evaluated using established metrics, allowing for an assessment of accuracy and effectiveness in sentiment analysis with synonym-enriched English language datasets.

3.2.4 The fourth stage of the proposed model

In the fourth stage of the proposed model, the focus is on translating the synonyms of the English datasets into Arabic. This involves converting the enriched English language datasets, containing synonyms, back into Arabic using Google Translate API. Once the translation is complete, the Arabic datasets undergo pre-processing steps similar to those applied in earlier stages, including tokenization, stop words removal, punctuation removal, stemming, and lemmatization. After pre-processing, relevant features are determined from the translated Arabic text. Machine learning classifiers are then applied to analyze the sentiment of the translated Arabic text based on the extracted features. Finally, the performance of the sentiment

analysis model is evaluated using established metrics to assess its accuracy and effectiveness in sentiment analysis with translated synonym-enriched Arabic language datasets.

3.3 Summary

In this chapter, we provided an extensive overview of the various stages involved in sentiment analysis model. Which begins with the collection of data, followed by a meticulous preprocessing phase aimed at refining the dataset. Subsequently, then extraction of relevant features essential for sentiment analysis. Utilizing machine learning classifiers, then a comprehensive evaluation employing predefined metrics.

Furthermore, we elucidated the four distinct stages constituting the proposed model. Initially, we engaged with Arabic datasets, progressing to their translation into English. Leveraging the English counterparts, we proceeded to extract synonyms, subsequently reverting to Arabic translation. This intricate process underscores our endeavor to augment sentiment analysis through multilingual and synonym-enriched datasets.

CHAPTER 4:

EXPERIMENTAL SETUP AND

EVALUATION

Chapter 4 - Experimental Setup and Evaluation

In our proposed four-stage model designed to assess sentiment analysis accuracy in Arabic, we conducted extensive experiments using the ASTD and AJGT datasets across each stage. Employing four machine learning classifiers —LR, RF, NB, and SVM— we leveraged TF-IDF technology for feature vectorization. Our approach encompassed various preprocessing techniques, such as tokenization, stop words removal, punctuation marks removal, stemming, lemmatization, and N-gram generation, as outlined in the third section. These methods were instrumental in refining the datasets and enhancing the quality of sentiment analysis outcomes.

4.1 Experiments – First Stage

In the initial part experiments conducted using Python and NLTK, a widely-used open-source library and platform for natural language processing (NLP) tasks, we employed various pre-processing techniques outlined in the previous chapter. Table 2 presents the accuracy results for the Arabic ASTD dataset categorized into positive, negative, neutral, and objective, focusing on one pre-processing step per experiment.

The first experiment involved no pre-processing, serving as a baseline. Subsequent experiments utilized word tokenization, achieved through the wordpunct_tokenize package, while also removing punctuation and Arabic stop words. Additionally, stemming was performed using three different stemmers: ISRI Stemmer (Syarief, Kurahman et al. 2019), Tashaphyne (ArabicLight Stemmer) (Zerrouki 2023), and Snowball tool. Furthermore, lemmatization was applied using the qalsadi lemmatizer for Arabic datasets and applied n-gram generation with n set to 2 and 3 (bigram and trigram).

These experiments aimed to assess the impact of each pre-processing step on sentiment analysis accuracy, providing insights into the effectiveness of different techniques in improving the performance of the sentiment analysis model.

Table 2: Accuracy Analysis Across Different cases Using Single Features for ASTD Dataset

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	69%	68%	67%	70%
Tokenizing	69%	68%	67%	70%
Remove punctuation	69%	68%	67%	70%
Remove stop word	69%	68%	67%	70%
ISRI_ Stem	70%	68%	67%	70%
ArListem_ Stem	70%	68%	67%	70%
Snowball_ Stem	70%	68%	67%	70%
Lemmatization	69%	69%	67%	70%
Bigram	69%	68%	67%	70%
Trigram	69%	68%	67%	69%

In addition to word tokenization using wordpunct_tokenize, we combined this feature with others, as demonstrated in Table 3. Furthermore, in Table 4, we incorporated Bigram and Trigram features alongside other preprocessing techniques to further explore their impact on sentiment analysis accuracy. Table 5 displays the accuracy outcomes of experiments employing two lemmatization features in conjunction with three stemmers.

Table 3: Accuracy Analysis Across Different cases Using Tokenizing for ASTD Dataset

Feature	**LR**	**RF**	**NB**	**SVM**
Tokenizing+ Remove punctuation	68%	68%	67%	68%
Tokenizing+ Remove stop word	69%	68%	67%	70%
Tokenizing+ ISRI_ Stem	70%	68%	67%	70%
Tokenizing+ ArListem_ Stem	70%	68%	67%	70%

Tokenizing + Snowball_ Stem	70%	68%	67%	70%
Tokenizing + Lemmatization	69%	69%	67%	70%

Table 4: Accuracy Analysis Across Different cases Using for Bigram and Trigram for ASTD Dataset

Feature	**LR**	**RF**	**NB**	**SVM**
Bigram + Remove punctuation	70%	68%	67%	70%
Bigram +Remove stop word	69%	68%	67%	69%
Bigram + ISRI_ Stem	69%	68%	67%	70%
Bigram + ArListem_ Stem	69%	68%	67%	69%
Bigram + Snowball_ Stem	69%	68%	67%	69%
Bigram + Lemmatization	69%	68%	67%	70%
Trigram + Remove punctuation	69%	68%	67%	70%
Trigram +Remove stop word	68%	68%	67%	70%
Trigram + ISRI_ Stem	70%	68%	67%	71%
Trigram + ArListem_ Stem	69%	68%	67%	70%
Trigram + Snowball_ Stem	69%	68%	67%	70%
Trigram + Lemmatization	69%	68%	67%	70%

Table 5: Accuracy Analysis Using lemmatization with three stemmers for ASTD Dataset

Feature	**LR**	**RF**	**NB**	**SVM**
Lemmatization + ISRI_ Stem	70%	68%	67%	70%
Lemmatization + ArListem_ Stem	69%	68%	67%	70%
Lemmatization +Snowball_ Stem	69%	69%	67%	70%

Tables 2, 3, 4, and 5 illustrate the accuracy performance results of the four classifiers (LR, RF, NB, and SVM classifiers). Table 2 indicates that the SVM classifier achieved the highest

accuracy, reaching 70%. Notably, the stemming feature yielded the best results across all experiments, with no significant variance observed among the three stemmers used.

When combining two features together, the most favorable results were observed with the tokenization feature combined with stemming, as depicted in Table 3. Notably, the SVM classifier exhibited the best performance. In Table 4, the highest accuracy was attained with the Trigram feature alongside the ISRI Stemmer, with the SVM classifier achieving an accuracy of 71%. Additionally, in Table 5, the fusion of Lemmatization and the ISRI Stemmer features yielded an accuracy of 70%.

In the second part of the experiments utilizing the ASTD dataset, we filtered the data by removing the objective category. Table 6 presents the accuracy results of the classifiers' performance using individual features. Additionally, Tables 7, 8, and 9 showcase the accuracy outcomes when employing two features simultaneously, akin to the experiments conducted in the first part.

Table 6: Accuracy Analysis Across Different cases Using Single Features for ASTD Dataset without objective category

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	62%	61%	58%	62%
Tokenizing	62%	62%	58%	62%
Remove punctuation	62%	62%	58%	62%
Remove stop word	62%	63%	58%	61%
ISRI_ Stem	62%	61%	57%	61%
ArListem_ Stem	61%	61%	57%	61%
Snowball_ Stem	62%	61%	57%	63%
Lemmatization	62%	61%	57%	62%
Bigram	61%	60%	56%	61%
Trigram	56%	60%	55%	59%

Table 7: Accuracy Analysis Across Different cases Using Tokenizing for ASTD Dataset without objective category

Feature	LR	RF	NB	SVM
Tokenizing+ Remove punctuation	62%	62%	58%	62%
Tokenizing+ Remove stop word	62%	61%	58%	61%
Tokenizing+ ISRI_ Stem	62%	60%	57%	61%
Tokenizing+ ArListem_ Stem	61%	61%	57%	61%
Tokenizing + Snowball_ Stem	62%	60%	57%	63%
Tokenizing + Lemmatization _	62%	60%	57%	62%

Table 8: Accuracy Analysis Across Different cases Using for Bigram and Trigram for ASTD Dataset without objective category

Feature	LR	RF	NB	SVM
Bigram + Remove punctuation	61%	60%	56%	62%
Bigram +Remove stop word	61%	59%	57%	62%
Bigram + ISRI_ Stem	61%	59%	56%	62%
Bigram + ArListem_ Stem	59%	61%	56%	61%
Bigram + Snowball_ Stem	60%	59%	56%	60%
Bigram + Lemmatization	62%	60%	57%	62%
Trigram + Remove punctuation	58%	61%	56%	59%
Trigram +Remove stop word	58%	62%	57%	60%
Trigram + ISRI_ Stem	62%	62%	55%	61%
Trigram + ArListem_ Stem	59%	62%	55%	60%
Trigram + Snowball_ Stem	59%	62%	55%	60%
Trigram + Lemmatization	61%	61%	56%	60%

Table 9: Accuracy Analysis Using lemmatization with three stemmers for ASTD Dataset without objective category

Feature	LR	RF	NB	SVM
Lemmatization + ISRI_ Stem	62%	61%	58%	61%
Lemmatization + ArListem_ Stem	61%	61%	57%	62%
Lemmatization + Snowball_ Stem	62%	60%	57%	62%

The performance results of the classifiers are illustrated in Table 6. When utilizing a single feature, there was a decrease in accuracy compared to Table 2 after filtering the dataset by removing the objective category. Nonetheless, the SVM classifier demonstrated the best performance when employing the snowball stemmer, achieving an accuracy of 63%. Furthermore, in Table 7, combining tokenization with other features resulted in the SVM classifier attaining the highest accuracy of 63% when combined with the snowball stemmer. While Table 8 exhibited mixed results, the SVM classifier outperformed other classifiers with most features. Regarding Table 9, the combination of Lemmatization and Snowball Stemmer yielded one of the best results for the SVM and LR classifiers.

In the third part of our experiments with the ASTD dataset, we further filtered the dataset by removing both the objective and neutral categories. We then conducted experiments similar to the previous ones, the results of which are presented in Tables 10, 11, 12, and 13.

Table 10: Accuracy Analysis Across Different cases Using Single Features for ASTD Dataset without objective and neutral categories

Feature	LR	RF	NB	SVM
Without any feature	76%	74%	68%	76%
Tokenizing	76%	74%	68%	76%
Remove punctuation	76%	74%	68%	76%
Remove stop word	77%	74%	69%	77%

ISRI_ Stem	76%	76%	69%	75%
ArListem_ Stem	75%	74%	69%	75%
Snowball_ Stem	75%	75%	68%	76%
Lemmatization	76%	76%	68%	75%
Bigram	77%	72%	66%	76%
Trigram	75%	72%	66%	77%

Table 11: Accuracy Analysis Across Different cases Using Tokenizing for ASTD Dataset without objective and neutral categories

Feature	**LR**	**RF**	**NB**	**SVM**
Tokenizing+ Remove punctuation	76%	74%	68%	76%
Tokenizing+ Remove stop word	77%	74%	69%	77%
Tokenizing+ ISRI_ Stem	76%	75%	69%	75%
Tokenizing+ ArListem_ Stem	75%	75%	69%	75%
Tokenizing + Snowball_ Stem	75%	74%	68%	76%
Tokenizing + Lemmatization	76%	76%	78%	75%

Table 12: Accuracy Analysis Across Different cases Using for Bigram and Trigram for ASTD Dataset without objective and neutral categories

Feature	**LR**	**RF**	**NB**	**SVM**
Bigram + Remove punctuation	74%	73%	66%	76%
Bigram +Remove stop word	76%	73%	67%	77%
Bigram + ISRI_ Stem	75%	72%	67%	74%
Bigram + ArListem_ Stem	74%	72%	67%	74%
Bigram + Snowball_ Stem	74%	72%	67%	74%
Bigram + Lemmatization	74%	72%	67%	75%
Trigram + Remove punctuation	75%	72%	66%	77%
Trigram +Remove stop word	75%	73%	67%	77%
Trigram + ISRI_ Stem	75%	74%	66%	74%

Trigram + ArListem_ Stem	74%	74%	66%	74%
Trigram + Snowball_ Stem	75%	74%	66%	75%

Table 13: Accuracy Analysis Using lemmatization with three stemmers for ASTD Dataset without objective and neutral categories

Feature	**LR**	**RF**	**NB**	**SVM**
Lemmatization + ISRI_ Stem	76%	76%	69%	76%
Lemmatization + ArListem_ Stem	76%	75%	69%	77%
Lemmatization + Snowball_ Stem	76%	76%	68%	74%

In the subsequent phase of our experiments with the ASTD dataset, where we filtered out the objective and neutral categories, we observed notable improvements in accuracy across all classifiers (LR, RF, and SVM), as depicted in Tables 10, 11, 12, and 13.

Table 10 showcases a significant advancement, with both the SVM and LR classifiers achieving the highest accuracy of 77% when stop words were eliminated from the dataset. This indicates that removing common, less informative words contributed to refining the sentiment analysis process.

Moreover, in Table 11, employing the tokenization feature combined with stop words removal resulted in a commendable accuracy of 77% for both SVM and LR classifiers. This underscores the efficacy of tokenization in segmenting text data for analysis, particularly when coupled with preprocessing steps like stop words removal.

The SVM classifier demonstrated consistent performance across multiple experiments, as evidenced by the 77% accuracy achieved in various settings. This includes scenarios where Bigram and Trigram features were utilized alongside stop words or punctuation removal,

highlighting the robustness of the SVM classifier in handling diverse preprocessing techniques.

Overall, these results indicate that refining the dataset by filtering out specific categories and employing meticulous preprocessing techniques can substantially enhance the accuracy of sentiment analysis models across various classifiers.

In our experimentation with the AJGT dataset, which comprises only positive and negative categories, we conducted similar analyses as with the ASTD dataset, examining the impact of individual and combined preprocessing features on sentiment analysis accuracy. The results of these experiments are summarized in Tables 14, 15, 16, and 17.

Table 14: Accuracy Analysis Across Different cases Using Single Features for AJGT Dataset

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	84%	83%	85%	84%
Tokenizing	84%	84%	85%	84%
Remove punctuation	84%	82%	85%	84%
Remove stop word	86%	84%	86%	85%
ISRI_ Stem	83%	85%	86%	85%
ArListem_ Stem	86%	84%	86%	86%
Snowball_ Stem	84%	85%	87%	84%
Lemmatization	86%	85%	87%	87%
Bigram	83%	80%	83%	83%
Trigram	83%	81%	83%	82%

Table 15: Accuracy Analysis Across Different cases Using Tokenizing for AJGT Dataset

Feature	**LR**	**RF**	**NB**	**SVM**
Tokenizing+ Remove punctuation	84%	84%	85%	84%
Tokenizing+ Remove stop word	86%	83%	86%	85%

Tokenizing+ ISRI_ Stem	83%	84%	86%	85%
Tokenizing+ ArListem_ Stem	86%	84%	86%	86%
Tokenizing + Snowball_ Stem	84%	85%	87%	84%
Tokenizing + Lemmatization	86%	85%	87%	87%

Table 16: Accuracy Analysis Across Different cases Using for Bigram and Trigram for AJGT Dataset

Feature	**LR**	**RF**	**NB**	**SVM**
Bigram + Remove punctuation	82%	80%	83%	83%
Bigram +Remove stop word	81%	80%	84%	81%
Bigram + ISRI_ Stem	80%	82%	83%	80%
Bigram + ArListem_ Stem	86%	83%	84%	85%
Bigram + Snowball_ Stem	84%	83%	83%	83%
Bigram + Lemmatization	85%	81%	83%	85%
Trigram + Remove punctuation	83%	80%	84%	82%
Trigram +Remove stop word	82%	80%	84%	83%
Trigram + ISRI_ Stem	84%	84%	84%	84%
Trigram + ArListem_ Stem	84%	84%	84%	84%
Trigram + Snowball_ Stem	84%	83%	85%	84%

Table 17: Accuracy Analysis Using lemmatization with three stemmers for AJGT Dataset

Feature	**LR**	**RF**	**NB**	**SVM**
Lemmatization + ISRI_ Stem	84%	84%	86%	85%
Lemmatization + ArListem_ Stem	86%	84%	87%	86%
Lemmatization +Snowball_ Stem	86%	86%	87%	86%

Table 14 presents the accuracy outcomes obtained from employing individual preprocessing features. Notably, the SVM classifier attained the highest accuracy rate, reaching 87%, when utilizing the Lemmatization feature.

When amalgamating the two preprocessing features, as depicted in Table 15, notable enhancements in accuracy were observed, especially with the incorporation of tokenization and Lemmatization features. Notably, both the NB and SVM classifiers exhibited commendable performance, achieving an accuracy of 87% each.

In Table 16, we extended our analysis by introducing Bigram and Trigram features in conjunction with other preprocessing techniques. Despite the inclusion of these additional features, we did not observe a notable improvement in accuracy compared to our previous experiments. However, it's noteworthy that the LR classifier exhibited strong performance in this setting. Specifically, it achieved its highest accuracy of 86% when employing the Bigram feature in combination with the ArabicLight Stemmer. This suggests that while the addition of n-gram features did not yield significant improvements overall, certain combinations of features may still contribute positively to the classification accuracy, particularly in the context of logistic regression.

Continuing with Table 17, our exploration of combining lemmatization with three stemming techniques revealed significant performance discrepancies among the classifiers. Notably, the Naive Bayes (NB) classifier exhibited superior accuracy, reaching 87% when combining Lemmatization with ArabicLight Stemmer and also with Snowball Stemmer. This outcome underscores the potential effectiveness of the NB classifier when coupled with tailored preprocessing techniques, emphasizing the importance of selecting appropriate feature combinations for optimal sentiment analysis performance.

Overall, these results underscore the effectiveness of various preprocessing techniques and feature combinations in enhancing sentiment analysis accuracy, particularly when applied to diverse datasets like the AJGT dataset.

4.2 Experiments – Second Stage

In order to evaluate the effects of transitioning from Arabic to English in our model, we utilized the Google Translate package within the Python environment to translate our dataset. This process involved converting the text from Arabic to English. Subsequently, we conducted a series of experiments with preprocessing techniques tailored specifically for English text.

The objective of these experiments was to assess how the shift in language impacted the overall performance and accuracy of sentiment analysis. By comparing the results obtained from the English-preprocessed data with those from the original Arabic dataset, we aimed to gain insights into the effectiveness of our model across different languages and preprocessing approaches.

Tables 18, 19, 20, and 21 present the accuracy results of the classifiers applied to the ASTD dataset categorized into positive, negative, neutral, and objective after translation into English. These tables offer insights into the performance of various machine learning classifiers when applied to English-translated data and demonstrate the impact of language translation on sentiment analysis accuracy.

Table 18: Accuracy Analysis Across Different cases Using Single Features for ASTD Dataset translated to English

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	70%	68%	67%	70%
Tokenizing	70%	68%	67%	70%
Remove punctuation	69%	68%	67%	70%

Remove stop word	69%	69%	67%	70%
Porter_ Stem	70%	68%	67%	70%
Snowball_ Stem	70%	68%	67%	70%
Lemmatization	70%	68%	67%	70%
Bigram	69%	68%	67%	69%
Trigram	69%	68%	67%	70%

Table 19: Accuracy Analysis Across Different cases Using Tokenizing for ASTD Dataset translated to English

Feature	**LR**	**RF**	**NB**	**SVM**
Tokenizing+ Remove punctuation	69%	68%	67%	70%
Tokenizing+ Remove stop word	69%	69%	67%	70%
Tokenizing+ Porter_ Stem	69%	68%	67%	70%
Tokenizing + Snowball_ Stem	70%	68%	67%	70%
Tokenizing + Lemmatization	70%	68%	67%	70%

Table 20: Accuracy Analysis Across Different cases Using for Bigram and Trigram for ASTD Dataset translated to English

Feature	**LR**	**RF**	**NB**	**SVM**
Bigram + Remove punctuation	69%	68%	67%	69%
Bigram +Remove stop word	69%	68%	67%	69%
Bigram + Porter_ Stem	69%	68%	67%	69%
Bigram +Snowball_ Stem	69%	68%	67%	69%
Bigram + Lemmatization	69%	68%	67%	69%
Trigram + Remove punctuation	69%	68%	67%	70%
Trigram +Remove stop word	69%	68%	67%	69%
Trigram + Porter_ Stem	69%	68%	67%	69%
Trigram + Snowball_ Stem	69%	67%	67%	69%
Trigram + Lemmatization	69%	68%	67%	70%

Table 21: Accuracy Analysis Using lemmatization with two stemmers for ASTD Dataset translated to English

Feature	LR	RF	NB	SVM
Lemmatization + Porter_ Stem	70%	68%	67%	70%
Lemmatization + Snowball_ Stem	70%	68%	67%	69%

Table 18 illustrates the SVM classifier's consistent high performance across individual features, achieving an accuracy of 70%. Interestingly, the translation from Arabic to English did not significantly impact the results. Moving to Table 19, combining the tokenizing feature with others showed improved performance for the SVM classifier, maintaining an accuracy of 70%. In Table 20, the SVM classifier's accuracy ranged from 69% to 70% when incorporating bigram or trigram features with other preprocessing techniques. Additionally, combining Lemmatization and Porter Stemmer resulted in 70% accuracy for both the SVM and LR classifiers.

In the subsequent phase of our experimentation, we translated the ASTD dataset into English while excluding the objective category. Tables 22, 23, 24, and 25 present the accuracy results of the classifiers, replicating the same experiments conducted previously.

Table 22: Accuracy Analysis Across Different cases Using Single Features for ASTD Dataset translated to English without objective category

Feature	LR	RF	NB	SVM
Without any feature	64%	62%	58%	64%
Tokenizing	64%	62%	58%	64%
Remove punctuation	63%	62%	58%	64%
Remove stop word	65%	63%	61%	64%
Porter_ Stem	64%	61%	58%	64%
Snowball_ Stem	65%	62%	59%	64%
Lemmatization	63%	61%	58%	64%

Bigram	62%	59%	57%	62%
Trigram	61%	61%	56%	63%

Table 23: Accuracy Analysis Across Different cases Using Tokenizing for ASTD Dataset translated to English without objective category

Feature	**LR**	**RF**	**NB**	**SVM**
Tokenizing+ Remove punctuation	63%	61%	58%	64%
Tokenizing+ Remove stop word	65%	63%	61%	64%
Tokenizing+ Porter_ Stem	64%	61%	58%	64%
Tokenizing + Snowball_ Stem	64%	62%	59%	64%
Tokenizing + Lemmatization	63%	61%	58%	63%

Table 24: Accuracy Analysis Across Different cases Using for Bigram and Trigram for ASTD Dataset translated to English without objective category

Feature	**LR**	**RF**	**NB**	**SVM**
Bigram + Remove punctuation	62%	60%	57%	61%
Bigram +Remove stop word	63%	61%	60%	61%
Bigram + Porter_ Stem	62%	61%	57%	62%
Bigram + Snowball_ Stem	62%	61%	57%	62%
Bigram + Lemmatization	62%	60%	57%	63%
Trigram + Remove punctuation	61%	61%	56%	62%
Trigram +Remove stop word	63%	62%	59%	63%
Trigram + Porter_ Stem	62%	62%	56%	63%
Trigram + Snowball_ Stem	62%	62%	56%	63%
Trigram + Lemmatization	62%	61%	56%	63%

Table 25: Accuracy Analysis Using lemmatization with two stemmers for ASTD Dataset translated to English without objective category

Feature	LR	RF	NB	SVM
Lemmatization + Porter_ Stem	64%	62%	58%	64%
Lemmatization + Snowball_ Stem	64%	62%	58%	65%

Table 22 exhibits a decline in accuracy compared to previous tables, with accuracy ranging between 56% and 63%. The LR and SVM classifiers performed best, achieving accuracy levels varying between 61% and 63% across all individual features. In Table 23, we observe a modest enhancement in results, particularly with the LR and SVM classifiers maintaining their performance. Both classifiers achieved accuracy rates ranging from 63% to 64%.

Table 24 did not exhibit any improvement compared to the preceding two tables, with the LR and SVM classifiers sustaining their performance levels. However, Table 25 indicated a slight enhancement in accuracy results, particularly when combining Lemmatization with stemmer in both LR and SVM classifiers. Furthermore, there was a marginal increase in accuracy for the SVM classifier when combining Lemmatization with Snowball stemmer, reaching an accuracy of 65%.

The experiments conducted on the translated ASTD dataset into English, after eliminating the objective and neutral categories, showcased notable advancements in accuracy. Particularly, when applying Porter stemmer features in one experiment and Lemmatization in another, the SVM classifier achieved an impressive accuracy of 80%, as shown in Table 26. This significant improvement compared to previous experiments underscores the efficacy of preprocessing techniques in enhancing sentiment analysis accuracy.

Table 26: Accuracy Analysis Across Different cases Using Single Features for ASTD Dataset translated to English without objective and neutral categories

Feature	LR	RF	NB	SVM
Without any feature	79%	75%	68%	79%
Tokenizing	79%	76%	68%	79%
Remove punctuation	79%	76%	68%	79%
Remove stop word	79%	77%	74%	79%
Porter_ Stem	79%	76%	69%	80%
Snowball_ Stem	79%	76%	69%	79%
Lemmatization	79%	76%	68%	80%
Bigram	77%	73%	67%	77%
Trigram	78%	74%	66%	78%

In experiments combining tokenization with other features, the SVM classifier demonstrated improved performance when combined with the Porter stemmer features in one experiment and Lemmatization in another experiment. Similarly, the LR classifier achieved an accuracy of 80%, as depicted in Table 27, when combining tokenization with removing stop words specific to the English language. These findings underscore the efficacy of integrating multiple preprocessing techniques to optimize sentiment analysis accuracy.

Table 27: Accuracy Analysis Across Different cases Using Tokenizing for ASTD Dataset translated to English without objective and neutral categories

Feature	LR	RF	NB	SVM
Tokenizing+ Remove punctuation	79%	74%	68%	79%
Tokenizing+ Remove stop word	80%	77%	73%	78%
Tokenizing+ Porter_ Stem	80%	76%	69%	80%
Tokenizing + Snowball_ Stem	79%	76%	69%	79%
Tokenizing + Lemmatization	79%	75%	68%	80%

Table 28: Accuracy Analysis Across Different cases Using for Bigram and Trigram for ASTD Dataset translated to English without objective and neutral categories

Feature	LR	RF	NB	SVM
Bigram + Remove punctuation	77%	74%	67%	77%
Bigram +Remove stop word	76%	75%	70%	75%
Bigram + Porter_ Stem	76%	75%	68%	78%
Bigram + Snowball_ Stem	77%	75%	68%	79%
Bigram + Lemmatization	77%	74%	67%	77%
Trigram + Remove punctuation	77%	74%	68%	77%
Trigram +Remove stop word	77%	74%	68%	77%
Trigram + Porter_ Stem	79%	75%	67%	79%
Trigram + Snowball_ Stem	77%	74%	68%	77%
Trigram + Lemmatization	78%	74%	67%	79%

Table 28 illustrates that the SVM and LR classifiers maintained their performance when combining the bigram and trigram features, with no noticeable increase in accuracy compared to previous experiments. Similarly, the results of combining lemmatization and stemmer features remained consistent, as depicted in Table 29.

Table 29: Accuracy Analysis Using lemmatization with two stemmers for ASTD Dataset translated to English without objective and neutral categories

Feature	LR	RF	NB	SVM
Lemmatization + Porter_ Stem	79%	76%	69%	79%
Lemmatization + Snowball_ Stem	79%	76%	69%	79%

When we translated the AJGT dataset into English, we didn't observe a significant boost in accuracy compared to its Arabic counterpart, as detailed in Tables 30, 31, 32, and 33. Notably, the SVM classifier performed best when using the Porter and Snowball Stemmer features, achieving an accuracy of 86%, as demonstrated in Table 30. Similarly, combining

tokenization with Porter and Snowball Stemmer features yielded an accuracy of 86%, as seen in Table 31. However, there wasn't a substantial change in results when combining Bigram and Trigram features with others, with LR, NB, and SVM classifiers showing similar outcomes. Interestingly, the SVM classifier achieved the highest accuracy of 86% when combining lemmatization and stemmer features.

Table 30: Accuracy Analysis Across Different cases Using Single Features for AJGT Dataset translated to English

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	83%	83%	84%	85%
Tokenizing	83%	81%	84%	84%
Remove punctuation	83%	83%	85%	84%
Remove stop word	84%	84%	84%	84%
Porter_ Stem	85%	84%	84%	86%
Snowball_ Stem	85%	84%	84%	86%
Lemmatization	85%	82%	85%	84%
Bigram	82%	80%	81%	81%
Trigram	83%	84%	84%	83%

Table 31: Accuracy Analysis Across Different cases Using Tokenizing for AJGT Dataset translated to English

Feature	**LR**	**RF**	**NB**	**SVM**
Tokenizing+ Remove punctuation	83%	83%	85%	84%
Tokenizing+ Remove stop word	84%	84%	84%	84%
Tokenizing+ Porter_ Stem	84%	85%	84%	86%
Tokenizing + Snowball_ Stem	85%	84%	84%	86%
Tokenizing + Lemmatization	85%	83%	85%	85%

Table 32: Accuracy Analysis Across Different cases Using for Bigram and Trigram for AJGT Dataset translated to English

Feature	LR	RF	NB	SVM
Bigram + Remove punctuation	83%	81%	81%	83%
Bigram +Remove stop word	82%	80%	80%	81%
Bigram + Porter_ Stem	84%	83%	82%	84%
Bigram + Snowball_ Stem	84%	83%	83%	83%
Bigram + Lemmatization	85%	81%	83%	85%
Trigram + Remove punctuation	83%	82%	84%	85%
Trigram +Remove stop word	80%	82%	81%	81%
Trigram + Porter_ Stem	85%	82%	85%	85%
Trigram + Snowball_ Stem	85%	83%	85%	84%
Trigram + Lemmatization	84%	82%	83%	82%

Table 33: Accuracy Analysis Using lemmatization with two stemmers for AJGT Dataset translated to English

Feature	LR	RF	NB	SVM
Lemmatization + Porter_ Stem	84%	83%	85%	86%
Lemmatization + Snowball_ Stem	85%	85%	85%	86%

Interestingly, despite translating the datasets into English, we observed that this process didn't notably alter the results. This finding suggests that the effectiveness of the sentiment analysis model remained consistent across both Arabic and English languages.

The translation from Arabic to English, particularly when dealing with colloquial dialects, often results in inaccuracies. These inaccuracies negatively affect the text, leading to poor performance of sentiment analysis classifiers. The accuracy of the classifiers was notably low, and the overall quality of the analysis suffered as well. This highlights the challenges of

translating culturally nuanced and region-specific language, where the loss of meaning and context can significantly undermine the effectiveness of sentiment classification.

4.3 Experiments – Third Stage

In the third stage of our experiments, we encountered the challenge of limited Arabic language resources. To address this issue, we turned to WordNet, an English-language lexical database, to generate synonyms for the Arabic dataset. Given the scarcity of Arabic sources, leveraging WordNet provided an alternative means to enrich the dataset with synonymous terms. This approach allowed us to expand the vocabulary available for analysis, potentially enhancing the model's performance in capturing nuanced linguistic nuances and sentiments. By incorporating English synonyms, albeit derived from Arabic text, we aimed to mitigate the limitations imposed by the lack of extensive Arabic language resources, thereby enabling more comprehensive sentiment analysis.

In the subsequent stage of our analysis, we revisited the experiments conducted on the translated ASTD dataset, this time incorporating English language synonyms derived from WordNet. Tables 34, 35, 36, and 37 present the accuracy outcomes of the classifiers following the inclusion of synonymous terms. Notably, the accuracy levels remained consistent, ranging between 67% and 70%. Both the SVM and LR classifiers exhibited the most favorable results, with no discernible variance in accuracy compared to the previous experiments on the translated ASTD dataset conducted prior to integrating synonyms.

Table 34: Accuracy Analysis Across Different cases Using Single Features for English Synonymous ASTD Dataset translated

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	70%	68%	67%	70%
Tokenizing	70%	68%	67%	70%

Remove punctuation	70%	68%	67%	70%
Remove stop word	69%	69%	68%	70%
Porter_ Stem	70%	68%	67%	70%
Snowball_ Stem	70%	68%	67%	70%
Lemmatization	70%	68%	67%	70%
Bigram	69%	68%	67%	69%
Trigram	70%	68%	67%	70%

Table 35: Accuracy Analysis Across Different cases Using Tokenizing for English Synonymous ASTD Dataset translated

Feature	**LR**	**RF**	**NB**	**SVM**
Tokenizing+ Remove punctuation	70%	68%	67%	70%
Tokenizing+ Remove stop word	69%	69%	68%	70%
Tokenizing+ Porter_ Stem	70%	68%	67%	70%
Tokenizing + Snowball_ Stem	70%	68%	67%	70%
Tokenizing + Lemmatization	70%	68%	67%	70%

Table 36: Accuracy Analysis Across Different cases Using for Bigram and Trigram for English Synonymous ASTD Dataset translated

Feature	**LR**	**RF**	**NB**	**SVM**
Bigram + Remove punctuation	70%	68%	67%	69%
Bigram +Remove stop word	69%	68%	67%	69%
Bigram + Porter_ Stem	69%	68%	67%	69%
Bigram + Snowball_ Stem	69%	68%	67%	69%
Bigram + Lemmatization	69%	68%	67%	69%
Trigram + Remove punctuation	70%	68%	67%	70%
Trigram +Remove stop word	69%	69%	67%	70%
Trigram + Porter_ Stem	70%	68%	67%	70%
Trigram + Snowball_ Stem	70%	68%	67%	70%
Trigram + Lemmatization	69%	68%	67%	70%

Table 37: Accuracy Analysis Using lemmatization with two stemmers for English Synonymous ASTD Dataset translated

Feature	LR	RF	NB	SVM
Lemmatization + Porter_ Stem	70%	68%	67%	70%
Lemmatization +Snowball_ Stem	70%	68%	67%	70%

On the contrary, upon eliminating the objective category from the translated ASTD dataset, we observed a slight uptick in classifier accuracy post-synonym integration, as depicted in Table 38. The SVM classifier attained an accuracy of 66% when punctuation removal was applied, mirroring the accuracy achieved when tokenization was coupled with punctuation removal, as demonstrated in Table 39. Furthermore, Table 40 illustrates comparable performance between the SVM and LR classifiers when combining either Bigram or Trigram features with other preprocessing techniques. Notably, Table 41 highlights the superior performance observed when integrating lemmatization with the Porter stemmer.

Table 38: Accuracy Analysis Across Different cases Using Single Features for English Synonymous ASTD Dataset translated without objective category

Feature	LR	RF	NB	SVM
Without any feature	65%	61%	59%	65%
Tokenizing	65%	61%	59%	65%
Remove punctuation	64%	62%	59%	66%
Remove stop word	64%	64%	62%	65%
Porter_ Stem	65%	62%	59%	65%
Snowball_ Stem	65%	62%	59%	65%
Lemmatization	65%	62%	58%	65%
Bigram	62%	60%	57%	62%
Trigram	62%	62%	55%	61%

Table 39: Accuracy Analysis Across Different cases Using Tokenizing for English Synonymous ASTD Dataset translated without objective category

Feature	LR	RF	NB	SVM
Tokenizing+ Remove punctuation	64%	62%	59%	66%
Tokenizing+ Remove stop word	64%	64%	62%	65%
Tokenizing+ Porter_ Stem	65%	62%	59%	65%
Tokenizing + Snowball_ Stem	65%	61%	59%	65%
Tokenizing + Lemmatization	65%	62%	58%	65%

Table 40: Accuracy Analysis Across Different cases Using for Bigram and Trigram for English Synonymous ASTD Dataset translated without objective category

Feature	LR	RF	NB	SVM
Bigram + Remove punctuation	62%	61%	57%	62%
Bigram +Remove stop word	63%	61%	60%	63%
Bigram + Porter_ Stem	63%	60%	57%	63%
Bigram +Snowball_ Stem	63%	60%	57%	63%
Bigram + Lemmatization	62%	61%	57%	61%
Trigram + Remove punctuation	62%	61%	55%	63%
Trigram +Remove stop word	64%	62%	58%	63%
Trigram + Porter_ Stem	62%	61%	56%	64%
Trigram + Snowball_ Stem	62%	62%	56%	64%
Trigram + Lemmatization	61%	61%	55%	64%

Table 41: Accuracy Analysis Using lemmatization with two stemmers for English Synonymous ASTD Dataset translated without objective category

Feature	LR	RF	NB	SVM
Lemmatization + Porter_ Stem	65%	63%	59%	65%
Lemmatization + Snowball_ Stem	64%	63%	59%	65%

When the objective and neutral categories were excluded, a notable enhancement in results was observed compared to previous tables. In Table 42, the SVM classifier demonstrated superior performance across most features, achieving an accuracy of 80%. However, the accuracy decreased for all classifiers when the remove stop word feature was applied. Additionally, Table 43 illustrated improved SVM performance when tokenization was combined with either remove punctuation or lemmatization. Furthermore, Table 44 showcased enhanced classifier performance when integrating either Bigram or Trigram features with other preprocessing techniques. However, the accuracy decreased for all classifiers when Trigram combined with remove punctuation. Notably, LR exhibited improved accuracy when lemmatization was combined with the Snowball stemmer, achieving an accuracy of 80%.

Table 42: Accuracy Analysis Across Different cases Using Single Features for English Synonymous ASTD Dataset translated without objective and neutral categories

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	79%	76%	69%	80%
Tokenizing	79%	76%	69%	80%
Remove punctuation	79%	75%	69%	80%
Remove stop word	74%	72%	69%	74%
Porter_ Stem	79%	77%	70%	79%
Snowball_ Stem	79%	76%	70%	79%
Lemmatization	79%	76%	69%	80%
Bigram	77%	75%	67%	77%
Trigram	78%	74%	66%	78%

Table 43: Accuracy Analysis Across Different cases Using Tokenizing for English Synonymous ASTD Dataset translated without objective and neutral categories

Feature	LR	RF	NB	SVM
Tokenizing+ Remove punctuation	79%	75%	69%	80%
Tokenizing+ Remove stop word	79%	78%	74%	79%
Tokenizing+ Porter_ Stem	79%	77%	70%	79%
Tokenizing + Snowball_ Stem	79%	77%	70%	79%
Tokenizing + Lemmatization	79%	76%	69%	80%

Table 44: Accuracy Analysis Across Different cases Using for Bigram and Trigram for English Synonymous ASTD Dataset translated without objective and neutral categories

Feature	LR	RF	NB	SVM
Bigram + Remove punctuation	78%	75%	67%	77%
Bigram +Remove stop word	77%	76%	71%	77%
Bigram + Porter_ Stem	79%	76%	68%	79%
Bigram + Snowball_ Stem	79%	75%	68%	78%
Bigram + Lemmatization	77%	75%	67%	77%
Trigram + Remove punctuation	65%	66%	65%	66%
Trigram +Remove stop word	79%	76%	67%	78%
Trigram + Porter_ Stem	79%	77%	67%	78%
Trigram + Snowball_ Stem	79%	76%	67%	78%
Trigram + Lemmatization	79%	76%	66%	79%

Table 45: Accuracy Analysis Using lemmatization with two stemmers for English Synonymous ASTD Dataset translated without objective and neutral categories

Feature	LR	RF	NB	SVM
Lemmatization + Porter_ Stem	79%	76%	69%	79%
Lemmatization + Snowball_ Stem	80%	77%	70%	79%

In the experiments with synonyms of the translated AJGT dataset, the accuracy results of the classifiers were documented in Tables 46 to 49. Table 46 underscored the remarkable performance of the SVM classifier, boasting an accuracy of 86%, indicative of its proficiency in sentiment analysis tasks. Similarly, Table 47 highlighted the consistent dominance of the SVM classifier across various feature combinations, particularly when tokenization was employed. Conversely, Table 48 revealed fluctuating accuracy levels between the LR and SVM classifiers, suggesting nuanced performance differences based on the selected features. Interestingly, Table 49 showcased the NB classifier's exceptional performance, outperforming other classifiers with an accuracy of 86%. These findings emphasize the significance of classifier selection and feature engineering in optimizing sentiment analysis outcomes.

Table 46: Accuracy Analysis Across Different cases Using Single Features for English Synonymous AJGT Dataset translated

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	83%	84%	86%	86%
Tokenizing	83%	84%	86%	86%
Remove punctuation	83%	84%	86%	86%
Remove stop word	85%	84%	86%	86%
Porter_ Stem	85%	84%	86%	85%
Snowball_ Stem	85%	85%	86%	85%
Lemmatization	86%	82%	86%	86%
Bigram	84%	82%	82%	84%
Trigram	85%	84%	84%	85%

Table 47: Accuracy Analysis Across Different cases Using Tokenizing for English Synonymous AJGT Dataset translated

Feature	**LR**	**RF**	**NB**	**SVM**
Tokenizing+ Remove punctuation	83%	84%	86%	86%
Tokenizing+ Remove stop word	85%	85%	86%	86%

Tokenizing+ Porter_ Stem	85%	83%	86%	85%
Tokenizing + Snowball_ Stem	85%	84%	86%	85%
Tokenizing + Lemmatization	86%	83%	86%	86%

Table 48: Accuracy Analysis Across Different cases Using for Bigram and Trigram for English Synonymous AJGT Dataset translated

Feature	**LR**	**RF**	**NB**	**SVM**
Bigram + Remove punctuation	83%	81%	82%	84%
Bigram +Remove stop word	83%	82%	82%	82%
Bigram + Porter_ Stem	85%	82%	82%	84%
Bigram +Snowball_ Stem	84%	84%	84%	84%
Bigram + Lemmatization	83%	81%	82%	84%
Trigram + Remove punctuation	85%	84%	84%	85%
Trigram +Remove stop word	81%	81%	82%	83%
Trigram + Porter_ Stem	86%	84%	85%	86%
Trigram + Snowball_ Stem	86%	83%	85%	86%
Trigram + Lemmatization	85%	83%	84%	85%

Table 49: Accuracy Analysis Using lemmatization with two stemmers for English Synonymous AJGT Dataset translated

Feature	**LR**	**RF**	**NB**	**SVM**
Lemmatization + Porter_ Stem	85%	84%	86%	85%
Lemmatization + Snowball_ Stem	85%	84%	86%	85%

4.4 Experiments – Fourth Stage

In the final stage of our experiments, we translated the synonyms generated from the English dataset back into Arabic and evaluated the accuracy of the classifiers. This allowed us to compare the performance of the classifiers using the translated synonyms with those using the

original Arabic dataset. Through this process, we aimed to assess the effectiveness of using synonyms in sentiment analysis tasks and ascertain whether the translated synonyms could yield comparable results to those obtained with the original Arabic dataset.

Tables 50 to 53 present the outcomes of our experiments involving the translation of synonyms from the ASTD dataset back into Arabic. These results indicate that there was no significant improvement in accuracy compared to using the original Arabic dataset. However, it's noteworthy that both LR and SVM classifiers consistently exhibited better performance compared to other classifiers across various individual and combined features. The SVM classifier demonstrated the highest accuracy of 70% when combining Lemmatization with ArListem stemmer, as depicted in Table 53.

Table 50: Accuracy Analysis Across Different cases Using Single Features for Synonymous ASTD Dataset translated to Arabic

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	68%	68%	67%	68%
Tokenizing	68%	68%	67%	68%
Remove punctuation	68%	67%	67%	68%
Remove stop word	68%	67%	67%	68%
ISRI_ Stem	69%	67%	67%	69%
ArListem_ Stem	68%	67%	67%	69%
Snowball_ Stem	69%	68%	67%	69%
Lemmatization	69%	68%	67%	69%
Bigram	67%	67%	67%	67%
Trigram	67%	68%	67%	68%

Table 51: Accuracy Analysis Across Different cases Using Tokenizing for Synonymous ASTD Dataset translated to Arabic

Feature	**LR**	**RF**	**NB**	**SVM**

Tokenizing+ Remove punctuation	68%	68%	67%	68%
Tokenizing+ Remove stop word	68%	68%	67%	68%
Tokenizing+ ISRI_ Stem	69%	68%	67%	69%
Tokenizing+ ArListem_ Stem	68%	68%	67%	69%
Tokenizing + Snowball_ Stem	69%	68%	67%	69%
Tokenizing + Lemmatization	69%	68%	67%	69%

Table 52: Accuracy Analysis Across Different cases Using Bigram and Trigram for Synonymous ASTD Dataset translated to Arabic

Feature	**LR**	**RF**	**NB**	**SVM**
Bigram + Remove punctuation	67%	67%	67%	67%
Bigram +Remove stop word	68%	67%	67%	68%
Bigram + ISRI_ Stem	69%	68%	67%	69%
Bigram + ArListem_ Stem	68%	67%	67%	68%
Bigram + Snowball_ Stem	68%	67%	67%	69%
Bigram + Lemmatization	68%	67%	67%	68%
Trigram + Remove punctuation	68%	68%	67%	68%
Trigram +Remove stop word	68%	68%	67%	68%
Trigram + ISRI_ Stem	69%	67%	67%	69%
Trigram + ArListem_ Stem	68%	67%	67%	68%
Trigram + Snowball_ Stem	68%	68%	67%	69%
Trigram + Lemmatization	68%	68%	67%	69%

Table 53: Accuracy Analysis Using lemmatization with three stemmers for Synonymous ASTD Dataset translated to Arabic

Feature	**LR**	**RF**	**NB**	**SVM**
Lemmatization + ISRI_ Stem	68%	67%	67%	69%
Lemmatization + ArListem_ Stem	68%	68%	67%	70%
Lemmatization + Snowball_ Stem	68%	68%	67%	69%

The results of the synonym experiments conducted on the ASTD dataset, excluding the objective category and translated into Arabic, are presented in Tables 54 to 57. These tables reveal a decrease in classifier accuracy compared to the original Arabic data. Specifically, Table 54 illustrates that the LR and SVM classifiers achieved their highest accuracy of 61% when utilizing the ISRI stemmer feature. Similarly, Table 55 demonstrates that combining Tokenization with either the Remove stop word or ISRI stemmer feature yielded an accuracy of 61% across LR, RF, and SVM classifiers. In Table 56, the RF classifier outperformed others, achieving 61% accuracy when combining Trigram with ArListem stemmer. Additionally, Table 57 indicates that combining Lemmatization with ArListem stemmer resulted in 61% accuracy across LR, RF, and SVM classifiers.

Table 54: Accuracy Analysis Across Different cases Using Single Features for Synonymous ASTD Dataset translated to Arabic without objective category

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	58%	60%	56%	58%
Tokenizing	58%	59%	56%	58%
Remove punctuation	58%	59%	56%	59%
Remove stop word	58%	58%	57%	57%
ISRI_ Stem	61%	59%	58%	61%
ArListem_ Stem	60%	60%	57%	59%
Snowball_ Stem	61%	60%	57%	60%
Lemmatization	61%	60%	58%	60%
Bigram	57%	58%	56%	57%
Trigram	57%	59%	56%	59%

Table 55: Accuracy Analysis Across Different cases Using Tokenizing for Synonymous ASTD Dataset translated to Arabic without objective category

Feature	LR	RF	NB	SVM
Tokenizing+ Remove punctuation	58%	59%	56%	59%
Tokenizing+ Remove stop word	61%	61%	58%	61%
Tokenizing+ ISRI_ Stem	61%	61%	58%	61%
Tokenizing+ ArListem_ Stem	60%	59%	57%	59%
Tokenizing + Snowball_ Stem	61%	60%	57%	60%
Tokenizing + Lemmatization _	61%	60%	58%	60%

Table 56: Accuracy Analysis Across Different cases Using Bigram and Trigram for Synonymous ASTD Dataset translated to Arabic without objective category

Feature	LR	RF	NB	SVM
Bigram + Remove punctuation	57%	58%	56%	56%
Bigram +Remove stop word	57%	54%	56%	57%
Bigram +ISRI_ Stem	60%	60%	57%	60%
Bigram + ArListem_ Stem	59%	59%	56%	58%
Bigram + Snowball_ Stem	59%	59%	56%	60%
Bigram + Lemmatization	60%	59%	57%	60%
Trigram + Remove punctuation	56%	58%	56%	58%
Trigram +Remove stop word	57%	59%	56%	56%
Trigram + ISRI_ Stem	59%	60%	56%	60%
Trigram + ArListem_ Stem	59%	61%	56%	59%
Trigram + Snowball_ Stem	59%	59%	56%	59%
Trigram + Lemmatization	59%	59%	56%	59%

Table 57: Accuracy Analysis Using lemmatization with three stemmers for Synonymous ASTD Dataset translated to Arabic without objective category

Feature	LR	RF	NB	SVM
Lemmatization + ISRI_ Stem	61%	61%	58%	59%

Lemmatization + ArListem_ Stem	61%	61%	58%	61%
Lemmatization + Snowball_ Stem	60%	61%	58%	59%

Tables 58 to 61 present the outcomes of experiments involving synonyms on the ASTD dataset, excluding the objective and neutral categories, translated into Arabic. Table 58 reflects a mixture of performance improvements and declines across various features. Notably, the LR and SVM classifiers achieved their highest accuracy of 77% when employing the Snowball stemmer. Similarly, Table 59 showcases the superior performance of the LR and SVM classifiers, reaching 77% accuracy when combining Tokenization with the Snowball stemmer. However, the accuracy varied among LR and SVM classifiers in Table 60 across different features. Lastly, Table 61 demonstrates that the SVM classifier exhibited the best performance, achieving 77% accuracy when combining Lemmatization with the ArListem stemmer.

Table 58: Accuracy Analysis Across Different cases Using Single Features for Synonymous ASTD Dataset translated to Arabic without objective and neutral categories

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	74%	71%	68%	74%
Tokenizing	74%	72%	68%	74%
Remove punctuation	74%	71%	68%	74%
Remove stop word	74%	72%	69%	74%
ISRI_ Stem	75%	74%	69%	75%
ArListem_ Stem	74%	73%	68%	75%
Snowball_ Stem	77%	74%	69%	77%
Lemmatization	76%	73%	70%	77%
Bigram	74%	70%	67%	74%
Trigram	72%	71%	66%	73%

Table 59: Accuracy Analysis Across Different cases Using Tokenizing for Synonymous ASTD Dataset translated to Arabic without objective and neutral categories

Feature	LR	RF	NB	SVM
Tokenizing+ Remove punctuation	74%	71%	68%	74%
Tokenizing+ Remove stop word	74%	72%	69%	74%
Tokenizing+ ISRI_ Stem	75%	74%	69%	75%
Tokenizing+ ArListem_ Stem	74%	73%	68%	75%
Tokenizing + Snowball_ Stem	77%	74%	69%	77%
Tokenizing + Lemmatization	76%	74%	70%	77%

Table 60: Accuracy Analysis Across Different cases Using Bigram and Trigram for Synonymous ASTD Dataset translated to Arabic without objective and neutral categories

Feature	LR	RF	NB	SVM
Bigram + Remove punctuation	74%	71%	67%	74%
Bigram +Remove stop word	75%	73%	67%	74%
Bigram + ISRI_ Stem	74%	73%	68%	75%
Bigram + ArListem_ Stem	74%	73%	67%	74%
Bigram + Snowball_ Stem	75%	72%	68%	74%
Bigram + Lemmatization	76%	73%	68%	75%
Trigram + Remove punctuation	73%	70%	66%	74%
Trigram +Remove stop word	74%	72%	67%	74%
Trigram + ISRI_ Stem	74%	74%	66%	75%
Trigram + Stem_ ArListem	75%	74%	66%	75%
Trigram + Stem_ Snowball	76%	72%	66%	76%
Trigram + Lemmatization	76%	73%	66%	76%

Table 61: Accuracy Analysis Using lemmatization with three stemmers for Synonymous ASTD Dataset translated to Arabic without objective and neutral categories

Feature	LR	RF	NB	SVM
Lemmatization + ISRI_ Stem	75%	75%	70%	76%

Lemmatization + ArListem_ Stem	75%	74%	69%	77%
Lemmatization + Snowball_ Stem	76%	73%	69%	75%

In the experiments involving the translation of synonyms of the AJGT dataset into Arabic, Tables 62 to 65 reveal a notable decrease in the classifiers' performance. However, despite this decline, the NB classifier exhibited the best performance across several experiments. In Table 62, the NB classifier achieved the highest accuracy of 83% when utilizing the Lemmatization feature. Similarly, in Table 63, the NB classifier maintained its superior performance, reaching 83% accuracy when combining Lemmatization with Tokenization. Notably, the accuracy varied between the NB and SVM classifiers in Table 64, with the NB and SVM classifiers achieving the best accuracy of 82% depending on the features used. Finally, in Table 65, both the RF and NB classifiers demonstrated the best performance, achieving an accuracy of 82% when combining Lemmatization with the ArListem stemmer.

Table 62: Accuracy Analysis Across Different cases Using Single Features for Synonymous AJGT Dataset translated to Arabic

Feature	**LR**	**RF**	**NB**	**SVM**
Without any feature	78%	76%	79%	76%
Tokenizing	77%	76%	79%	76%
Remove punctuation	77%	75%	79%	77%
Remove stop word	77%	75%	77%	78%
ISRI_ Stem	80%	78%	79%	80%
ArListem_ Stem	80%	79%	81%	78%
Snowball_ Stem	79%	79%	82%	79%
Lemmatization	81%	81%	83%	81%
Bigram	74%	73%	77%	76%
Trigram	77%	73%	76%	77%

Table 63: Accuracy Analysis Across Different cases Using Tokenizing for Synonymous AJGT Dataset translated to Arabic

Feature	LR	RF	NB	SVM
Tokenizing+ Remove punctuation	77%	76%	79%	77%
Tokenizing+ Remove stop word	77%	74%	77%	78%
Tokenizing+ ISRI_ Stem	80%	79%	79%	80%
Tokenizing+ ArListem_ Stem	80%	80%	81%	78%
Tokenizing + Snowball_ Stem	79%	79%	82%	79%
Tokenizing + Lemmatization	81%	82%	83%	81%

Table 64: Accuracy Analysis Across Different cases Using Bigram and Trigram for Synonymous AJGT Dataset translated to Arabic

Feature	LR	RF	NB	SVM
Bigram + Remove punctuation	74%	73%	77%	75%
Bigram +Remove stop word	77%	73%	78%	76%
Bigram + ISRI_ Stem	79%	77%	80%	79%
Bigram + ArListem_ Stem	78%	75%	79%	78%
Bigram + Snowball_ Stem	80%	79%	81%	82%
Bigram + Lemmatization	80%	79%	81%	79%
Trigram + Remove punctuation	77%	72%	76%	77%
Trigram +Remove stop word	76%	72%	76%	74%
Trigram + ISRI_ Stem	79%	76%	79%	80%
Trigram + ArListem_ Stem	80%	79%	81%	79%
Trigram + Snowball_ Stem	80%	76%	80%	80%
Trigram + Lemmatization	81%	79%	82%	81%

Table 65: Accuracy Analysis Using lemmatization with three stemmers for Synonymous ASTD Dataset translated to Arabic

Feature	LR	RF	NB	SVM
Lemmatization + ISRI_ Stem	80%	81%	81%	80%

Lemmatization + ArListem_ Stem	81%	82%	82%	81%
Lemmatization + Snowball_ Stem	82%	81%	82%	81%

The aim of the experiments was to assess the impact of various preprocessing steps on sentiment analysis using the ASTD dataset. Particularly the SVM classifier emerged as the top performer. Interestingly, when combining two features simultaneously, the results remained consistent, with accuracy levels ranging between 67% and 70%. This suggests that the choice of preprocessing techniques can significantly influence sentiment analysis accuracy, with SVM demonstrating robust performance across various scenarios, Figure 5 shows the effect of preprocessing steps on the four classifiers.

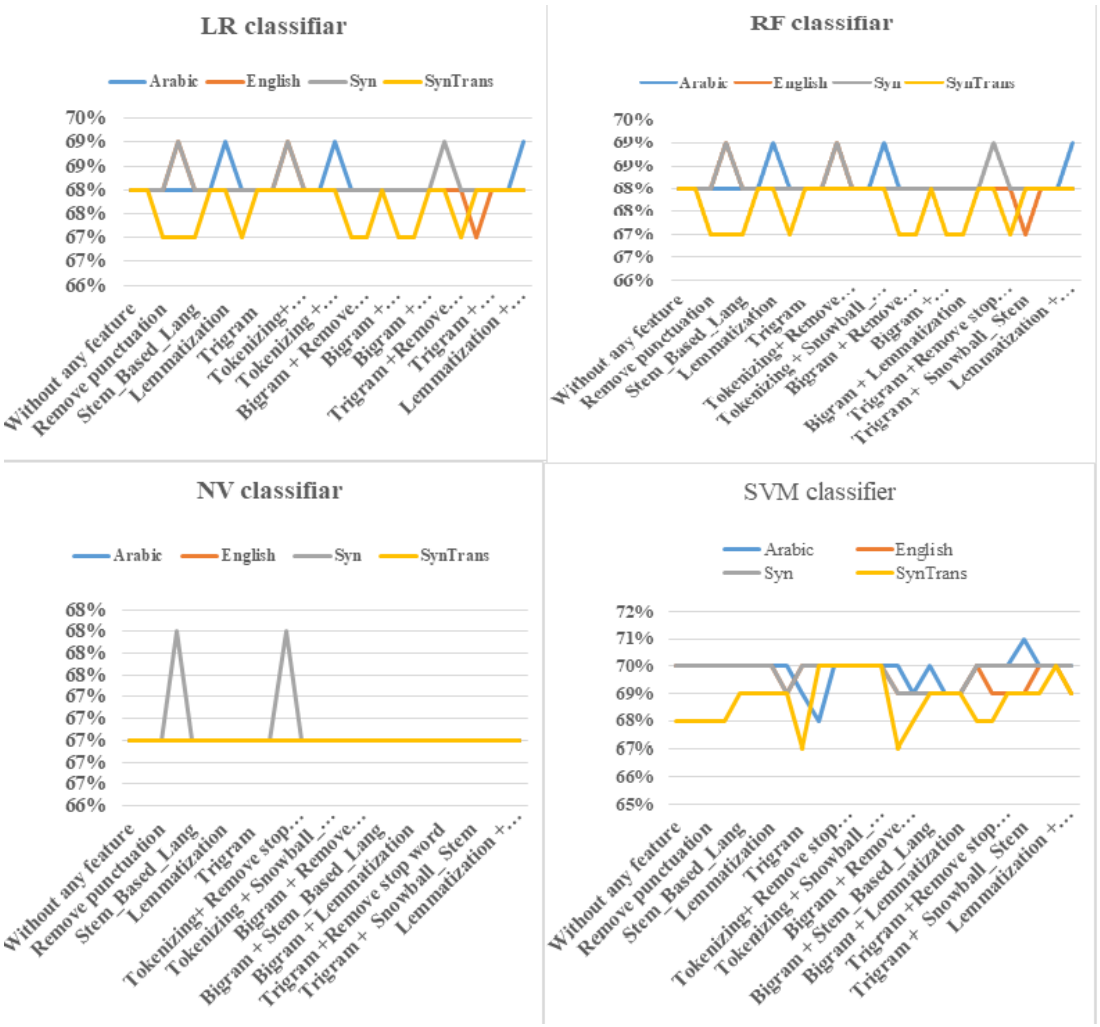


Figure 5:the effect of the preprocessing steps on the four classifiers for ASTD Dataset

Additionally, the experiments involving the removal of the objective category from the ASTD dataset exhibited a notable decrease in accuracy compared to both the original Arabic dataset and the translated version, as depicted in the preceding tables and figure 6. Conversely, there was an observed enhancement in accuracy results upon removing sentences classified into the objective and neutral categories, as shown in Figure 7. This suggests that the removal of such categories may have a positive impact on sentiment analysis accuracy, potentially by reducing noise or irrelevant data in the dataset.

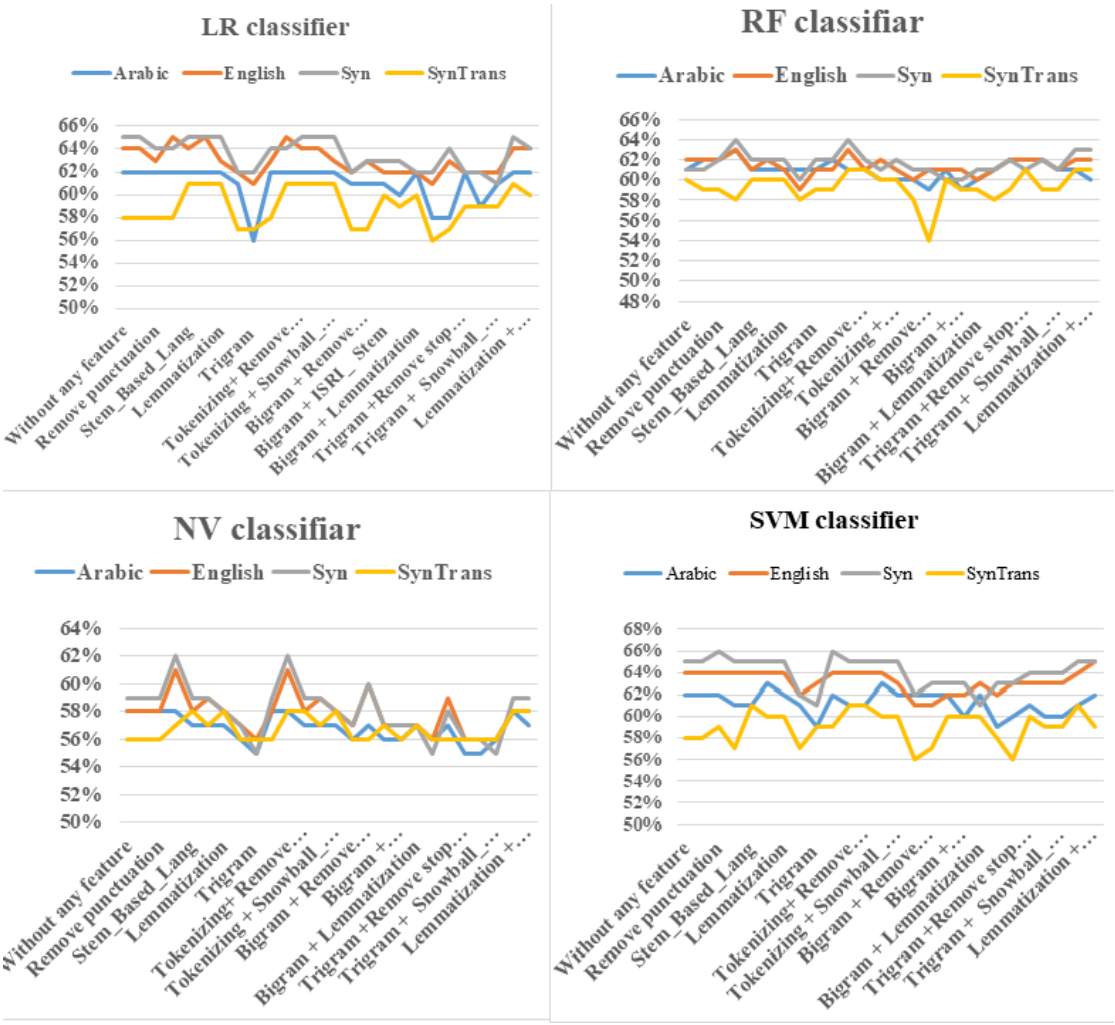


Figure 6:The effect of preprocessing steps on the four classifiers for ASTD Dataset without Objective category.

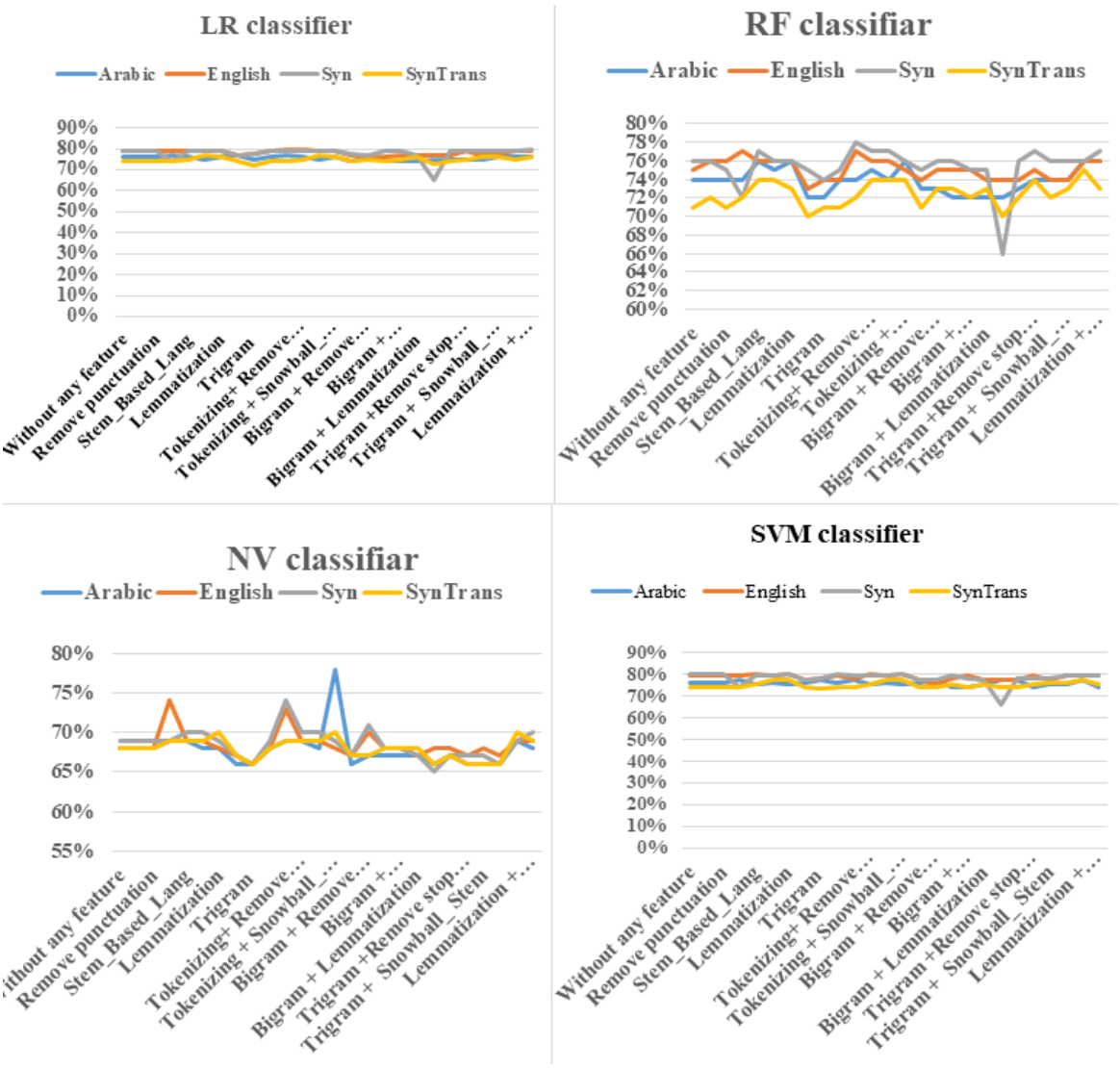


Figure 7: The effect of preprocessing steps on the four classifiers for ASTD Dataset without Objective and Neutral categories.

Similar experiments were conducted using the AJGT dataset, yielding accuracy results ranging between 82% and 87%. Interestingly, the results showed convergence among the classifiers, with the NB classifier achieving the highest accuracy. This indicates the effectiveness of the NB classifier in accurately predicting sentiment in the context of the AJGT dataset, this is shown in Figure 8.

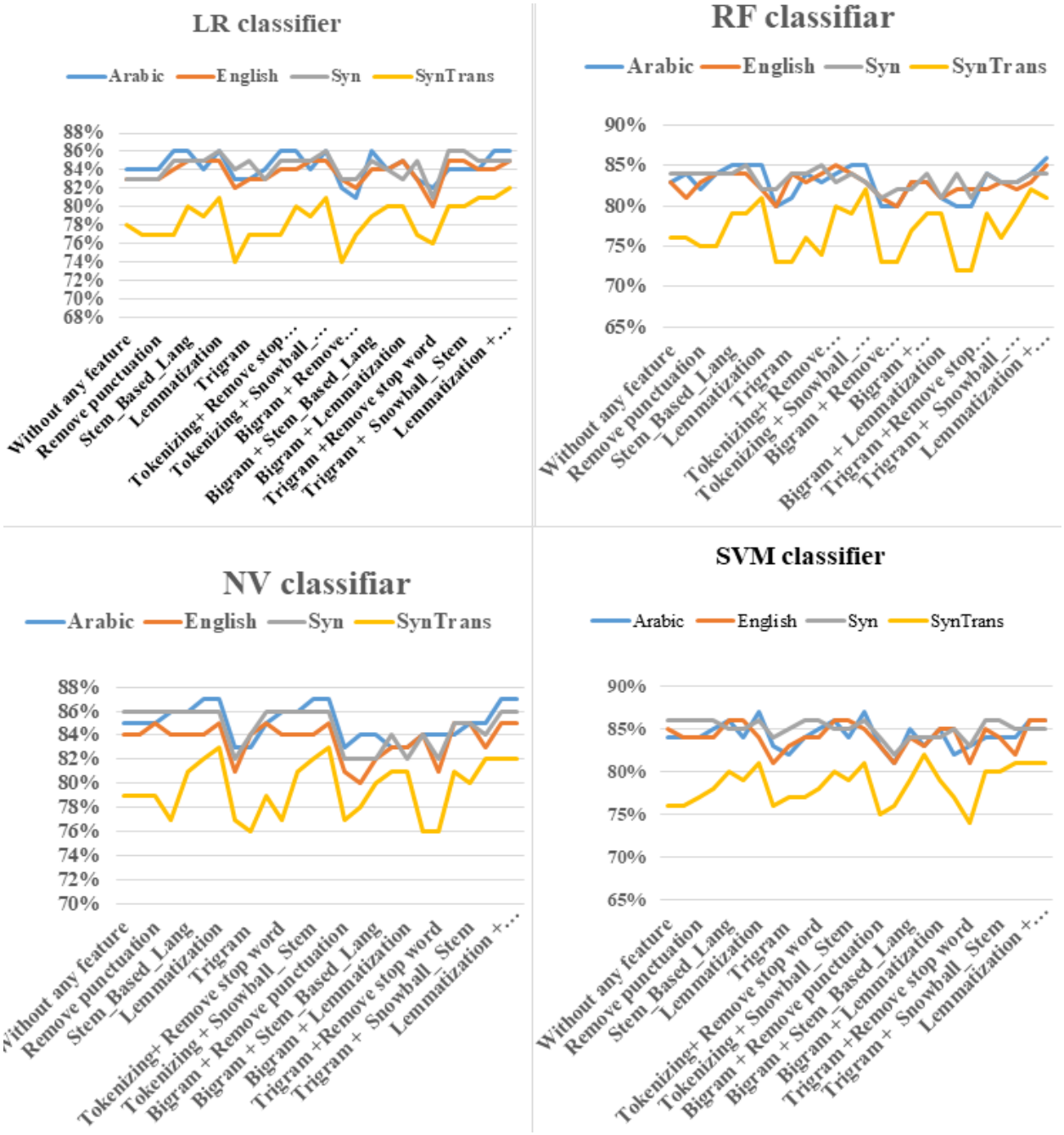


Figure 8: The effect of preprocessing steps on the four classifiers for AJGT Dataset.

Table 66 showcases the culmination of all experiments conducted, highlighting the performance of the four classifiers across different scenarios. Across these experiments, the SVM classifier consistently demonstrated superior performance compared to the other

classifiers. Notably, in the experiment involving the AJGT dataset after translating English language synonyms into Arabic, the NB classifier notably achieved the highest accuracy result, closely matching the performance of the SVM classifier. These findings underscore the robustness of the SVM classifier in sentiment analysis tasks, while also indicating the potential efficacy of the NB classifier in specific contexts, such as when dealing with translated datasets.

Table 66: Summary of Accuracy Results from Previous Experiments Using Classifiers

Dataset / Classifier	**LR**	**RF**	**NB**	**SVM**
ASTD AR	70%	69%	67%	71%
ASTD translate to Eng	70%	69%	67%	70%
ASTD Eng synonyms	70%	69%	68%	70%
ASTD synonyms translate to AR	69%	68%	67%	70%
ASTD (wo) AR	62%	63%	58%	63%
ASTD (wo) translate to Eng	65%	63%	61%	65%
ASTD (wo) Eng synonyms	65%	64%	62%	66%
ASTD (wo) synonyms translate to AR	61%	61%	58%	61%
ASTD (won) AR	77%	76%	69%	77%
ASTD (won) translate to Eng	80%	77%	74%	80%
ASTD (won) Eng synonyms	80%	78%	74%	80%
ASTD (won) synonyms translate to AR	77%	75%	70%	77%
AJGT AR	86%	86%	87%	87%
AJGT translate to Eng	85%	85%	85%	86%
AJGT Eng synonyms	86%	85%	86%	86%
AJGT synonyms translate to AR	82%	82%	83%	82%

(wo) refers to without objective categories
(won) objective and neutral categories

we can conclude from the above that SVM classifier performance tends to be superior in most experiments when combining two key features, particularly Lemmatization and Stemming. The combination of these features improves the ability of the SVM classifier to distinguish

between different classes more effectively. The improvement in performance can be attributed to SVM's effectiveness in finding the optimal hyperplane that separates different classes when the data is well-preprocessed and normalized.

Based on the research conducted, here are the answers to the research questions and the conclusions regarding the hypotheses:

Research Questions:

1. **How accurate is sentiment analysis in texts in both Arabic and English?**

   The accuracy of sentiment analysis varies between Arabic and English texts. Generally, sentiment analysis models tend to perform better on English texts due to the abundance of resources, tools, and pre-trained models available for the English language. However, in the context of Arabic texts, especially when dealing with colloquial dialects, accuracy tends to be lower. This is largely due to the complexity of Arabic language structures, including diverse dialects and the lack of extensive annotated datasets.

2. **What factors affect the performance of sentiment analysis when used in Arabic texts?**

   - Dialectal Variations: The existence of numerous Arabic dialects adds complexity to sentiment analysis, as different regions use different expressions and structures.
   - Linguistic Complexity: Arabic’s rich morphology, including roots, prefixes, and suffixes, makes it challenging for standard NLP models.

- Quality of Datasets: The limited availability and quality of annotated datasets in Arabic hinder the development of accurate models.
- Translation Issues: When translating Arabic to English for sentiment analysis, nuances and cultural meanings can be lost, reducing model effectiveness.

3. **What are the machine learning techniques used for Arabic Language Sentiment Analysis?**

   Several machine learning techniques are utilized for Arabic sentiment analysis, including:

   - Traditional Methods: Naive Bayes, Support Vector Machines (SVM), and Logistic Regression are used for their simplicity and effectiveness in handling text classification tasks.
   - Deep Learning Methods: Recurrent Neural Networks (RNNs), Long Short-Term Memory Networks (LSTMs), and Convolutional Neural Networks (CNNs) are employed to capture complex patterns and contextual information in Arabic text.
   - Transformers and BERT-based Models: Transformer architectures, including BERT (Bidirectional Encoder Representations from Transformers) and its Arabic variant AraBERT, are used for their superior performance in understanding context and meaning in Arabic texts.

4. **How efficient is the reuse of existing English-based resources in processing Arabic natural languages?**

   The reuse of English-based resources, such as pre-trained models and lexicons, can be beneficial but also comes with limitations. While these resources provide a starting

point, the translation from Arabic to English often leads to loss of nuance, which can degrade the accuracy of sentiment analysis. Additionally, English models may not fully capture the linguistic and cultural intricacies of Arabic, necessitating adaptations or the development of new resources specifically for Arabic.

Hypotheses:

- **First Hypothesis**: The utilization of multiple manually grouped datasets in Arabic significantly impacts the quality of existing sentiment prediction techniques.

  Conclusion: This hypothesis was validated by the research. Manually curated datasets improved the quality of sentiment analysis models for Arabic by ensuring that the data more accurately reflects the linguistic and cultural nuances of the language. Grouping datasets by dialect, context, or domain allowed for more precise modeling and better performance, highlighting the importance of dataset quality in sentiment analysis.

- **Second Hypothesis**: Expanding techniques for the determination of polarity will significantly influence the overall quality of the improved sentiment analyzer.

  Conclusion: The research supported this hypothesis. Expanding and refining polarity detection techniques, particularly those tailored to handle the grammatical and morphological complexity of Arabic, led to noticeable improvements in sentiment analysis accuracy. using machine learning models, and specifically those designed to address Arabic's unique features, proved essential in enhancing the sentiment analysis process. This underscores the need for specialized approaches when dealing with languages like Arabic.

Overall, the research confirms that while sentiment analysis for Arabic texts presents unique challenges, focused efforts on dataset curation, advanced modeling techniques, and the careful adaptation of existing resources can lead to significant improvements in accuracy and effectiveness.

4.1 Comparison between Classifiers based on dataset

In this section, we delve into previous studies that have utilized the ASTD dataset, employing various classifiers to gauge sentiment analysis accuracy. These studies serve as valuable benchmarks for understanding the performance of sentiment analysis models on Arabic text data.

Researchers have explored the ASTD dataset using a range of classifiers. By employing the classifiers, they aimed to discern the sentiment expressed in Arabic text and evaluate the effectiveness of different machine learning and deep learning approaches in sentiment analysis tasks.

In (Elfaik and Nfaoui 2020) , researchers explored sentiment analysis on Arabic text, leveraging various datasets, notably the ASTD dataset. They employed a range of machine learning algorithms, including NB, SVM, LR, as well as convolutional deep learning methods like CNN. In their experiments, they utilized diverse features such as TF, TF-IDF, POS tagging, Lexicon-based features, and Automatic Lexicon generation. Additionally, they incorporated the Word2Vec model to capture semantic relationships between words. Their findings indicated promising accuracy results, ranging from 85% to 90% when analyzing the entire dataset. Notably, accuracy improved further to 95% when focusing on a subset that constituted 85% of the main datasets.

In (Baly, Badaro et al. 2017), researchers conducted sentiment analysis using SVM and RNTN algorithms, focusing on three distinct Arabic dialects: Egyptian, Gulf, and Levantine. Their objective was to compare the performance of these algorithms across different dialects. To facilitate sentiment analysis, researchers employed various identifying features within both algorithms, including n-grams and lemma, in addition to baseline and raw word features. By incorporating these features, they aimed to capture the nuanced characteristics of each Arabic dialect and enhance sentiment analysis accuracy. The study reported differing accuracy levels for the SVM and RNTN algorithms across the three Arabic dialects. The SVM algorithm achieved an accuracy rate of 51.7% using all lemma features, whereas the RNTN algorithm outperformed with a higher accuracy rate of 58.5%.

In (Heikal, Torki et al. 2018), researchers explored sentiment analysis using CNN and LSTM algorithms. They focused on determining optimal parameter values for each model and utilized the Word2Vec technique to represent pre-trained words, enhancing the models' ability to capture semantic meaning. The study reported notable accuracy results for both the CNN and LSTM algorithms. Specifically, the experiments with the CNN algorithm, after adjusting coefficient values, achieved an accuracy rate of 64.30%. Similarly, the LSTM algorithm yielded promising results, with the highest accuracy recorded at 64.75%. Furthermore, researchers employed ensemble modeling, combining the strengths of both CNN and LSTM algorithms. This approach resulted in even higher accuracy, with the ensemble model achieving an impressive accuracy rate of 65.05%. Comparing these results to those obtained from the RNTN model in previous studies, the CNN and LSTM algorithms, especially when used in combination, demonstrated improvements in accuracy.

In (Dahou 2016), researchers delved into the application of the CNN algorithm alongside a word embedding model to represent pre-trained words as vectors. They evaluated the quality of these vectors by analyzing word analogy questions, aiming to determine relationships between word pairs and predict missing word identities through algebraic arithmetic, often employing similarity measures like the cosine measure. Their experiments focused on precision assessment, particularly in the context of unbalanced and balanced trained datasets. Surprisingly, the CNN algorithm demonstrated higher precision rates when applied to the unbalanced dataset, achieving an impressive precision rate of 79.07%. In contrast, the balanced dataset yielded slightly lower precision results, with an accuracy of 75.9%.

In (Al-Azani and El-Alfy 2017), researchers explored the efficacy of CNN and LSTM algorithms, alongside various combined models derived from these two algorithms. These combined models included CNN-LSTM, Stacked-LSTM, Combined-LSTM-SUM, Combined-LSTM-MUL, and Combined-LSTM-CONC. The study conducted several experiments, leveraging two different models: CBOW and SG to represent words, thereby comparing results for static and dynamic words.

Among the experimented models, the Combined-LSTM-MUL model, utilizing dynamic words and CBOW word embeddings, yielded the highest accuracy rate at 81.63%. Additionally, with the SG model, the Combined-LSTM-CONC model achieved the best accuracy at 80.42% for dynamic words. These findings underscore the significance of word embedding techniques and dynamic word representations in enhancing sentiment analysis accuracy, as evidenced by the superior performance of these combined models.

In (Alayba, Palade et al. 2018), researchers employed a combination of CNN and LSTM algorithms, conducting experiments structured into three levels to extract various features from short sentences. The first level operated at the character level, where each word in the sentence was converted into characters, resulting in a plethora of features. Additionally, the second level focused on character n-gram analysis (char5Gram), aiming to measure the average word length in the dataset. Typically, most words averaged five characters, while words exceeding this length were segmented into sub-words. Finally, the third level operated at the word level, dividing the sentence into individual words based on the spaces between them.

The accuracy results obtained for the ASTD dataset, consisting of 2,479 tweets (1,684 negative and 795 positive), were as follows: 74.19% at the character level, 77.62% at the char5Gram level, and 76.41% at the word level. Remarkably, the highest accuracy was achieved using the char5Gram level, highlighting its effectiveness in sentiment analysis tasks.

In (Hawalah 2019), researchers explored various N-gram features, including Unigram, Bigram, and Trigram, both individually and in combination, to evaluate the accuracy of machine learning algorithms such as SVM, NB, LR, Linear SVM, RBF, and MLP. The experiments aimed to identify the most effective feature combinations for sentiment analysis tasks.

The findings revealed that the Unigram feature yielded the best results in terms of accuracy across the tested models. Notably, when combining Unigram with Bigram features in the MLP model, a notable accuracy of 75.47% was achieved, demonstrating the effectiveness of incorporating multiple features for improved performance. However, experiments with the Trigram feature yielded lower accuracy, attributed to the presence of noise that adversely

impacted performance. Specifically, the accuracy dropped to 68.41% when using the Trigram feature in the same MLP model

The studies have reported diverse accuracy levels across different classifiers. While some classifiers may exhibit higher accuracy rates, others may perform better in specific scenarios or with certain feature sets. Understanding these nuances provides valuable insights into the strengths and limitations of different classifiers when applied to sentiment analysis on Arabic text data.

By synthesizing the findings of these studies, researchers can gain a comprehensive understanding of the performance metrics associated with sentiment analysis models utilizing the ASTD dataset. This knowledge lays the groundwork for developing more robust and accurate sentiment analysis solutions tailored to Arabic language text data.

4.2 Summary

In this study, we conducted a series of experiments to evaluate sentiment analysis models on Arabic text datasets. We explored various pre-processing techniques, including tokenization, stop word removal, stemming, and lemmatization, and assessed their impact on sentiment analysis accuracy using machine learning classifiers such as LR, RF, NB, and SVM.

Our experiments involved multiple stages, including translating Arabic datasets into English, using English synonyms, and translating them back into Arabic. We analyzed the accuracy of classifiers at each stage to determine the effectiveness of different preprocessing techniques and language translations.

Overall, our findings revealed that SVM consistently outperformed other classifiers across various preprocessing techniques and dataset translations. Additionally, the NB classifier demonstrated notable performance, particularly in experiments involving English synonym translations. We observed varying degrees of accuracy improvement or decline based on the dataset, preprocessing techniques, and classifier used.

It is worth noting that the SVM classifier demonstrated superior performance compared to other classifiers in almost all experiments, particularly when combining two features. This was especially evident when the combination included Lemmatization and Stemming.

Furthermore, our experiments highlighted the importance of dataset preprocessing and language translation in sentiment analysis accuracy. While some techniques and translations resulted in improved accuracy, others showed no significant difference or even a decline. This underscores the complexity of sentiment analysis tasks and the need for careful consideration of preprocessing techniques and dataset characteristics to achieve optimal results.

CHAPTER 5:

CONCLUSIONS

AND

FUTURE WORK

Chapter 5 - Conclusions and Future Work

5.1 Conclusions and Future Work

In response to the burgeoning use of social media platforms and the prominence of Arabic as a widely utilized language online, our study delved into the realm of sentiment analysis, particularly focusing on Arabic datasets. Given the limited availability of resources in this domain and the relatively few studies dedicated to Arabic sentiment analysis, our research aimed to fill this gap by conducting a comprehensive investigation. We explored the efficacy of different pre-processing techniques and machine learning algorithms to ascertain their effectiveness in analyzing sentiment in Arabic text.

Our proposed sentiment analysis model for the Arabic language operates through four distinct stages based Arabic and English, of which designed to leverage English language resources and techniques to assess their influence on classifier accuracy. This innovative approach allows us to exploit the wealth of resources available in English, potentially enhancing the performance of sentiment analysis classifiers when applied to Arabic text.

First and foremost, our findings underscored the remarkable performance of the Support Vector Machine classifier across different preprocessing methods and datasets. Its consistent superiority over other classifiers highlights its suitability for sentiment analysis tasks in Arabic, reaffirming its status as a robust and reliable choice for such tasks.

Moreover, we observed the pivotal role played by preprocessing techniques such as tokenization, stemming, and lemmatization in enhancing sentiment analysis accuracy. By combining these techniques with feature extraction methods like N-grams, we witnessed

tangible improvements in performance, indicating the significance of meticulous data preprocessing in sentiment analysis workflows.

Despite our efforts to explore translation and synonyms as potential avenues for enhancing accuracy, our experiments revealed only marginal improvements compared to the original Arabic datasets. While these findings underscore the importance of retaining the integrity of the original language in sentiment analysis tasks, they also highlight the need for further research to uncover more effective strategies for leveraging multilingual resources.

Nevertheless, our study also illuminated certain challenges, such as dataset imbalance and noise in specific features like Trigrams. Addressing these challenges could pave the way for more robust sentiment analysis models tailored to Arabic text, offering greater accuracy and reliability in real-world applications.

In future work, we plan to prioritize optimizing accuracy by expanding the dataset with additional Arabic text and exploring advanced deep learning classifiers, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) networks. These models have demonstrated considerable success in sentiment analysis tasks due to their ability to capture complex patterns and contextual relationships within the data, which could significantly enhance the performance of our sentiment analysis model for Arabic text.

Furthermore, we will ensure that the text is meticulously translated from Arabic to English, allowing us to effectively utilize the extensive resources and pre-trained models available in English. This step is essential for improving the accuracy of sentiment classification, as it

enables the model to benefit from the rich linguistic data and sophisticated techniques developed for the English language.

In addition to these strategies, we will explore the use of an Arabic limiter—a tool designed to focus on and refine specific linguistic features unique to Arabic. This approach could help improve the quality of classifiers by enhancing their ability to accurately capture and predict emotions in the Arabic language. By addressing these distinct challenges posed by Arabic texts, we aim to refine our sentiment analysis techniques and contribute meaningfully to the advancement of natural language processing, particularly in multilingual and culturally diverse contexts.

References

A. Al Shamsi, A., R. Bayari and S. Salloum (2021). "Sentiment Analysis in English Texts." Advances in Science Technology and Engineering Systems Journal **5**: 1683-1689.

Abainia, K., S. Ouamour and H. Sayoud (2016). "A novel robust Arabic light stemmer." Journal of Experimental & Theoretical Artificial Intelligence: 1-17.

Abdullah, M. and M. Hadzikadic (2017). Sentiment Analysis on Arabic Tweets: Challenges to Dissecting the Language.

Abo, M. E. M., N. Idris, R. Mahmud, A. Qazi, I. A. T. Hashem, J. Z. Maitama, U. Naseem, S. K. Khan and S. Yang (2021). "A Multi-Criteria Approach for Arabic Dialect Sentiment Analysis for Online Reviews: Exploiting Optimal Machine Learning Algorithm Selection." Sustainability **13**(18): 10018.

Abo, M., N. Shah, V. Balakrishnan and A. Abdelaziz (2018). "Sentiment analysis algorithms: evaluation performance of the Arabic and English language." IEEE Expert.

Abu Farha, I. and W. Magdy (2021). "A comparative study of effective approaches for Arabic sentiment analysis." Information Processing & Management **58(**2): 102438.

Al-Amrani, Y., M. Lazaar and k. e. el kadiri (2018). "Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis." Procedia Computer Science **127**: 511-520.

Alayba, A., V. Palade, M. England and R. Iqbal (2018). A Combined CNN and LSTM Model for Arabic Sentiment Analysis: Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Hamburg, Germany, August 27–30, 2018, Proceedings**:** 179-191.

Al-Azani, S. and E.-S. El-Alfy (2017). Hybrid Deep Learning for Sentiment Polarity Determination of Arabic Microblogs.

Al-Batah, M., S. Mrayyen and M. Alzaqebah (2019). "Arabic Sentiment Classification using MLP Network Hybrid with Naive Bayes Algorithm." Journal of Computer Science **14**: 1104-1114.

Ali, N., M. Hamid and A. Youssif (2019). "SENTIMENT ANALYSIS FOR MOVIES REVIEWS DATASET USING DEEP LEARNING MODELS." International Journal of Data Mining & Knowledge Management Process **09**: 19-27.

Almaghrabi, M. and G. Chetty (2020). Improving Sentiment Analysis in Arabic and English Languages by Using Multi-Layer Perceptron Model (MLP). 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA).

Almansor, E. H. and A. Al-Ani Translating dialectal Arabic as low resource language using word embedding.

Alomari, K. M., H. M. ElSherif and K. Shaalan (2017). Arabic Tweets Sentimental Analysis Using Machine Learning. Advances in Artificial Intelligence: From Theory to Practice, Cham, Springer International Publishing.

Alrefai, M. a., H. Faris and I. Aljarah (2018). Sentiment analysis for Arabic language: A brief survey of approaches and techniques.

Anoual, E. k. and I. Zeroual (2021). "The effects of Pre-Processing Techniques on Arabic Text Classification." International Journal of Advanced Trends in Computer Science and Engineering **10**: 41-48.

Baly, R., G. Badaro, G. Khoury, R. Moukalled, R. Aoun, H. Hajj, W. El-Hajj, N. Habash and K. Shaban (2017). A Characterization Study of Arabic Twitter Data with a Benchmarking for State-of-the-Art Opinion Mining Models.

Barhoumi, A., C. Aloulou, N. Camelin, Y. Estève and L. H. Belguith (2018). Arabic Sentiment Analysis: An Empirical Study of Machine Translation's Impact. LPKM.

Başarslan, M. and F. Kayaalp (2021). "Sentiment Analysis with Machine Learning Methods on Social Media." Advances in Distributed Computing and Artificial Intelligence Journal **9**: 5-15.

Birjali, M., M. Kasri and A. Beni-Hssane (2021). "A comprehensive survey on sentiment analysis: Approaches, challenges and trends." Knowledge-Based Systems **226**: 107134.

Boudad, N., R. Faizi, O. h. t. Rachid and R. Chiheb (2017). "Sentiment analysis in Arabic: A review of the literature." Ain Shams Engineering Journal **9**.

Chola, C., M. B. B. Heyat, F. Akhtar, O. AlShorman, B. B. Jv, A. Muaad, M. Masadeh and F. Alkahtani (2021). IoT Based Intelligent Computer-Aided Diagnosis and Decision Making System for Health Care.

Dahou, A. (2016). Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification.

Duwairi, R. (2014). "A Study of the Effects of Preprocessing Strategies on Sentiment Analysis for Arabic Text." Journal of Information Science **40**: 501-513.

El-Awady, R., S. Barakat and N. Elrashidy (2015). SENTIMENTANALYSIS FOR ARABIC AND ENGLISH DATASETS.

Elfaik, H. and E. H. Nfaoui (2020). "Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text." Journal of Intelligent Systems **30**: 395-412.

Elhassan, R. and M. Ahmed (2016). "Arabic Text Stemming: Comparative Analysis." Proceedings of the IEEE.

El-Masri, M., N. Berardinelli and H. Ahmed (2017). "Successes and challenges of Arabic sentiment analysis research: a literature review." Social Network Analysis and Mining **7**: 22.

Elnagar, A., L. Lulu and O. Einea (2018). "An Annotated Huge Dataset for Standard and Colloquial Arabic Reviews for Subjective Sentiment Analysis." Procedia Computer Science **142**: 182-189.

Fouad, M., A. Mahany, N. Aljohani, R. Abbasi and S.-U. Hassan (2020). "ArWordVec: efficient word embedding models for Arabic tweets." Soft Computing **24**.

Gamal, D., M. Alfonse, E.-S. M. El-Horbaty and A.-B. M. Salem (2019). "Implementation of Machine Learning Algorithms in Arabic Sentiment Analysis Using N-Gram Features." Procedia Computer Science **154**: 332-340.

Hamdi, A., K. Shaban and A. Zainal (2016). "A Review on Challenging Issues in Arabic Sentiment Analysis." Journal of Computer Science **12**(9).

Hawalah, A. (2019). "A FRAMEWORK FOR ARABIC SENTIMENT ANALYSIS USING MACHINE LEARNING CLASSIFIERS."

Heikal, M., M. Torki and N. El-Makky (2018). "Sentiment Analysis of Arabic Tweets using Deep Learning." Procedia Computer Science **142**: 114-122.

Maree, M., M. Eleyat, S. Rabayah and M. Belkhatir (2023). "A hybrid composite features based sentence level sentiment analyzer." IAES International Journal of Artificial Intelligence (IJ-AI) **12**: 284-294.

Mohammad, S., M. Salameh and S. Kiritchenko (2016). "Sentiment Lexicons for Arabic Social Media."

Muaad, A. Y., G. H. Kumar, J. Hanumanthappa, J. V. B. Benifa, M. N. Mourya, C. Chola, M. Pramodha and R. Bhairava (2022). "An effective approach for Arabic document classification using machine learning." Global Transitions Proceedings **3**(1): 267-271.

Nabil, M., M. A. Aly and A. F. Atiya (2015). ASTD: Arabic Sentiment Tweets Dataset. Conference on Empirical Methods in Natural Language Processing.

Nassif, A., A. Elnagar, I. Shahin and S. Henno (2020). "Deep learning for Arabic subjective sentiment analysis: Challenges and research opportunities." Applied Soft Computing **98**: 106836.

Oueslati, O., E. Cambria, M. B. HajHmida and H. Ounelli (2020). "A review of sentiment analysis research in Arabic language." Future Generation Computer Systems **112**: 408-430.

Oussous, A., F.-Z. Benjelloun, A. A. Lahcen and S. Belfkih (2020). "ASA: A framework for Arabic sentiment analysis." Journal of Information Science **46**(4): 544-559.

Oussous, A., F.-Z. Benjelloun, A. Ait Lahcen and S. Belfkih (2019). "ASA: A framework for Arabic sentiment analysis." Journal of Information Science **46**: 016555151984951.

Sayed, A., E. Elgeldawi, A. Zaki and A. Galal (2020). Sentiment Analysis for Arabic Reviews using Machine Learning Classification Algorithms.

Shoukry, A. and A. Rafea (2012). Sentence-Level Arabic Sentiment Analysis.

Soliman, A. B., K. Eissa and S. R. El-Beltagy (2017). "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP." Procedia Computer Science **117**: 256-265.

Syarief, M., O. Kurahman, A. Huda and W. Darmalaksana (2019). Improving Arabic Stemmer: ISRI Stemmer.

Taghva, K., R. Elkhoury and J. Coombs (2005). Arabic stemming without a root dictionary. International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II.

Touahri, I. and A. Mazroui (2021). "Deep analysis of an Arabic sentiment classification system based on lexical resource expansion and custom approaches building." International Journal of Speech Technology **24**: 1-18.

Umarani, V., A. Julian and J. Deepa (2021). "Sentiment Analysis using various Machine Learning and Deep Learning Techniques." Journal of the Nigerian Society of Physical Sciences **3**: 385-394.
Zerrouki, T. (2023). Tashaphyne: A Python package for Arabic Light Stemming.

Zou, X., Y. Hu, Z. Tian and K. Shen (2019). Logistic Regression Model Optimization and Case Analysis. 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT).

Appendices

Appendices

The following is a sample of the ASTD dataset and the changes that occur to it by applying the pre-processing steps and its impact on sentences.

1. Original dataset

	text	category	
0	بعد استقاله رٸيس ال+ محكمه ال+ دستوريه ن...	2-	0
1	امنٶ ال+ دكتور احمد جمال ال+ دين ال+ قيا...	1	1
2	ال+ برادعي يستقوي ب+ امريكا مره اخري و ...	1-	2
3	ال+ حريه و+ ال+ عداله شاهد ال+ ان ليله ا...	2-	3
4	ال+ والده لو اقول +ها ب+ خاطري حشيشه تضحك...	0	4

2. Tokenization

0	[بعد, استقاله, رٸيس, ال, +, محكمه, ال, +, دستو...
1	[امنٶ, ال, +, دكتور, احمد, جمال, ال, +, دين, ا...
2	[ال, +, برادعي, يستقوي, ب, +, امريكا, مره, اخر...
3	[ال, +, حريه, و, +, ال, +, عداله, شاهد, ال, +,...
4	[ال, +, والده, لو, اقول, +, ها, ب, +, خاطري, ح...

3. Removing Special Characters and Punctuation

0	بعد استقاله رٸيس ال محكمه ال دستوريه ننت...
1	امنٶ ال دكتور احمد جمال ال دين ال قيادي ...
2	ال برادعي يستقوي ب امريكا مره اخري و ير...
3	ال حريه و ال عداله شاهد ال ان ليله ال ات...
4	ال والده لو اقول ها ب خاطري حشيشه تضحك ب...

4. Removing Stop Words

0	استقاله رءيس ال+ محكمه ال+ دستوريه ننتظر استقا...
1	امنء ال+ دكتور احمد جمال ال+ دين ال+ قيادي ب+ ...
2	ال+ برادعي يستقوي ب+ امريكا مره اخري يرسل عصام...
3	ال+ حريه و+ ال+ عداله شاهد ال+ ان ليله ال+ اتح...
4	ال+ والده اقول +ها ب+ خاطري حشيشه تضحك اقول +ه...

5. ISRI Stemming

0	بعد قال رءس ال+ حكم ال+ دستور تظر قال رءس ال+ ...
1	منء ال+ دكتور حمد جمل ال+ دين ال+ قيد ب+ حزب م...
2	ال+ ردع سقو ب+ امر مره اخر و رسل عصم ال+ عري ا...
3	ال+ حره و+ ال+ عدل شهد ال+ ان ليل ال+ تحد اول ...
4	ال+ ولد لو اقل +ها ب+ خطر حشش تضح بس من اقل +ه...

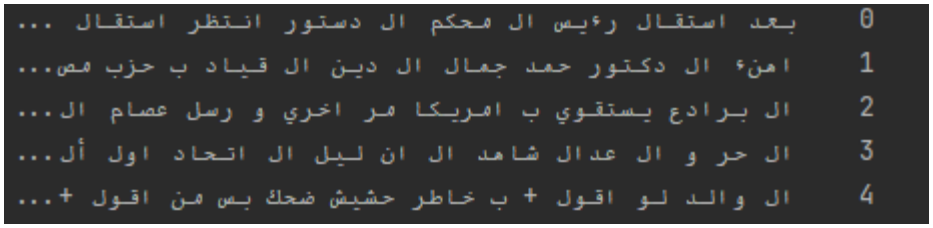
6. Snowball Stemming

0	بعد استقال رءيس ال+ محكم ال+ دستور ننتظر استقا...
1	امنء ال+ دكتور احمد جمال ال+ دين ال+ قياد ب+ ح...
2	ال+ رادع يستق ب+ امر مره اخر و يرسل عصام ال+ ع...
3	ال+ حر و+ ال+ عدال شاهد ال+ ان ليل ال+ اتحاد ا...
4	ال+ والد لو اقول +ها ب+ خاطر حشيش تضح بس من اق...

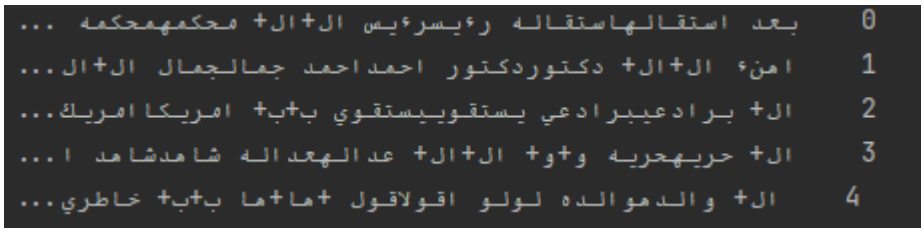
7. Arlight Stemming

0	بعد استقال رءيس ال+ محكم ال+ دستور ننتظر استقا...
1	امنء ال+ دكتور احمد جمال ال+ دين ال+ قياد ب+ ح...
2	ال+ رادع يستق ب+ امر مره اخر و يرسل عصام ال+ ع...
3	ال+ حر و+ ال+ عدال شاهد ال+ ان ليل ال+ اتحاد ا...
4	ال+ والد لو اقول +ها ب+ خاطر حشيش تضح بس من اق...

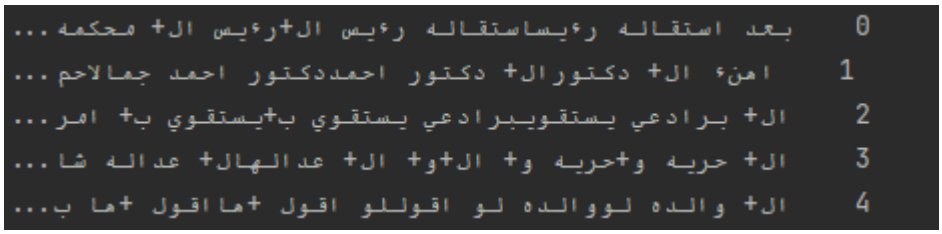
8. Lemmatization



9. Bigram



10. Trigram



الملخص

لقد عززت التطورات الأخيرة في تقنيات معالجة اللغة الطبيعية (NLP) بشكل كبير من قدرات معالجة وتحليل وفهم المشاعر المعبر عنها في المراجعات التي ينشئها المستخدمون عبر مختلف المنتجات والخدمات. وقد أدى هذا الاهتمام المتزايد بتحليل المشاعر إلى تحفيز جهود بحثية كبيرة. في هذه الدراسة، نستكشف تحليل المشاعر مع التركيز بشكل خاص على اللغة العربية. من خلال الاستفادة من تقنيات المعالجة المسبقة التقليدية وخوارزميات التعلم الآلي، نقترح نموذجًا شاملاً لتحليل المشاعر يتكون من أربع مراحل.

الهدف الرئيسي من نموذجنا هو الاستفادة من الموارد والتقنيات الخاصة باللغة الإنجليزية لقياس تأثيرها على دقة التصنيف عند تطبيقها على الجمل العربية. من خلال سلسلة من التجارب التي أجريناها على مجموعات البيانات العربية وترجماتها إلى اللغة الإنجليزية، قمنا بتقييم فعالية مختلف طرق المعالجة المسبقة ومصنفات التعلم الآلي: الانحدار اللوجستي ((LR، وغابة القرارات العشوائية (RF)، وخوارزمية نايف بايز (NB)، وآلة دعم المتجهات (SVM) . ومن الجدير بالذكر أن مصنف SVM تفوق باستمرار على الآخرين، حيث أظهر أعلى دقة في معظم السيناريوهات خاصة عند الجمع بين التلميم وإزالة الجذور.

علاوة على ذلك، نستكشف تأثير ترجمة مجموعات البيانات وإدراج المرادفات على دقة تحليل المشاعر. في حين أن ترجمة مجموعات البيانات من العربية إلى الإنجليزية والعكس لم تسفر عن تغييرات كبيرة في الدقة، إلا أن إدراج المرادفات من مجموعات البيانات الإنجليزية في تجارب تحليل المشاعر العربية أنتج نتائج متباينة. وهذا يسلط الضوء على تعقيدات الفروق اللغوية الخاصة والتحديات في التقاط المشاعر بفعالية عبر اللغات المختلفة.

عند مقارنة دراستنا مع البحوث السابقة التي استخدمت مجموعة بيانات ASTD، تبرز عدة اختلافات وتشابهات رئيسية. استكشفت الدراسات السابقة مجموعة من المصنفات، بما في ذلك SVM وNB وLR وCNN وRNTN، مع نتائج دقة تتراوح بين 85% و90% بالنسبة للميزات التقليدية مثل n-grams وTF-IDF وتضمينات الكلمات مثل Word2Vec.ومع ذلك، أظهرت خوارزمية RNTN معدل دقة أقل بنسبة 58.5%، بينما حققت خوارزمية SVM معدل دقة بنسبة 51.7%. وركزت أبحاث أخرى على نماذج التعلم العميق مثل CNN وLSTM، والتي أسفرت عن معدلات دقة تبلغ 64.3% و64.75% على التوالي. في المقابل، أكدت دراستنا على أهمية تقنيات المعالجة المسبقة المحددة، حيث أظهرت أن طرق المعالجة مثل التلميم وإزالة الجذور يمكن أن تحسن بشكل كبير من أداء مصنفات التعلم الآلي مثل SVM، مع تحقيق نتائج دقة تصل إلى 80%.

بشكل عام، تعرض دراستنا مشهدًا متطورًا لبحث تحليل المشاعر، مما يبرز قدرة التقنيات على التكيف لمعالجة التحديات والاختلافات اللغوية. تساهم هذه النتائج في الفهم الأوسع لمنهجيات تحليل المشاعر وتؤكد على أهمية مراعاة الفروقات اللغوية في مهام تحليل المشاعر. وأخيرًا، تشمل التوصيات للبحث المستقبلي توسيع مجموعة البيانات العربية واستكشاف نماذج التعلم العميق المتقدمة لالتقاط أنماط أكثر تعقيدًا. بالإضافة إلى ذلك، يمكن أن يؤدي تحسين الأدوات اللغوية الخاصة باللغة العربية إلى تعزيز دقة تحليل المشاعر بشكل أكبر. تهدف هذه الخطوات إلى معالجة تعقيدات التحديات اللغوية بشكل أفضل والمساهمة في تطوير منهجيات تحليل المشاعر بشكل أكثر فعالية.