Optimizing Machine Learning-based Sentiment Analysis Accuracy in Bilingual Sentences via Preprocessing Techniques

Mohammed Maree
Department of Information Technology, Arab American University, Palestine
mohammed.maree@aaup.edu

Mujahed Eleyat
Department of Computer Systems Engineering, Arab American University, Palestine
mujahed.eleyat@aaup.edu

Enas Mesqali
Department of Natural, Engineering and Technology Sciences, Arab American University, Palestine
e.mesqali@student.aaup.edu

Abstract: *With the recent advances in Natural Language Processing (NLP) technologies, the ability to process, analyze, and understand sentiments expressed in user-generated reviews regarding the products and services they use is becoming more achievable. Despite the latest improvements in this field, little attention has been given to multilingual sentiment analysis. In this article, a framework is presented for sentiment analysis in Arabic and English using two datasets (ASTD, AJGT) along with their translations. Preprocessing techniques, including n-gram tokenization, Arabic-specific stop words removal, punctuation removal, removing repeating characters, parts of speech tagging, stemming, and lemmatization, are applied. Four machine learning classifiers, namely Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB), and Support Vector Machine (SVM), are employed. We highlight existing specialized research in sentiment analysis for Arabic and English, as well as the employed techniques in each. Furthermore, the impact of preprocessing on accuracy results for both Arabic and English languages is investigated through separate experiments for each step. Experimental results on the ASTD dataset demonstrate close performance across classifiers, with the SVM classifier achieving the highest accuracy of 70%. However, the accuracy varied when using the AJGT dataset, with the NB classifier yielding the best accuracy at approximately 87%. The experiments on the translated datasets from Arabic to English did not exhibit significant differences, although some features performed slightly better using the Arabic datasets.*

Keywords: *Machine learning, bilingual sentiment analysis, NLP, sentiment datasets.*

Received July 20, 2023; accepted February 1, 2024
https://doi.org/10.34028/iajit/21/2/8

1. Introduction

In light of the expanding volume of data on social media platforms, sentiment analysis has garnered significant attention. This attention is attributed to the need to classify emotions expressed in natural language text. The growing interest in sentiment analysis stems from its role in supporting informed decisions in product development and service delivery. Additionally, it plays a crucial role in understanding consumers' perceptions regarding specific products or services. The ability to analyze sentiments expressed in user-generated content on social media platforms has become increasingly vital for businesses seeking actionable insights.

In particular, the goal of text sentiment analysis is to discern and articulate the sentiments expressed in texts, specifically determining whether the conveyed opinions lean positively or negatively toward the services and products offered in diverse domains of life. The significance of sentiment analysis has experienced a recent surge across multiple languages, with a particular focus on the Arabic language, which is currently in a developmental stage. This is mainly attributed to factors such as limited available sources, the intricate richness of its vocabulary, both formally and linguistically, and the varied ways in which it is written, encompassing various forms such as singular and plural. Additionally, the existence of numerous Arabic dialects alongside the standard language adds to the complexity of sentiment analysis in Arabic.

As suggested by Alrefai *et al*. [9], sentiment analysis can be categorized into four levels: document level, sentence level, aspect level, and word level Within this context, three prevalent approaches are employed to analyze text sentiment: lexicon-based, machine learning, and hybrid algorithms [9, 13].

In the lexicon-based approach, a lexicon captures words and their polarity according to the value of each word. Words whose value is greater than zero have positive polarity, and words whose value is less than zero have negative polarity, otherwise, words are of neutral polarity. In the context of machine learning, the approach employed relies on the machine's ability to learn from a dataset and make informed decisions based on the respective categories. As defined by Arthur Samuel, machine learning is the field of study that gives computers the ability to learn without being explicitly

programmed. In the same line of research, IBM[1] provides another definition for machine learning that is commonly quoted in the literature. It is defined as "a branch of Artificial Intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy". This can be accomplished through either supervised or unsupervised techniques, each with distinct characteristics. In supervised learning, the input and output data are pre-defined and labeled, enabling the construction of a model capable of analyzing sentiment and making predictions on new texts. This approach involves dividing the dataset into two parts: one for training the model and the other for testing and evaluation. The goal is to learn the relationship between inputs and their associated outputs, enabling the model to make predictions about new, unseen data based on patterns learned from the training data. On the other hand, unsupervised learning operates on unlabeled data, automatically uncovering patterns by analyzing the words and their polarity, distinguishing them as positive, negative, or neutral. Supervised techniques include decision tree, linear, probabilistic, and rule-based classifiers, in addition to deep learning techniques [9, 13]. While K-Means represents an example of an unsupervised techniques.

The hybrid approach on the other hand combines the strengths of both the lexicon-based approach and the machine learning approach. It utilizes features extracted from a lexicon or a collection of words, removing those that are not present in the dataset. Machine learning classifiers are then employed to leverage these features. By integrating these two approaches, the hybrid approach has proved to enhance accuracy results, as indicated by previous studies [9, 21].

The methodology of text sentiment analysis models includes several steps, starting from collecting datasets, then pre-processing the datasets, followed by identifying features to be used, then using sentiment analysis techniques, whether based on machine learning algorithms or lexicons, or the combination of both, and finally evaluating the results of the algorithms used by El-Masri *et al* [17]. This article will introduce relevant research works that deal with the mechanism of sentiment analysis of texts in Arabic, English and bilingual (Arabic-English), and present the methods and techniques used. Furthermore, we propose and discuss our methodology upon which the experiments are based, in addition to the results of the experiments and their discussion. We experimentally demonstrate that the pre-processing steps of the texts can significantly influence the quality of the sentiment analyzer for both English and Arabic texts.

In the subsequent sections, this paper is organized to provide a comprehensive understanding of the proposed sentiment analysis framework. Section 2 delves into a detailed review of the related work, highlighting existing research in sentiment analysis for both Arabic and English, and discussing the employed techniques in each. Section 3 provides an in-depth description of the proposed methodology, elucidating the preprocessing steps and machine learning classifiers employed in the analysis. Following this, section 4 presents the experimental setup, detailing the datasets used and the specific procedures undertaken to evaluate the performance of the sentiment analysis framework. The experimental results and their implications are discussed in section 5, shedding light on the impact of preprocessing on accuracy for both Arabic and English languages. Finally, section 6 concludes the paper, summarizing key findings, discussing potential extensions for future research, and emphasizing the broader significance of the study in the context of multilingual sentiment analysis.

2. Literature Review

The process of analyzing sentiments and understanding the context of the text is not an easy task, especially when the sentiment sentences are expressed in multiple languages, such as English and Arabic. In the rest of this section, we highlight the main research approaches employed for analyzing sentiments in these languages.

2.1. Sentiment Analysis for English Language Texts

Several studies presented the mechanism of sentiment analysis of English text, methods used in preprocessing text, and techniques applied to English datasets in order to make supportive decisions about a particular product or service.

The researchers used different classifiers and applied them to many datasets. Başarslan and Kayaalp [12], used machine learning algorithms: Naïve Base (NB), Support Vector Machine (SVM), and Artificial Neural Networks (ANNs) to classify textual sentiment for the two datasets, the first set of 4,500 tweets aggregated through an Application Programming Interface (API) and the second dataset of reviews for IMDB movies bundled using Kotzias. Also Al Shamsi *et al*. [2], the researchers used machine learning algorithms: NB, Iterative Dichotomiser 3 (ID3), K-Nearest Neighbor (K-NN), Decision Tree (DT), Random Forest (RF), and Random Tree (RT), and applied them to balanced and unbalanced datasets consisting of more than 14,000 Kaggle tweets taken from six US airlines (United, Delta, Southwest, Virgin America, US Airways, and American). While Ali *et al*. [6] the researchers used deep learning algorithms: Multilayer Perceptron (MLP), Long-Term Memory (LSTM), and Convolutional Neural Network (CNN) to analyze the emotions of the IMDB dataset and also

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

proposed a hybrid model of Long Short Term Memory (LSTM) and CNN.

Al shamsi *et al.* [2] and Başarslan and kayaalp [12] performed text preprocessing which included removing punctuations, stop words, word root derivation, converting uppercase to lowercase, tokenization, and then extracting features using Term Frequency-Inverse Document Frequency (TF-IDF), and Word2vec, where the TF-IDF method is concerned with distinguishing distinctive words and unimportant words such as Stop words, knowing how many repeats a term is in the document and determining its weight, while word2vec is concerned with representing words as vectors and similar words are close in coordinates and uses two different representations: Continuous Bag Of Words (CBOW) and Skip-Gram (SG). The first method, CBOW, depends on the representation of the target word through the neighboring words in the sentence after determining the size of the frame. In SG, the adjacent words are represented in the sentence after determining the target word. In addition to text pre-processing, they performed in Al Shamsi *et al.* [2] the dataset separation into two parts, where the first for training 66% of the dataset and the second 34% for testing.

Ali *et al.* [6], addition to pre-processing and word embedding using Word2vec, authors employed a neural network for sentiment classification. The length of the text for processing is limited to 500 words and not more. Then comes the role of the CNN by using weights and biases to teach the multiple layers formed in each neuron in the training phase. In each neuron, the output transformation is activated using the Rectified Linear Unit (RLU), and the convolutional layer outputs are aggregated and condensed by reducing the dimensions, also CNN learning the structure of text to be categorized.

The Maxpooling layer is one of the processing layers in the LSTM and CNN hybrid model, in which word weights of the highest value are chosen and other values are ignored by applying a Maxpooling mask to the data sequentially, which leads to reducing the text length specified for processing from 500 words to 250 words.

The output of the Maxpooling layer constitutes the input to the LSTM layer, which reads, writes, and stores data from the cell that is a component of the LSTM, as well as other components such as input, output, and forget gates. Depending on the strength and weakness of the signals received by the gates, the cell takes the decision to read, write or delete data through the gates. A confusion matrix was used in the evaluation process [12], Where the results was for the IMDB dataset of 500 positive reviews, 500 negative reviews and Twitter datasets of 4,500 tweets related to health data showed as follows: 1,220 positive tweets, 1,600 negative tweets, and 1,680 tweets neutral. Experiments using NB, SVM, and ANN techniques showed that the best results were in favor of ANN in both datasets using TF-IDF and word2vec features. The performance of ANN using the TF-IDF feature for the IMDB dataset using the four scales was as follows: 89%, 88%, 88%, and 89%, and for the Twitter dataset it was 86%, 84%, 87%, and 85%, while using the word2vec feature was for a dataset IMDB were as follows: 90%, 90%, 91% and 96% and for the Twitter dataset 87%, 86%, 88% and 86%, and the worst performance was using NB technology in both datasets.

Al Shamsi *et al.* [2], the NB machine learning algorithm in addition to ID3 was one of the best results. The results of the experiments showed the unbalanced dataset for the six airlines: United, Delta, Southwest, Virgin America, US Airways, and American, and the dataset for each company was as follows: 3822, 2222, 2420, 504, 2913, and 2759, respectively. The accuracy of the classifications in the dataset of some airlines is high, such as United and US Airways, and low in another dataset such as Virgin America, and the reason is due to the size of the data, the larger the dataset, the better the accuracy. While the best accuracy results were using the NB, Decision Tree, ID3, and Random Tree classifiers.
As for the results of the experiments for the balanced dataset for the six airlines: United, Delta, Southwest, Virgin America, US Airways, and American, the dataset for each company was as follows: 2635, 5518, 6608, 8276, 5924, and 8276, respectively. The best results were using NB and ID3 for the Southwest, Virgin America, and American datasets. When comparing the performance of the classifiers in the experiments of the balanced and unbalanced datasets, the best results were with the balanced datasets for each of the NB and ID3 classifiers, which achieved a maximum accuracy of 97.65%, while it reached with the unbalanced datasets 82.72%, in contrast, K-NN and DT achieved Better accuracy results with unbalanced datasets, reaching 82.72%, while it was low with balanced datasets, reaching 38.79%. Also, RF and Random Tree were better with unbalanced datasets, reaching 82.72%, while it was 35.06% with balanced datasets.

Ali *et al.* [6] the experimental work applied to IMDB dataset of 50,000 movie reviews using the proposed LSTM and CNN hybrid approach and its comparison with deep learning models (MLP, CNN, LSTM) and traditional machine learning techniques (SVM, NB, Recursive Neural Tensor Network (RNTN)) using Python software, Keras library, Tensor and selecting 80% of the dataset is for the training phase and 20% for the testing phase. The results showed the highest accuracy of the proposed LSTM and CNN hybrid model with 89.20%, followed by CNN, MLP, and LSTM models with 87.70%, 86.74%, and 86.64%, respectively. Experimental work was also carried out using another dataset of movie reviews in the English language consisting of 11,855 reviews and by applying the RNTN model, the accuracy was one of the lowest results by 80.70%. On the other hand, SVM and NB were used with a dataset consisting of 2053 reviews, including 1301 positive reviews and 752 negative reviews, the accuracy results were 82.90% and 81%, respectively.

2.2. Sentiment Analysis for Arabic Language Texts

Despite the recent interest of researchers in the field of sentiment analysis, the analysis of sentiments of the Arabic language is still in its infancy due to the lack of available resources in the Arabic language compared to the English language. The Arabic language is one of the most difficult languages in the field of sentiment analysis because it is one of the complex texts in writing and understanding. The Arabic language contains two writing styles, one of which is in Standard Arabic, which is used in books, letters and other official matters, and the other is dialectal, which is used in people's daily lives. Some of the complications in the way of writing the Arabic text is to write the word in more than one form, where it is possible to write the Ta' marbootah (ة) at the end of the word (ه), such as "المقدمة" or "المقدمه", as well as negating phrases with words that are considered stop words that are removed when pre-processing the text, which affects the classification of the text. The sentence from being negative to a sentence with a positive feeling, and there are also some phrases that carry negative feelings and do not contain negation words such as "حسبي الله ونعم الوكيل". Writing the same verb in several forms according to the subject being plural, singular, feminine or masculine, such as "هي تحب المطر" (She loves rain), and "هو يحب المطر" (he loves rain). Also, there are nouns that do not carry any feelings that are written in the same way as adjectives that carry feelings, such as the noun "جميلة" and the adjective "جميلة" [9].

The researchers used the lexicon-based method in analyzing the feelings of the Arabic language as, in Mohammad *et al*. [22], while in [25, 18] they used the word embedding based on the machine learning methods. Where in Mohammad *et al*. [22] they combined ancient and modern lexicons and compared the performance of each. The researchers explained the mechanism of creating the Arabic sentiments lexicon, either by using remotely supervised techniques or by machine translation of the Arabic text from the English language via the free Google Translate website, whether it is to translate a word or a sentence. In addition, the authors provided three manual lexicons for analyzing Arabic sentiments. The results of the accuracy experiments conducted to analyze Arabic feeling at the sentence level by training the SVM classifier on different datasets using basic features (n-grams and other forms) reached 62%, and improved it to 63% using manual dictionaries.

Whereas researchers in Soliman *et al.* [25] used the open-source technology AraVec to represent Arabic text words in several fields from different sources: Twitter, Wikipedia, and World Wide Web pages. Where various AraVec models were used to embed the words. Pre-processing the collected data is one of the most important stages in building a word embedding model because it has an impact on accuracy results. The preprocessing of the text includes the beginning of removing the non-Arabic text from the Arabic text after distinguishing it, especially the languages that overlap with the Arabic language in some letters such as Persian and Urdu, and then normalization represented by removing the diacritical movements from the text and replacing the letters "أ،آ،إ" with the letter "ا", as well as the letter "ة" with the letter "ه" and the letter "ى" with the letter "ي", also converting repeated letters in the word to the same single letter, such as "سلاااااام" converting it to "سلام", as well as replacing emojis and URLs with text to distinguish them [8]. Also, the authors used Vector Space Models (VSMs) to represent data in a word embedding model, where words are represented in a continuous space, and therefore similar words are close in space, and VSMs depend on the assumption that words that come in the same text content have similar connotations and it is called the distributive hypothesis. VSMs use two approaches based on counting and prediction in representing the data. The first approach relies on word occurrence statistics and representing each word with a dense vector, while the second approach relies on training the neural model, knowing the values of word vectors, predicting the target word through its neighboring words, and using Word2Vec techniques for word embedding.

Fouad *et al*. [18], the researchers presented an ArWordVec model to analyze the feelings of Arabic texts by word embedding using the three methods: the CBOW, the SG and global vectors for word representations (GloVe). The performance of the ArWordVec model was evaluated using two Twitter datasets: The ASTD and AraSenti. To build the ArWordVec model, several steps are taken, starting with collecting data from 55 million tweets, covering many areas, then pre-processing the data, which includes removing hashtags, symbols, non-Arabic letters, punctuation, stop words, repeated texts, diacritics, spaces, normalization of letters, and character processing duplicate and long. Thus, the text is ready for processing in the ArWordVec model using the word2vec toolkit that classifies the text into a list of input and output words so that it can be used in building a neural network model and supervised learning where the CBOW method analyzes the text to find out the target word from the surrounding words while the SG method defines a set Surrounding words of the target word. The researchers evaluated the word embedding model for the English language text through word similarity tasks, and due to the poor Arabic language resources and the lack of resources for word similarity tasks, the word embedding model was evaluated by using a seed group of words categorized into 107 positive words and 118 negative words, then the search and comparison process is done in tweets on the most similar words with the seeds and giving them the same classification category as well as collecting the seed word vectors and comparing them with the similar word matrix through the training phase

of the classifiers used such as SVM or NB. Barhoumi *et al.* [11], presented a different methodology from the previously mentioned in analyzing sentiments in Arabic, depending on a translation mechanism, where sentiments for Arabic text are analyzed and compared with the machine translation of the same text in the English language to verify the extent of the change in polarity and its impact on performance. The two LR and MLP classifiers were used in the experiments using vector embedding from the documents and composed of a series of two vectors, the first is Distributed Memory (DM) and the second is the Distributed Bag of Words (DBOW). Using the Large Scale Arabic Book Reviews (LABR) dataset.

Experiments were evaluated using the error rate scale, the results of the error rate for the Arabic baseline was 25.37% and the text translated into English was 23.70%. This is explained by the fact that the polarity of the text has not changed. The noticeable improvement in the results of the translated text is due to the removal of words that do not belong to polarity and considering them as annoying words because they are not translated into English either because they are a proper name or contain duplicate letters or because they are from colloquial dialects or the word was written in Arabic letters and it is not Arabic for example ("رفيو" is the origin of the word "review" and "بروتكشن" is derived from "protection"). So, the presence of these words in the text is considered confusing and misleading to discover polarity. When the experiment was repeated using the original Arabic text with the deletion of misleading words, the error rate increased and it was 26.86%, and this is explained by the importance of the omitted words in the Arabic text and their lack of importance in the machine-translated text. Experiments were also conducted on the original Arabic text using various additions, such as the experiment of using light stemming, which is one of the preprocessing tools for the text, and the error rate was 23.31%, which is close to the result of using the machine-translated text and better than the result of using the original text. The results indicate that using the machine translation approach as a tool Statistical or preprocessing using the light stemming approach as a linguistic tool gives close results, that is, they work in the same way, and the possibility of using machine translation as an alternative to stemming in order to obtain a good sentiment analysis system.

2.3. Sentiment Analysis for Bilingual (English-Arabic)

Since the majority of studies are devoted to the field of sentiment analysis towards English language texts and the scarcity of resources in Arabic and other languages, despite the presence of a high percentage of Arabic language pioneers on the Internet, as people express their own language and circulate among them, and because the challenge is the lack of available resources This challenge to the field of sentiment analysis has been dealt with by transferring knowledge from resource-rich languages such as English to resource-scarce languages such as Arabic.

El-Awady *et al.* [15] the Senti-Word lexicon of Arabic vocabulary was created and used machine learning algorithms: Decision trees, Naïve Bayes, and support vector machines to classify sentiments for text in Arabic and English. The authors conducted sentiment classification experiments using datasets represented by (Movie, DVD, Books, and Electronics) collected from Amazon, consisting of equally positive and negative data, numbering 1000. The sentiment analysis features were selected using several methods (Information Gain, Unigram, rough set, mRMR, and Hybrid), IG method is concerned with determining the number of times the data is repeated, the percentage of word importance, and arranging the features in descending order of importance. And use the RS method with the IG method to get the perfect feature with less time and effort by eliminating redundant and annoying data. It used a confusion matrix to assess the performance of the classifiers used. Experiments using the English language dataset and using the four specific features after applying the feature selection techniques to obtain ideal features. The reason for the accuracy of identifying features in the IG and mRMR methods, because the first chooses the most important features and the second gets rid of excess and unwanted features. While the accuracy increases when using IG and Rs together as well when using Rs, IG and mRMR by 1.5% and 4.2%, respectively.

As for the Arabic language experiments using the lexicon that contains the word and its polarity, and applying the classification process to a dataset collected from the YouTube website, which included 214 Arabic films, whether texts of Arabic origin or translated from other languages, where 25% of the dataset were translated into Arabic and 3% Translated from French, the dataset was prepared by performing a preprocessing of removing redundant, unimportant and non-Arabic texts as well as removing stop words and then tokenizing and Stemming the text using Arabic Stemmer Khoja. The evaluation used a procedure classification without selecting the feature in an experiment and classification by choosing a feature with another experiment. The accuracy rate in the first experiment using the SVM and NB technologies was 83.96% and 89.34%, respectively, while in the second experiment, the results were better using the IG feature and the accuracy rate was 91%, and the results showed the best use of NB technology ranks above the rest of the technologies used. By applying preprocessing in Senti-Word Lexicon experiments, accuracy results varied according to the factors used in the sixteen experiments, and the experiment using the factors: normalization, removing stop words, and weights achieving the best accuracy of 95.9%, while the accuracy decreased to 85.5% when using the stemming factor only, and the results vary between them for the rest

of the experiments. The accuracy was 93.8% when using the normalization factor only, and the accuracy increased when using the weights factor only, and it reached 94.5%, while when using the Removing SW factor only, the accuracy reached 93.7%. The results of the experiments showed a decrease in the percentage of accuracy when using the stemming factor with one or more other factors. The accuracy when using the stemming factor with weights or with the Removing SW factor or with both, according to the following percentages, respectively, was 87.9%, 86.7%, 87.4%. It also showed an improvement in the results when using the Removing SW factor from 85.9% to 87.9%. Finally, the researchers compared the results of accuracy in their experiments and the experiences of other researchers using methods and different Arabic datasets, and the best results were for the proposed approach in this study using machine learning techniques and the Senti-Word Lexicon, where the accuracy rate was 94.5%.

Also Abo *et al.* [1] applied the NB and DT algorithms to three datasets, the first in English, collected from the social network Facebook, through the use of the Facebook developer API, and a total of 658 comments were made about the football match. The second dataset is reviews of books in modern Arabic, collected from Goodreads, with a total of 63,000 reviews within a month, of which 2,648 reviews were used. As for the third dataset, in Arabic dialects, collected manually from JEERAN, amounting to 409 customer reviews. It also pre-processed the English language dataset was done, then analyzing the feelings of the text using RapidMiner software and categorizing it into positive and negative. Repeat the sentiment analysis process for the two datasets of Modern Standard Arabic and Dialects Arabic, and use the accuracy and runtime scales in evaluating the NB and DT algorithms on the three datasets. The best results of classification experiments for the modern standard Arabic dataset were in both algorithms by evaluating the accuracy scale. The accuracy using DT was 97% and 89.50% using NB. The worst results were in the Dialects Arabic dataset, where the accuracy in both DT and NB reached 54.4% and 50.8%, respectively. In the English language dataset, the accuracy of DT was 83.87% and 84.25% for NB. In terms of evaluating the running time, the DT algorithm consumed more time in the modern standard Arabic dataset than the Dialects Arabic.

In a similar line of research Almaghrabi *et al.* [7], used a deep learning approach to analyze the feelings of Arabic and English texts using the Multilayer Perceptron (MLP) model represented by a neural network that depends on data representation through the word vector and prediction of feelings, taking into account the Times of the fonts used in word processing such as the use of Times Roman or Times New Roman. An Arabic language dataset containing 1524 movie reviews was used. While the English dataset contained 515,000 reviews aggregated from 1,493 hotels. The experiments were evaluated the performance of the MLP model in text prediction and measure the accuracy of the model through measures of Accuracy, Precision, Recall and F1-Score for both datasets. The accuracy of the MLP model through experiments on the Arabic dataset reached 87%, while it reached 96% on the English dataset. This means that the prediction using the MLP model gave good results compared to the results of experiments in other studies using the Word2Vec model, where the classification outcome for the results was negative for all the reviews that contained negative and positive reviews, while the results of using the Word2Vec model for the English dataset were good. This means that the accuracy results have improved on the Arabic datasets using the MLP model.

3. Proposed Methodology

Our proposed approach to sentiment analysis revolves around data preprocessing [18], encompassing tasks such as tokenization, removal of stop words, elimination of punctuation marks, handling repeated characters, and applying stemming and lemmatization techniques [24]. Following this preprocessing stage, we extract relevant features and leverage machine learning algorithms to assess performance. The dataset is divided into a training set comprising 70% of the data and a testing set comprising the remaining 30%.

In our approach, several experiments are performed using Arabic datasets and some of them included translating the same datasets to English using Google Translate API. Pre-processing is performed in all experiments and it includes tokenization and removal of Arabic-specific stop-words, punctuation removal, and removing repeating characters and parts of speech. Moreover, lemmatization is performed using qalsadi lemmatizer while stemming is achieved using three stemmers: ISRI Stemmer, Tashaphyne (ArabicLight Stemmer), and snowball stemmer for Arabic datasets. In addition, the same preprocessing steps were implemented on the translated dataset but using different Python libraries that are suitable for the English language. After preprocessing, feature selection and four machine learning classifiers (LR, RF, NB, and SVM classifiers) to obtain the results. Finally, the accuracy is calculated for each classifier and a comparison is made between them. Figure 1 depicts the overall framework of the proposed approach.

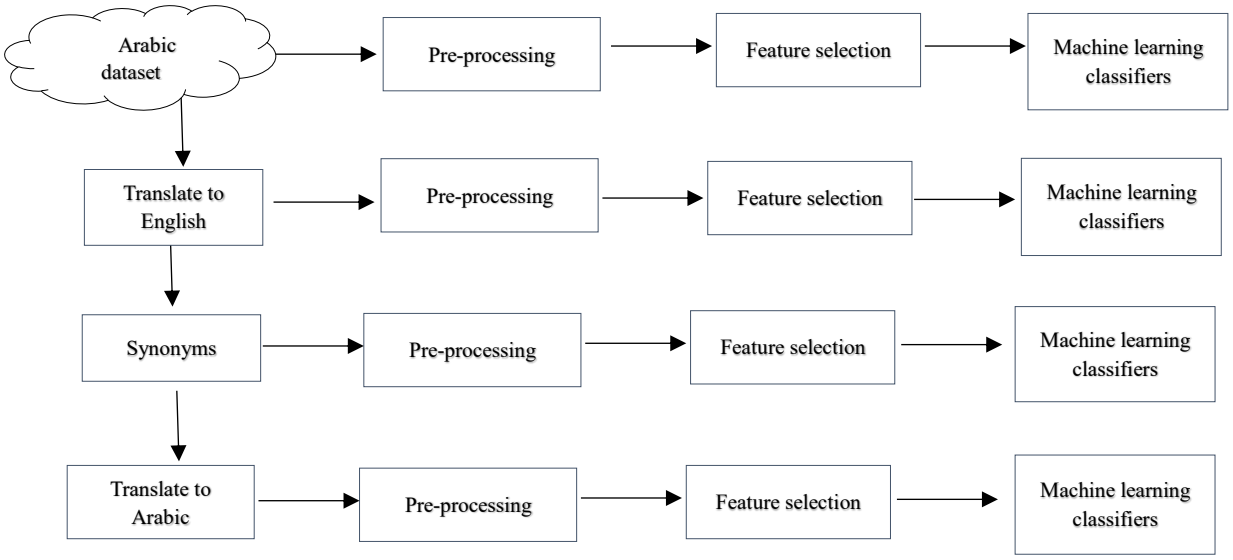

Figure 1. The overall framework of the proposed approach.

Our experiments, as depicted by Figure 1, are based on four axes. The first axis is the use of an Arabic-origin datasets, pre-processing it, identifying the feature, and then using machine learning classifiers to measure accuracy. The second axis is translating Arabic datasets into English, pre-processing them, identifying the feature, and then using machine learning classifiers. As for the third axis, it is extracting synonyms for English datasets, pre-processing them, identifying the feature, and then using machine learning classifiers. Finally, the fourth axis is translating synonyms of English datasets into Arabic, pre-processing them, defining the feature, and then using machine learning classifiers. Of course, each axis permeates many experiments by identifying one or more features shown in Tables 1 to 16 as shown in the following section.

Certainly, the use of the confusion matrix in the experiments aligns with best practices for evaluating the performance of classification models. The confusion matrix provides a granular view of the model's predictions, allowing to assess not only overall accuracy but also other key metrics such as precision, recall, and the F1_score.

The Accuracy (ACC) is calculated using the following Equation (1) [4]:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Where:

- True Positives (TP): the number of instances correctly predicted as positive.
- True Negatives (TN): the number of instances correctly predicted as negative.
- False Positives (FP): the number of instances incorrectly predicted as positive.
- False Negatives (FN): the number of instances incorrectly predicted as negative.

Precision (P) is calculated as in Equation (2):

$$P = \frac{TP}{TP + FP} \quad (2)$$

Recall (R) is calculated as in Equation (3):

$$R = \frac{TP}{TP + FN} \quad (3)$$

Finally, F1_Score is calculated as in Equation (4):

$$\text{F1\_Score} = \frac{2 * P * R}{P + R} \quad (4)$$

In our experiments, we used the accuracy measure to evaluate the performance of the classifiers, where accuracy represents the proportion of correctly classified cases (positive and negative) out of the total cases. It is a commonly used metric to evaluate the overall performance of a classification model.

4. Experimental Evaluation and Results

In our experiments, we used two Arab datasets: ASTD and AJGT [8, 23]. The Arabic Sentiment Tweets Dataset: "ASTD" dataset consisting of 10006 tweets in MSA "Modern Standard Arabic" and Egyptian dialect, categorized into positive, negative, neutral, and objective numerically expressed (1, -1, 0 and -2), distributed as 799, 1684, 832 and 6691 respectively, as shown in the Figure 2.

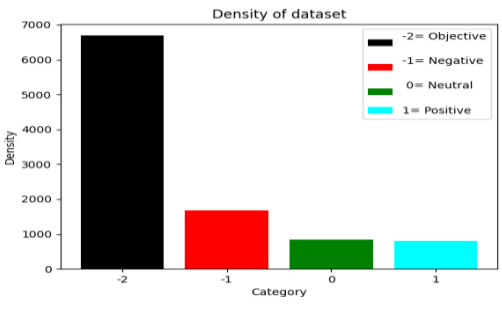

Figure 2. Distributed ASTD dataset.

The dataset was also filtered and the data referring to the objective category, which numbered 6,691, was

removed, so that the number became 3,315 distributed over the three categories previously mentioned and expressed numerically (1, -1, and 0), as shown in Figure 3. The same dataset was used after removing the neutral dataset, resulting in a dataset of 2483 tweets as shown in Figure 4. The second used dataset is the Arabic Jordanian General Tweets: "AJGT" which consists of 1800 tweets in MSA "Modern Standard Arabic" and Jordanian dialect, categorized into positive and negative, distributed as 900 positive and 900 negative, as shown in the Figure 5.

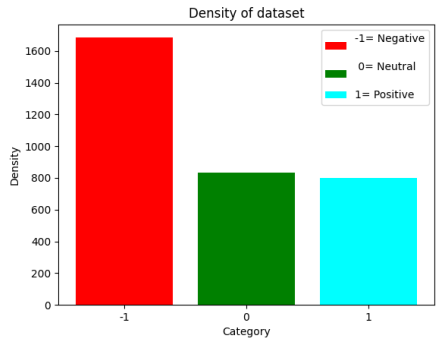

Figure 3. Distributed ASTD dataset without objective category.

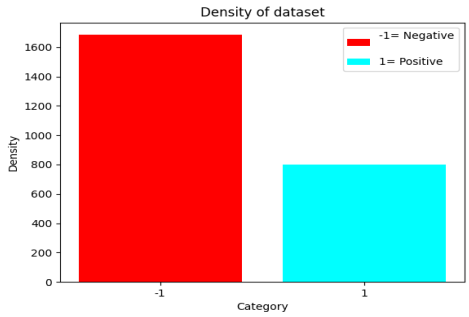

Figure 4. Distributed ASTD dataset without objective and neutral category.

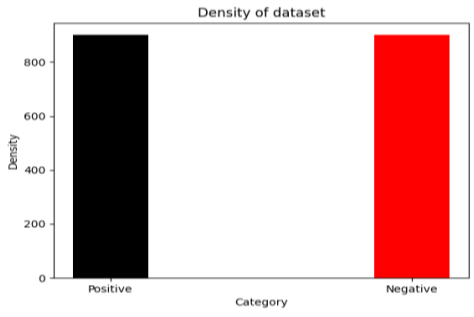

Figure 5. Distributed AJGT dataset.

The results presented in Table 1 showcase the accuracy outcomes of experiments conducted with the Arabic ASTD dataset. The accuracy figures ranged between 67% and 71% when employing one or two features simultaneously for the four machine learning classifiers. The SVM classifier demonstrated superior performance, particularly when utilizing the ISRI Stemmer and Trigrams. Moving to Table 2, which displays the results after the translation of the Arabic ASTD dataset into English, the accuracy ranged between 66% and 70%. Notably, the SVM classifier continued to outperform others, while the least favorable outcomes were observed when incorporating parts of speech tags with a focus on verb identification.

Table 1. Accuracy results for ASTD dataset in a variety of cases.

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%
Stem_ISRI	70%	68%	67%	70%
Stem_ArListem	70%	68%	67%	70%
Stem_Snowball	70%	68%	67%	70%
Lemmatization	69%	69%	67%	70%
Bigram	69%	68%	67%	70%
Trigram	69%	68%	67%	69%
Trigram + Stem_ISRI	70%	68%	67%	71%
Lemmatization + Stem_ Snowball	69%	69%	67%	70%

In the subsequent phase of experiments, as depicted in the Table 3, synonyms from the English language were introduced after being translated from the Arabic dataset using the WordNet program. The accuracy results in these tables mirrored those in Table 2, ranging between 66% and 70%, with the SVM classifier consistently achieving the highest accuracy.

Table 2. Accuracy results for ASTD dataset translated to English in a variety of cases.

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%
Stem_ Porter	70%	68%	67%	70%
Stem_Snowball	70%	68%	67%	70%
Lemmatization	70%	68%	67%	70%
Bigram	69%	68%	67%	69%
Trigram	69%	68%	67%	70%
Pos_tag "VB"	67%	66%	67%	67%
Tokenizing+ Remove stop word	69%	69%	67%	70%
Lemmatization + Pos_tag "VB"	67%	66%	67%	67%

Table 3. Accuracy results for English synonymous ASTD dataset in a variety of cases.

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%
Stem_ Porter	70%	68%	67%	70%
Stem_Snowball	70%	68%	67%	70%
Lemmatization	70%	68%	67%	70%
Bigram	69%	68%	67%	69%
Trigram	70%	68%	67%	70%
Pos_tag "VB"	70%	68%	67%	70%
Tokenizing+ Remove punctuation	70%	68%	67%	70%
Trigram + Remove punctuation	70%	68%	67%	70%

The fourth axis of experiments involved translating the English language equivalents of the dataset into Arabic, as shown in the Table 4. In these experiments, a decline in accuracy results compared to the Table 1 was observed, ranging between 67% and 70%. Notably, the best results were obtained using Lemmatization and

ArListem (Arabic Light Stemmer) features, with the SVM classifier once again showcasing the most robust performance.

Table 4. Accuracy results for synonymous ASTD dataset translated to Arabic in a variety of cases.

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%
Stem_ISRI	69%	67%	67%	69%
Stem_ArListem	68%	67%	67%	69%
Stem_Snowball	69%	68%	67%	69%
Lemmatization	69%	68%	67%	69%
Bigram	67%	67%	67%	67%
Trigram	67%	68%	67%	68%
Tokenizing + Lemmatization _	69%	68%	67%	69%
Bigram + Stem_ISRI	69%	68%	67%	69%

In Table 5, the AJGT dataset was utilized, resulting in accuracy ranging between 80% and 87%. The accuracy results were similar across all classifiers, and the NB classifier performed the best, especially when Snowball Stemmer and Lemmatization were selected as features.

Table 5. Accuracy results for AJGT dataset in a variety of cases.

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%
Stem_ISRI	83%	85%	86%	85%
Stem_ArListem	86%	84%	86%	86%
Stem_Snowball	84%	85%	87%	84%
Lemmatization	86%	85%	87%	87%
Bigram	83%	80%	83%	83%
Trigram	83%	81%	83%	82%
Tokenizing + Lemmatization _	86%	85%	87%	87%
Lemmatization + Stem_ Snowball	86%	86%	87%	86%

Moving on to Table 6, which presents the accuracy of the AJGT dataset after automatic translation into English, a slight decrease in accuracy was observed. The results ranged between 80% and 85%, with the SVM classifier achieving the best results. However, when utilizing the parts of speech tags feature in combination with other features, the accuracy dropped significantly, ranging from 49% to 54%.

Table 6. Accuracy results for AJGT dataset translated to English in a variety of cases.

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%
Stem_ Porter	85%	84%	84%	86%
Stem_Snowball	85%	84%	84%	86%
Lemmatization	85%	82%	85%	84%
Bigram	82%	80%	81%	81%
Trigram	83%	84%	84%	83%
Pos_tag "VB"	52%	52%	52%	52%
Trigram + Pos_tag "VB"	49%	49%	49%	49%
Lemmatization + Stem_ Snowball	85%	85%	85%	86%

Table 7 demonstrates an improvement in results compared to the previous tables. The accuracy in this set of experiments ranged between 81% and 86%, with the SVM classifier achieving the best performance. Nevertheless, when the parts of speech tags feature was selected with any other feature, the results still exhibited relatively lower accuracy, ranging between 48% and 55%.

Table 7. Accuracy results for English synonymous AJGT dataset in a variety of cases.

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%
Stem_ Porter	85%	84%	86%	85%
Stem_Snowball	85%	85%	86%	85%
Lemmatization	86%	82%	86%	86%
Bigram	84%	82%	82%	84%
Trigram	85%	84%	84%	85%
Pos_tag "VB"	53%	53%	53%	53%
Trigram + Stem_ Porter	86%	84%	85%	86%
Trigram + Pos_tag "VB"	48%	48%	48%	48%

Conversely, Table 8 indicates a noticeable drop in accuracy compared to Table 5. The accuracy in this set of experiments ranged between 73% and 82%, with the NB classifier achieving the best performance among the classifiers used.

Table 8. Accuracy results for synonymous AJGT dataset translated to Arabic in a variety of cases.

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%
Stem_ISRI	80%	78%	79%	80%
Stem_ArListem	80%	79%	81%	78%
Stem_Snowball	79%	79%	82%	79%
Lemmatization	81%	81%	83%	81%
Bigram	74%	73%	77%	76%
Trigram	77%	73%	76%	77%
Tokenizing + Lemmatization _	81%	82%	83%	81%
Trigram + Remove punctuation	77%	72%	76%	77%

Table 9 presents the results of experiments conducted on the ASTD dataset, specifically excluding the objective category of polarity. The accuracy in this case ranged between 55% and 63%, with SVM yielding the best results and the NB classifier performing the worst.

Table 9. Accuracy results for ASTD dataset without objective category in a variety of cases.

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%
Stem_ISRI	62%	61%	57%	61%
Stem_ArListem	61%	61%	57%	61%
Stem_Snowball	62%	61%	57%	63%
Lemmatization	62%	61%	57%	62%
Bigram	61%	60%	56%	61%
Trigram	56%	60%	55%	59%
Tokenizing+ Remove stop word	62%	61%	58%	61%
Tokenizing + Stem_Snowball	62%	60%	57%	63%

In Table 10, the accuracy ranged between 51% and 65%. The LR and SVM classifiers achieved the highest accuracy, while the RF classifier showed poor performance, particularly when selecting the bigram feature and incorporating parts of speech tags with a focus on verb identification. In such cases, the accuracy dropped to 25%.

Table 10. Accuracy results for ASTD dataset translated to English without objective category in a variety of cases.

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%
Stem_ Porter	64%	61%	58%	64%
Stem_Snowball	65%	62%	59%	64%
Lemmatization	63%	61%	58%	64%
Bigram	62%	59%	57%	62%
Trigram	61%	61%	56%	63%
Pos_tag "VB"	54%	53%	54%	54%
Tokenizing+ Remove stop word	65%	63%	61%	64%
Bigram+ Pos_tag "VB"	54%	25%	54%	54%

Moving to Table 11, the accuracy ranged between 54% and 65%, with the SVM classifier demonstrating the best performance among the classifiers used.

Table 12 displayed accuracy results ranging between 56% and 61%. This indicates a decrease in accuracy compared to Table 9. The LR and SVM classifiers performed the best in this scenario.

Table 11. Accuracy results for English synonymous ASTD dataset translated without objective category in a variety of cases.

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%
Stem_ Porter	65%	62%	59%	65%
Stem_Snowball	65%	62%	59%	65%
Lemmatization	65%	62%	58%	65%
Bigram	62%	60%	57%	62%
Trigram	62%	62%	55%	61%
Pos_tag "VB"	54%	54%	55%	55%
Tokenizing+ Stem_ Porter	65%	62%	59%	65%
Lemmatization + Stem_ Snowball	64%	63%	59%	65%

Table 12. Accuracy results for synonymous ASTD dataset translated to Arabic without objective category in a variety of cases.

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%
Stem_ISRI	61%	59%	58%	61%
Stem_ArListem	60%	60%	57%	59%
Stem_Snowball	61%	60%	57%	60%
Lemmatization	61%	60%	58%	60%
Bigram	57%	58%	56%	57%
Trigram	57%	59%	56%	59%
Tokenizing+ Remove stop word	61%	61%	58%	61%
Tokenizing + Lemmatization _	61%	60%	58%	60%

Table 13 showcases the results obtained from experiments conducted on the ASTD dataset, excluding both the objective and neutral polarity categories. The accuracy results in this context ranged between 66% and 77%, representing an improvement compared to Table 9, where only the objective category of polarity was excluded.

Table 13. Accuracy results for ASTD dataset without objective and neutral category in a variety of cases.

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%
Stem_ISRI	76%	76%	69%	75%
Stem_ArListem	75%	74%	69%	75%
Stem_Snowball	75%	75%	68%	76%
Lemmatization	76%	76%	68%	75%
Bigram	77%	72%	66%	76%
Trigram	75%	72%	66%	77%
Tokenizing+ Remove stop word	77%	74%	69%	77%
Trigram +Remove stop word	75%	73%	67%	77%

Table 14 presents accuracy results ranging between 65% and 80%. Optimal outcomes were achieved when utilizing Tokenization and Porter Stemmer features with LR and SVM classifiers. Conversely, the worst results, with an accuracy of 39%, were observed when employing the bigram feature and parts of speech tags while selecting the action feature with the RF classifier.

Table 14. Accuracy results for ASTD dataset translated to English without objective and neutral category in a variety of cases.

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%
Stem_ Porter	79%	76%	69%	80%
Stem_Snowball	79%	76%	69%	79%
Lemmatization	79%	76%	68%	80%
Bigram	77%	73%	67%	77%
Trigram	78%	74%	66%	78%
Pos_tag "VB"	65%	64%	66%	65%
Tokenizing+ Stem_ Porter	80%	76%	69%	80%
Bigram + Pos_tag "VB"	65%	39%	65%	65%

In Table 15, the accuracy improved further, ranging between 65% and 80%. Once again, the SVM classifier emerged as the top-performing classifier.

Table 15. Accuracy results for English synonymous ASTD dataset translated without objective and neutral category in a variety of cases.

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%
Stem_ Porter	79%	77%	70%	79%
Stem_Snowball	79%	76%	70%	79%
Lemmatization	79%	76%	69%	80%
Bigram	77%	75%	67%	77%
Trigram	78%	74%	66%	78%
Pos_tag "VB"	65%	65%	65%	65%
Tokenizing + Lemmatization _	79%	76%	69%	80%
Lemmatization + Stem_ Snowball	80%	77%	70%	79%

Table 16 displayed accuracy results ranging between 66% and 77%. Although these results are similar to those in Table 13, specific experiments showed differences, with the NB classifier exhibiting the worst performance in certain scenarios.

Table 16. Accuracy results for synonymous ASTD dataset translated to Arabic without objective and neutral category in a variety of cases.

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%
Stem_ISRI	75%	74%	69%	75%
Stem_ArListem	74%	73%	68%	75%
Stem_Snowball	77%	74%	69%	77%
Lemmatization	76%	73%	70%	77%
Bigram	74%	70%	67%	74%
Trigram	72%	71%	66%	73%
Tokenizing + Stem_Snowball	77%	74%	69%	77%
Lemmatization + Stem_ ArListem	75%	74%	69%	77%

5. Discussions

The aim of the experiments was to know the effect of the pre-processing steps, and the results of the experiments were using the ASTD dataset by applying several cases, namely removing the Arabic stop words, Tokenizing, Stemming used (ISRI Stemmer, Tashaphyne, and snowball stemmer), and Lemmatization used qalsadi lemmatizer. Experiments show that the results are similar in all cases using different machine learning (ML) classifiers, but the SVM classifier has proved to be the best performing classifier. Also, the results of the experiments were similar when using two features together, as shown in the previous Table 1. Accuracy ranged between 67% and 70%.

Similar experiments were conducted using the AJGT dataset, the results ranged between 82% and 87%, and the results converged between the classifiers, and the best was the accuracy of the NB classifier. Similar results are shown by the experiments performed after the translation of the ASTD and AJGT datasets into English and applying the same previous cases and the same ML classifiers. The experiments also showed similar results with the experiments of the Arabic language dataset. However, the accuracy was slightly better applying some features on the Arabic datasets, such as the use of the stem and lemma features together in both Arabic datasets (ASTD, AJGT) is better than the translated datasets. When using three types of stemming, the results were similar in the ASTD dataset, while in the AJGT dataset, snowball stemmer was the best. The accuracy results for the ASTD dataset showed slight differences between 67% and 70% for the applied cases. The accuracy results for the translated ASTD dataset results ranged between 67% and 70%, which means that there is no significant translation improvement. On the other hand, the results for the AJGT dataset were better than the ASTD, as it ranged between 83% and 87% for the Arabic dataset, while it ranged between 82% and 86% for the translated dataset. There are minor differences between the Arabic and the translated dataset.

Furthermore, the results of experiments removing the objective category from the ASTD dataset showed a significant decline in accuracy at the levels of the Arabic dataset and the translator as shown in the previous tables. Conversely, there was an improvement in accuracy results when removing sentences classified into the objective and neutral categories.

Table 17. Summary of results of 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%
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%
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%

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

Table 17 illustrates the optimal outcomes from the sixteen experiments detailed earlier, employing the four classifiers. Across all experiments, it is evident that the SVM classifier consistently outperformed others, emerging as the top-performing classifier. However, it is noteworthy that in the experiment involving the AJGT dataset after translating the English language synonym into Arabic, the NB classifier achieved the highest accuracy result, closely converged with the SVM classifier.

Elfaik and Nfaoui [16], presented experiences of sentiment analysis in previous studies for the Arabic language using a variety of datasets, including ASTD. Where the used of Alayba *et al*. [3] machine learning algorithms: NB, SVM, LR and convolutional deep learning algorithms such as CNN. With the use of different features in the experiments (TF, TFIDF, POS, Lex, and Auto-Lex) and the choice of the Word2Vec model, the accuracy results ranged between 85% and 90% in the experiments of the total datasets, while the accuracy results improved when choosing a subset that constituted 85% of the main datasets and reached 95 %.

Baly [10], used the SVM and RNTN algorithms to study sentiment analysis and compare the results using

three Arabic dialects (Egyptian, Gulf, and Levantine) with identifying features in both algorithms (n-grams and lemma in addition to the commonly used baseline, and raw words) and the best results were The accuracy using the SVM all, lemmas algorithm 51.7%, while the RNTN algorithm achieved better results by 58.5%.

Heikal *et al*. [20], used the CNN and LSTM algorithms with determining the required parameter values for each model using the Word2Vec technique to represent the pre-trained words. The best results of accuracy were in the experiments of the CNN algorithm with adjusting coefficient values by 64.30%, while the best results of accuracy using LSTM were 64.75%. And by using ensemble modeling for the best model of the two algorithms CNN and LSTM, the accuracy results were 65.05%. Accordingly, the results showed a significant improvement compared to the results of the RNTN model in the previous study.

Dahou *et al*. [14], used the CNN algorithm and the word embedding model to represent the previously trained words in the form of vectors, where the quality of the vectors is evaluated based on word analogy questions and to determine the relationships between word pairs because there is a common relationship between them and to predict the identity of the missing word among the words. Based on algebraic arithmetic using a similarity measure such as cosine measure to discover word identity. The results of precision experiments using CNN for an unbalanced trained dataset yielded 79.07% better than the results of the balanced dataset, which was 75.9%.

Also Al-Azani and El-Alfy [5], used CNN and LSTM algorithms and other combined algorithms from these two algorithms as (CNN-LSTM, Stacked-LSTM, Combined-LSTM-SUM, Combined-LSTM-MUL, and Combined-LSTM-CONC) by conducting several experiments that consist of using two CBOW models and SG to include words and compare results for static and dynamic words. The best accuracy results were in the Combined-LSTM-MUL model for dynamic words, and it was 81.63% using CBOW word embedding. Also, in the SG model, the best accuracy was 80.42% for the Combined-LSTM-CONC model for dynamic words.

Alayba *et al*. [4], combined the CNN and LSTM algorithms together and conducted several experiments that consisted of three levels to extract various features from short sentences. The levels were first at the character level, which means converting each word in the sentence into characters and thus obtaining many features. The features can also be expanded through the second level, the second is the character _NGram level (char5Gram), which means measuring the average length of the words that make up the dataset, which averages most words 5 characters and words that exceed 5 characters are divided into sub-words, while the third level is the level of words and divides the sentence into words using the spaces between them. The accuracy results for the ASTD dataset of 2,479 tweets, including 1,684 negative and 795 positive, according to the three levels were as follows: 74.19%, 77.62%, and 76.41%, respectively. The best measure of accuracy is using char5gram level.

Hawalah [19], apply different N-gram features like Unigram, Bigram, and Trigram individually and combines two features together to calculate the accuracy of machine learning algorithms like SVM, NB, LR, Linear SVM, RBF, and MLP. The best results were using the Unigram feature, followed by the use of the Unigram + Bigram features in the MLP model, where the accuracy was 75.47% and 75%, respectively, while the accuracy was worse using the Trigram feature because it contained noise that affected the performance, where the accuracy for the same model reached 68.41%.

The results of accuracy in experiments vary according to the dataset size, the applied features, and the classifiers used for sentiment analysis in the Arabic language.

6. Conclusions

This research presented several experiments by used two datasets (ASTD, AJGT) in Arabic and English with their translated versions using Python and Google Translate Library in the field of sentiment analysis. Experiments included the use of one feature separately as well as the use of two features together for each classifier. Where four classifiers of machine learning were used (LR, RF, NB, and SVM classifiers). The accuracy of the SVM classifier was the best using ASTD dataset, while the NB classifier gave better accuracy using AJGT dataset.

The research progress is still ongoing in the field of sentiment analysis using different algorithms to improve results and obtain the best accuracy. Future plans include investigating the integration of additional extrinsic knowledge bases and lexicons in the feature extraction and engineering process. The goal is to leverage such resources to improve sentiment classification tasks in both Arabic and English, exploring the impact of Part-Of-Speech (POS) tags of concepts and their synonyms on the overall quality of the sentiment analysis approach.

References

[1] Abo M., Shah N., Balakrishnan V., and Abdelaziz A., "Sentiment Analysis Algorithms: Evaluation Performance of the Arabic and English Language," *IEEE Expert*, pp. 1-5, 2018. doi:10.1109/ICCCEEE.2018.8515844.

[2] Al Shamsi A., Bayari R., and. Salloum S., "Sentiment Analysis in English Texts," *Advances in Science Technology and Engineering Systems Journal*, vol. 5, pp. 1683-1689, 2021. Doi:10.25046/aj0506200.

[3] Alayba A., Palade V., England M., and Iqbal R., "Improving Sentiment Analysis in Arabic Using

Word Representation," *in Proceedings of the IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition*, London, pp. 13-18, 2018. doi: 10.1109/ASAR.2018.8480191

[4] Alayba A., Palade V., England M., and Iqbal R., "A Combined CNN and LSTM Model for Arabic Sentiment Analysis," *in Proceedings of Machine Learning and Knowledge Extraction: 2nd IFIP TC5, TC8/WG8.4, 8.9, TC12/WG12.9 International Cross-Domain Conference, CD-MAKE*, Hamburg, pp. 179-191, 2018. https://doi.org/10.1007/978-3-319-99740-7_12

[5] Al-Azani S. and El-Alfy E., "Hybrid Deep Learning for Sentiment Polarity Determination of Arabic Microblogs," *International Conference on Neural Information Processing*, Guangzhou, pp. 491-500, 2017. https://doi.org/10.1007/978-3-319-70096-0_51

[6] Ali N., Hamid M., and Youssif A., "Sentiment Analysis for Movies Reviews Dataset Using Deep Learning Models," *International Journal of Data Mining and Knowledge Management Process*, vol. 9, no. 2/3, pp. 19-27, 2019. https://ssrn.com/abstract=3403985

[7] Almaghrabi M. and Chetty G., "Improving Sentiment Analysis in Arabic and English Languages by Using Multi-Layer Perceptron Model (MLP)," *in Proceedings of IEEE 7th International Conference on Data Science and Advanced Analytics*, Sydney, pp. 745-746, 2020. doi: 10.1109/DSAA49011.2020.00095

[8] Alomari K., ElSherif H., and Shaalan K., "Arabic Tweets Sentimental Analysis Using Machine Learning," *in Proceedings of International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Arras, pp. 602-610, 2017. DOI: 10.1007/978-3-319-60042-0_66

[9] Alrefai M., Faris H., and Aljarah I., "Sentiment Analysis for Arabic Language: A Brief Survey of Approaches and Techniques," *International Journal of Advanced Science and Technology*, vol. 119, pp. 13-24, 2018. DOI:10.14257/ijast.2018.119.02

[10] Baly R., Badaro G., El-Khoury G., Moukalled R., and Aoun R., "A Characterization Study of Arabic Twitter Data with A Benchmarking for State-Of-The-Art Opinion Mining Models," *in Proceedings of the 3rd Arabic Natural Language Processing Workshop, EACL*, Valencia, pp. 110-118, 2017. DOI:10.18653/v1/W17-1314

[11] Barhoumi A., Aloulou C., Camelin N., Estève Y., and Belguith L., "Arabic Sentiment Analysis: An Empirical Study of Machine Translation's Impact," *in Proceedings of Language Processing and Knowledge Management International Conference*, Sfax, pp. 1-11, 2018. https://hal.science/hal-02042313

[12] Başarslan M. and Kayaalp F., "Sentiment Analysis with Machine Learning Methods on Social Media," *Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 9, pp. 5-15, 2021. DOI:10.14201/ADCAIJ202093515

[13] Boudad N., Faizi R., Rachid O., and Chiheb R., "Sentiment Analysis in Arabic: A review of the Literature," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 2479-2490, 2017. https://doi.org/10.1016/j.asej.2017.04.007

[14] Dahou A., Xiong S., Zhou J., Haddoud M., Duan P., "Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification," *in Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, pp. 2418-2427, 2016. https://aclanthology.org/C16-1228.pdf

[15] El-Awady R., Barakat S., and Elrashidy N., "Sentiment Analysis for Arabic and English Datasets," *International Journal of Intelligent Computing and Information Science*, vol. 15, no. 1, 2015. DOI:10.21608/ijicis.2015.10911

[16] Elfaik H. and Nfaoui E., "Deep bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 395-412, 2020. DOI:10.1515/jisys-2020-0021

[17] El-Masri M., Berardinelli N., and Ahmed H., "Successes and challenges of Arabic Sentiment Analysis Research: A Literature Review," *Social Network Analysis and Mining*, vol. 7, no. 54, 2017. https://doi.org/10.1007/s13278-017-0474-x

[18] Fouad M., Mahany A., Aljohani N., Abbasi R., and Hassan S., "ArWordVec: Efficient Word Embedding Models for Arabic Tweets," *Soft Computing*, vol. 24, 2020. https://doi.org/10.1007/s00500-019-04153-6

[19] Hawalah A., "A Framework for Arabic Sentiment Analysis Using Machine Learning Classifiers," *Journal of Theoretical and Applied Information Technology*, 2019. https://hal.science/hal-02300717/file/Framework-arabic.pdf

[20] Heikal M., Torki M., and El-Makky N., "Sentiment Analysis of Arabic Tweets Using Deep Learning," *Procedia Computer Science*, vol. 142, pp. 114-122, 2018. https://doi.org/10.1016/j.procs.2018.10.466

[21] Maree M., Eleyat M., Rabayah S., and Belkhatir M., "A Hybrid Composite Features Based Sentence Level Sentiment Analyzer," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 1, pp. 284-294, 2023. http://doi.org/10.11591/ijai.v12.i1.pp284-294

[22] Mohammad S., Salameh M., and Kiritchenko S., "Sentiment Lexicons for Arabic Social Media," *in Proceedings of the 10th International Conference on Language Resources and Evaluation*, LREC, pp. 33-37, 2016. https://aclanthology.org/L16-

1006.pdf
[23] Nabil M., Aly M., and Atiya A., "ASTD: Arabic Sentiment Tweets Dataset," *in Proceedings of the Empirical Methods in Natural Language Processing Conference*, Lisbon, pp. 2515-2519, 2015. DOI:10.18653/v1/D15-1299
[24] Oussous A., Benjelloun F., Lahcen A., and Belfkih S., "ASA: A Framework for Arabic Sentiment Analysis," *Journal of Information Science*, vol. 46, no. 4, pp. 544-559, 2020. DOI: 10.1177/0165551519849516
[25] Soliman A., Eissa K., and El-Beltagy S., "AraVec: a Set of Arabic Word Embedding Models for Use in Arabic NLP," *Procedia Computer Science*, vol. 117, pp. 256-265, 2017. https://doi.org/10.1016/j.procs.2017.10.117

Mohammed Maree received the Ph.D. degree in Information Technology from Monash University. He has published articles in various high-impact journals and conferences, such as ICTAI, Knowledge-Based Systems, IEEE Access, Behaviour and Information Technology, Journal on Computing and Cultural Heritage, Information Development and the Journal of Information Science. He is also a Committee Member/Reviewer of several conferences and journals, such as the World Wide Web, Computational Intelligence, and Expert Systems journals. He has supervised a number of Master's and PhD students in the fields of knowledge engineering, data analysis, information retrieval, natural language processing, and hybrid intelligent systems. He began his career as the Manager of Research and Development at gSoft Technology Solution Inc. Then, he worked as the Director of Research and QA with Dimensions Consulting Company. Subsequently, he joined the Faculty of Engineering and Information Technology (EIT), Arab American University, Palestine (AAUP), as a full-time Lecturer. From September 2014 to August 2016, he was the Head of the Multimedia Technology Department, and from September 2016 to August 2018, he was the Head of the Information Technology Department. Subsequently, he was appointed as the Assistant to the Vice President for Academic Affairs at the Arab American University from August 2021 to July 2023. In addition to his work at AAUP, he worked as a Consultant for SocialDice and Dimensions Consulting companies. Dr. Mohammed is currently an Associate Professor of Information Technology and the Dean of Faculty of Information Technology at the Arab American University.

Mujahed Eleyat is an assistant professor of computer science at the Arab American University (AAUP) in Palestine. He obtained a Ph.D scholarship in Norway and received his Phd from Norwegian University of Science and Technology in 2014. During his Ph.D study, he worked as an employee in a Norwegian company called miraim as for three years and did research in the field of high performance computation and gas flow networks. Before that, he obtained a scholarship from USA, called the presidential scholarship, to study at the university in Arkansas where he received his Master in Computer Science. In addition to teaching at AAUP for more than 10 years, Dr. Eleyat had also been the head of the Department of Computer Systems Engineering for 6 years and the assistant of the academic vice president for one year. In addition, He has been the Dean of the Faculty of Engineering and Information Technology since august 2019. Moreover, Dr. Eleyat is a member of high performance and embedded architecture and compilation (Hipeac) and his areas of expertise include High Performance Computing, Embedded Systems, and Natural Language Processing.

Enas Mesqali received a B.S. degree in an Information and communications technology from Al-Quds Open University, Jenin, Palestine, in 2013 . She is currently pursuing a master degree in computer science at the Arab American University, Jenin, Palestine (AAUP).