Arabic Sentiment Analysis for Student Evaluation using Machine Learning and the AraBERT Transformer

Huda Alamoudi
College of Computer Science and Engineering, Department of Computer Science and Artificial Intelligence, University of Jeddah, Saudi Arabia
halamoudi0045.stu@uj.esu.sa

Nahla Aljojo
College of Computer Science and Engineering, Department of Information System and Technology, University of Jeddah, Saudi Arabia
nmaljojo@uj.edu.sa (corresponding author)

Asmaa Munshi
College of Computer Science and Engineering, Cybersecurity Department, University of Jeddah, Saudi Arabia
ammunshi@uj.edu.sa

Abdullah Alghoson
College of Computer Science and Engineering, Department of Information System and Technology, University of Jeddah, Saudi Arabia
alghoson@uj.edu.sa

Ameen Banjar
College of Computer Science and Engineering, Department of Information System and Technology, University of Jeddah, Saudi Arabia
abanjar@uj.edu.sa

Araek Tashkandi
College of Computer Science and Engineering, Department of Information System and Technology, University of Jeddah, Saudi Arabia
astashkandi@uj.edu.sa

Anas Al-Tirawi
College of Engineering, Computing and Design, Department of Computer Science, Dar Al-Hekma University, Saudi Arabia
atirawi@dah.edu.sa

Iqbal Alsaleh
Faculty of Economic and Administration, Management Information System Department, King Abdulaziz University, Saudi Arabia
ealsaleh@kau.edu.sa

Received: 31 August 2023 | Revised: 13 September 2023 | Accepted: 18 September 2023

 DOI: https://doi.org/10.48084/etasr.6347

ABSTRACT

Recently, Sentiment Analysis (SA) has become a crucial area of research as it enables us to gauge people's opinions from various sources such as student evaluations, social media posts, product reviews, etc. This paper aims to create an Arabic dataset derived from student satisfaction surveys conducted at the University of Jeddah regarding their subjects and instructors. In addition, this study presents an evaluation of classical machine learning models such as Naive Bayes, Support Vector Machine, Logistic Regression, Decision Tree, and Random Forest classifier for Arabic SA, whereas the results are compared using various metrics. Furthermore, AraBERT was used for the pre-trained transformer to improve the performance, achieving an accuracy of 78%. The paper fills the lack of SA research in the education domain in the Arabic language.

Keywords-sentiment analysis; natural language processing; machine learning; pre-trained transformer

I. INTRODUCTION

Sentiment Analysis (SA) is a Natural Language Processing (NLP) application that analyzes the words in a sentence using linguistic and textual assessment to classify the sentiments, often into positive, negative, or neutral [1]. SA approaches can be classified into two main classes [2, 3]: The Lexicon-Based Approach and the Machine Learning-Based Approach. The Lexicon-Based Approach uses a sentiment dictionary to classify the sentiment, which is a collection of a group of lexical units accompanied by their emotional orientation. The Machine Learning approach uses machine learning algorithms such as Naïve Bayes (NB), Support Vector Machine (SVM), and Logistic Regression (LR). Furthermore, there have been some studies that have explored hybrid approaches, combining lexicon and corpus techniques [4]. SA is useful for many applications, including social media, customer reviews, education, instruction evaluations, etc.

Arabic is a national language and it differs from other languages in structure. It is morphologically rich, and has multiple dialects, which makes it more difficult for NLP and requires some pre-processing. The Arabic language is written from right to left and has 28 letters, including 3 vowels. Also, it has diacritics which are used to distinguish between words that have the same letters but differ in meaning. For example, the word "علم" has more than one meanings, "عِلْم" that means science, "عَلَم" that means flag, or "عُلِم" that means known [5]. Also, some pronouns are related to the word, such as "سيعلمونه" that means "they will teach him". For that, there are some challenges in Arabic language processing which are summarized in [6].

In this work, we employ machine learning algorithms namely SVM, NB, LR, Decision Tree (DT), and Random Forest (RF). Also, we apply the AraBERT model to improve the result of Arabic SA on our dataset that is collected from the students of the University of Jeddah.

II. RELATED WORKS

There are several research studies of SA in different languages and techniques. We summarize some of them in this section.

A. Classical Machine Learning Approaches

Author in [7] employed Discriminative Multinomial Naïve Bayes (DMNB) to improve the performance of Arabic SA and compared the result with other machine learning algorithms in the same dataset. The dataset contains 2,000 Arabic tweets classified as positive or negative. The accuracy is 87.5%, which is better than that of the other algorithms. The aim in [8] is to create a dataset from a student survey written in Spanish and classify it into positive, negative, and neutral. The dataset contains 2,925 rows. The authors use two learning algorithms: SVM and LR with different feature extraction techniques. They found the best result, which is 72.6%, when they used an LR model with bi-grams and tri-grams. Authors in [9] trained an SVM classifier for students' feedback documents in mixed languages (Indonesian and English). The dataset was created from student surveys. There were 636 documents annotated as positive, negative, and neutral. The accuracy of this work is 74%. Additionally, within [10], the Amazon platform was employed to conduct SA on product reviews. This analysis utilized LR and DT algorithms on a dataset consisting of 4,960 reviews for Titan Men watches written in English, with the objective of classifying them as either positive or negative. The results indicated an accuracy of 99% for DT and 94% for LR.

B. Deep Learning Approaches

There are several studies that use Deep Learning (DL) algorithms. Authors in [11] compared Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), and used them together. They were applied on the semEval dataset, which contains 32,000 English tweets. The results of the CNN model were similar to the results of LSTM, but the results were better when they were used together. Authors in [12] used Deep Neural Networks (DNNs) for SA and product review classification. They collected more than 5,000 reviews from Bangla e-commerce sites written in Bangla dialect, which were annotated as good or bad and as complaint, recommended, wrong delivery, and appreciation. The achieved accuracy was 84% for SA and 69% for product classification. Authors in [13] introduced a novel approach known as the Adaptive Particle Grey Wolf Optimizer with Deep Learning-based Sentiment Analysis (APGWO-DLSA) to effectively categorize sentiments within product reviews. This method was applied to two distinct datasets: Cell Phones And Accessories (CPAA) and Amazon Products (AP), both aimed at distinguishing between positive and negative sentiments. In CPAA dataset, consisting of 100,000 reviews, their method yielded an impressive accuracy of 94.77%, while in the AP dataset, comprising of 14,000 reviews, it achieved an accuracy of 85.31%.

Regarding the Arabic language, authors in [14] applied LSTM to enhance the prediction accuracy for Arabic SA. They used the Large-Scale Arabic Book Reviews (LABR) dataset, which contains 14,448 reviews labeled as 0 and 1 for negative

and positive. The accuracy was 82%. Some researchers applied machine learning and DL algorithms [15, 16]. SVM was used as a baseline in [15] and was compared with four DL models (LTSM, LTSM +CNN, GRU, and GRU+CNN). The authors aimed to identify Arabic hate speech on Twitter. They collected 11,000 tweets labeled into five categories (none, religious, racial, sexism, or general hate). The best accuracy achieved was 72% when using CNN with LTSM. In [16], the author analyzes instructor evaluation reviews for SA. The author uses five machine learning algorithms: NB, SVM, LR, k-nearest-neighbors (KNN), and RF along with various ensemble learning techniques. He also employed five DL algorithms: CNN, RNN, bidirectional RNN-AM, GRU, and LSTM, with different word embeddings. The dataset used contained 154,000 reviews that were collected from popular instructor review websites and were labeled as positive or negative. The best accuracy was 98.29%, when using RNN with GloVe word embedding. Authors in [17] applied NB, LR, and SVM, which achieved the highest accuracy of 63%. They improved the accuracy to 70% by applying LSTM, which is a DL algorithm, using an Arabic dataset containing 32,186 tweets that were classified as positive, negative, and neutral.

C. The AraBERT Pre-trained Transformer

AraBERT is the first Arabic transformer model. It was built based on the BERT transformer model that was developed by Google. The AraBERT model was trained on a huge dataset of the modern standard Arabic language and various Arabic dialects, and it was evaluated on several downstream tasks, including SA, Named Entity Recognition, and Question Answering, which has helped advance the field of Arabic NLP [18]. AraBERT is designed to overcome the challenges of Arabic language processing, such as the presence of dialects, multiple negations, and compound words. It is able to understand the context and meaning of Arabic text, which makes it valuable for businesses and organizations seeking to understand their customers' opinions and emotions. It is a powerful tool in the field of Arabic NLP, which has been shown to achieve state-of-the-art results on various Arabic NLP tasks. Recently, BERT and AraBERT have been used in several NLP and SA researches, e.g. in [19] the authors built their Arabic BERT approach based on the BERT model to overcome challenges related to the Arabic language. They applied it on five different datasets and compared the results with classical machine learning and DL algorithms that have been used in the same datasets in previous works. Similarly, in [20], several machine learning algorithms and AraBERT were employed on various datasets from different domains, including restaurant, movies, and product reviews. Additionally, authors in [21] utilized the AraBERT model to analyze the comments and reviews on different websites using the ARev dataset, which contains 40,000 Arabic reviews classified as positive or negative. Their approach achieved an accuracy of 92.5%

III. PROPOSED METHODOLOGY

In this paper, we present machine learning algorithms that were previously employed on an Arabic dataset to classify sentiments into positive, negative, or neutral. However, the performance of these algorithms falls short when compared to other works, due to the unique nature of the Arabic language. To address this issue, we employ the use of the pre-trained AraBERT model, which has been fine-tuned specifically for the Arabic language. The proposed methodology is outlined in Figure 1. We collected the dataset from a student survey, then we applied preprocessing to clean the data. After that, we employed the machine learning algorithms SVM, NB, LR, DT, and RF along with the AraBERT pre-trained model. Finally, we compared the performances of the applied models based on accuracy, precision, recall, F1-score and ROC-AUC.

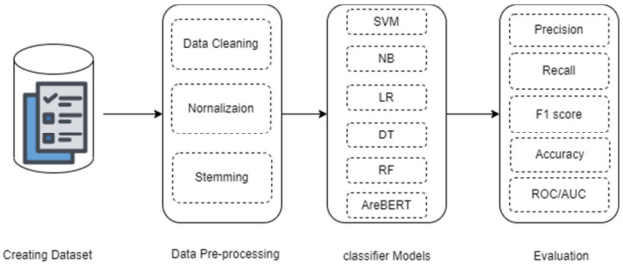

Fig. 1. The proposed methodology.

A. Dataset

We created the dataset from a student survey that was filled out at the end of every semester by the students at the University of Jeddah. The survey is provided by the university system and it is a requirement for academic accreditation. There are three steps in creating our dataset: Text Extraction, Data Cleaning, and Annotating. The student survey contains two sections: subject evaluation and instructor of the subject evaluation. Each section has two types of questions: open and closed questions. Questions and answers are written in the Arabic language. Firstly, we extracted the text to build our dataset from the answers to open questions. The open questions for the subject evaluation are:

- What do you like about this subject?
- What you do not like about this subject?
- What are your suggestions to improve this subject?

While the open questions for instructor evaluation are:

- What aspects of instructor's activity helped students learn?
- What aspects of instructor's activity needs to improve to help students learn?

We received 1,044 responses from students of the University of Jeddah. Hence, the number of comments we got is 5,220.

Many of the comments are not useful, such as blank fields, opaque letters, etc. We removed these comments from the dataset and its size became 3,472 rows after cleaning. Additionally, there are a few non-Arabic comments that were translated into Arabic. Finally, each row in the dataset was manually labeled as positive, negative, or neutral by native Arabic speakers. The distribution of our dataset was: 1,651 positive, 1,270 negative, and 551 neutral comments, as shown in Figure 2. Table I provides a sample of the dataset.

TABLE I. SAMPLE OF OUR DATASET

Negative	الماده جدا صعبه واللغه جديده على معظم الطلاب وتحتاج جهد واحنا اغلبنا خريجين وساعاتنا مليانه وفوق كدا مافي تشجيع من الاستاذهتهدد بالترسيب والحرمان وتحطم الطالبة في نطقها لو جاوبت غلط ودايما تهزئ في مستوى الطالبات
Negative	تنقيص الدرجات و عدم مراعات مشاعر الطلابات
Positive	جميع الاشياء كويسة
Positive	اسلوب الاستاذة ساعد ف فهم المقرر كثيرا

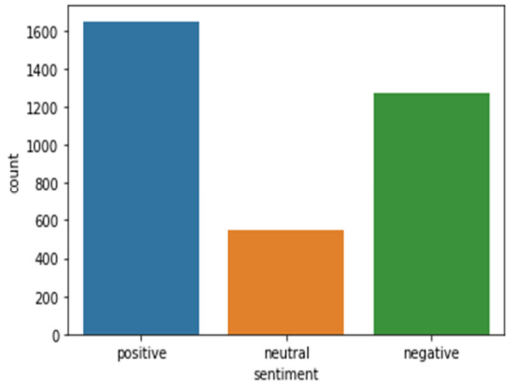

Fig. 2. The dataset classes.

B. *Pre-processing*

Data pre-processing is an important step in NLP that cleans all noisy characters from the text and additions in some words that negatively affect the results. It is done through removing noise, normalization, and stemming. The removing list consists of:

- Punctuations and special characters.
- Numbers.
- English characters and repeat characters.
- Stop words like (أما ، إن ، اياك، اذا).

Arabic Normalization: standardizes the spelling of some letters like Alef and Hamza by replacing all (آ, إ, أ) into (ا), (ؤ, ئ) into (ء) and others.

Additionally, during the pre-processing step, stemming was applied to each word. Stemming is the process of removing suffixes/prefixes and reducing words to their root or base form. For example, the word "حركات" becomes "حرك" after word stemming by using the ISRIStemmer stemming algorithm.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

We split the dataset into 80% and 20% for training and validation, respectively. We employed five machine learning classifiers and the AraBERT model using the scikit-learn library for machine learning and AraBERT library. TF-IDF feature extraction was applied that refers to term frequency–inverse document frequency. It calculates a weight for each word that shows the importance of this word based on the number of times it appears. We used the standard performance metrics precision, recall, f-score, and accuracy to evaluate our work:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1 = 2 \times \frac{Recall \times Precision}{Recall+Precision} \quad (3)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

where TP stands for True Positive, FP for False Positive, TN for True Negative, and FN for False Negative.

In addition, the performance of the models was evaluated using the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC). The ROC curve provides a visual representation of the performance of a classification model across all classification thresholds by plotting the True Positive Rate (TPR) and False Positive Rate (FPR), which are calculated by (5) and (6). The AUC quantifies the entire two-dimensional area under the ROC curve as a metric of performance.

$$TPR = \frac{TP}{(TP + FN)} \quad (5)$$

$$FPR = \frac{FP}{FP+TN} \quad (6)$$

A. *Re-sampling Dataset*

When there is unequal distribution of the classes in a dataset, it is called imbalanced. This can affect the results of machine learning models. There are several methods to handle this problem including Over-Sampling and Under-Sampling [22]. Over-sampling involves adding some samples to the minority class, while under-sampling involves removing samples from the majority class. We used these two techniques to obtain a balanced dataset. We applied the Random Over-Sampling on the "neutral" class, which originally had 551 samples and it and at the end 1,649 samples. Then, we applied the Random Under-Sampling for "neutral" and "positive" classes. Their size became 1,271. Figure 3 shows the dataset classes before and after re-sampling. We encode the target variable so the label becomes [0: "negative", 1: "neutral", 2: "positive"].

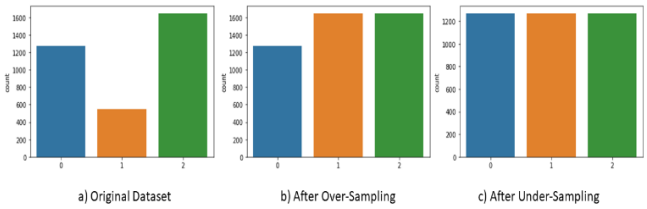

Fig. 3. Dataset before and after re-sampling.

B. *Machine Learning Algorithms*

1) *Support Vector Machine*

SVM is a classification algorithm for linear and nonlinear data that builds a decision boundary (hyperplane) to classify two classes [23]. The SVM algorithm's formula can be represented mathematically as an optimization problem: Given a set of training data (x_i, y_i), $i = 1, 2, ..., n$, where x_i is a feature vector and y_i is the corresponding label, the goal is to find the optimal hyperplane that separates the data into classes. This is done by minimizing the following objective function [24]:

$$min\frac{1}{2}||w||^2 + C * sum(max(0.1 - y_i * (w.x_i + b))) \quad (7)$$

where $\|w\|^2$ is the square of the Euclidean norm of the weight vector w, C is the regularization parameter, which determines the trade-off between maximizing the margin and minimizing the training errors (we set C=1 in our experiment), $w.x_i + b$ is the linear equation of the hyperplane, $y_i * (w.x_i + b)$ is the decision function, which returns positive values for samples that lie on the one side of the hyperplane and negative values for samples that lie on the other side, and max(.) is the hinge loss function, which is used to penalize samples that are misclassified. The resulting hyperplane can be used to make predictions on new data. The result of our SVM experiment is shown in Table II.

TABLE II. CLASSIFICATION REPORT OF SVM

	Imbalanced Data			Balanced Data		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Neutral	0.61	0.41	0.49	0.64	0.72	0.68
Negative	0.73	0.73	0.73	0.69	0.61	0.64
Positive	0.74	0.83	0.78	0.64	0.64	0.64
Accuracy	0.72			0.66		

2) Naïve Bayes

NB is based on the Bayes theorem that is the probability of the existence of a class based on a given text. It is a supervised learning algorithm used for classification problems [25]. There are more than one versions of NB which are MultinomialNB, BernoulliNB, CategoricalNB, ComplementNB, and GaussianNB. We used MultinomialNB because it most suitable for text classification. Let's consider a text comment x with N words and let's assume there are 3 classes in our sentiment analysis problem. For a given comment x and a class y, the probability that the comment belongs to class y is calculated as follows:

$$P(y|x) = \frac{P(x|y) * P(y)}{P(x)} \quad (8)$$

where $P(y|x)$ is the posterior probability of class y given the text x. $P(x|y)$ is the likelihood of observing the text x given class y. $P(y)$ is the prior probability of class y. $P(x)$ is the evidence, which is the probability of observing the text x regardless of the class. It is applied to each class to calculate the posterior probability for each class, and the class with the highest probability can be selected as the predicted class for the document x. Table III shows the result of our experiment for MultinomialNB.

TABLE III. CLASSIFICATION REPORT OF MULTINOMIALNB

	Imbalanced Data			Balanced Data		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Neutral	0.72	0.20	0.31	0.60	0.74	0.66
Negative	0.71	0.69	0.70	0.72	0.57	0.64
Positive	0.67	0.85	0.75	0.66	0.65	0.66
Accuracy	0.68			0.65		

3) Logistic Regression

LR is a linear classification that is a statistical model for solving classification problems [26]. It aims to learn the relationship between the data. For a given comment as a vector feature x and its sentiment y, the LR model is defined as:

$$p(y = 1|x) = \frac{1}{(1 + exp(-z))} \quad (1)$$

$$z = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n \quad (2)$$

where $p(y = 1|x)$ is the predicted probability that $y = 1$, given x, z is the linear combination of features and coefficients, and W_i is the weight assigned to feature x_i.

The goal of training an LR model is to learn the weights w_i that maximize the likelihood of the observed data. The prediction for a new sample is made by computing the predicted probability $p(y = 1|x)$, and the class label is assigned as 1 if $p(y = 1|x)$ is greater than or equal to 0.5. Also in scikit-learn, it has a C parameter like in SVM. We set it equal to [0.01, 0.1, 1, 10, 100] and then find the optimal hyperparameter using GridSearchCV. The result of LR is shown in Table IV.

TABLE IV. CLASSIFICATION REPORT OF LR

	Imbalanced Data			Balanced Data		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Neutral	0.67	0.37	0.48	0.64	0.88	0.71
Negative	0.73	0.69	0.71	0.77	0.63	0.70
Positive	0.71	0.85	0.77	0.67	0.64	0.65
Accuracy	0.71			0.69		

4) Decision Tree

DT is useful for classification problems. It aims to predict the values by simple decision rules [27]. It works by recursively dividing the data into smaller subsets based on the values of the input features, until the subsets contain only samples of the same class or the samples are homogeneous with respect to the target variable. The tree structure is used to represent the decisions and their consequences. A DT can be represented by a series of decision rules, each represented by a node in the tree. At each node, a test is performed on one of the input features, and based on the test results, the sample is either sent to the left or right child node. The process is repeated until a terminal node (also known as a leaf node) is reached. We set all hyperparameters to their default values except for 'criterion,' which is set to 'entropy' to measure the quality of a split. Table V shows the results.

TABLE V. THE CLASSIFICATION REPORT OF DT

	Imbalanced Data			Balanced Data		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Neutral	0.45	0.47	0.46	0.65	0.88	0.75
Negative	0.64	0.63	0.63	0.69	0.56	0.62
Positive	0.72	0.71	0.71	0.64	0.54	0.58
Accuracy	0.64			0.66		

5) Random Forest

RF algorithm is used for classification problems. It consists of multiple DTs, and it can improve the accuracy of the DT algorithm [28]. The basic idea behind RF is to randomly sample the training data with replacement (bootstrapping) and fit multiple DTs to the bootstrapped samples. The final

prediction is made by taking a majority vote in the predictions of all the trees by taking a majority vote in the classification problems based on the predictions of all the trees. Table VI shows the results of random forest classifier [29].

TABLE VI. CLASSIFICATION REPORT OF RF

	Imbalanced Data			Balanced Data		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Neutral	0.60	0.45	0.52	0.72	0.89	0.80
Negative	0.73	0.72	0.73	0.77	0.60	0.68
Positive	0.74	0.81	0.77	0.68	0.68	0.68
Accuracy	0.72			0.72		

C. AraBERT

AraBERT is a pre-trained language model developed by OpenAI for NLP tasks in the Arabic language. The basic architecture of AraBERT [21] consists of multi-layer transformer blocks and includes a deep bidirectional encoding mechanism, where each word is encoded based on the context from both the left and right sides of the word. In SA, the final hidden state of the AraBERT network is typically used to predict the sentiment of a given text by passing it through a linear layer followed by a softmax activation function to generate a probability distribution over the predefined sentiment classes. The classification report of our experiments for the AraBERT model is shown in Table VII.

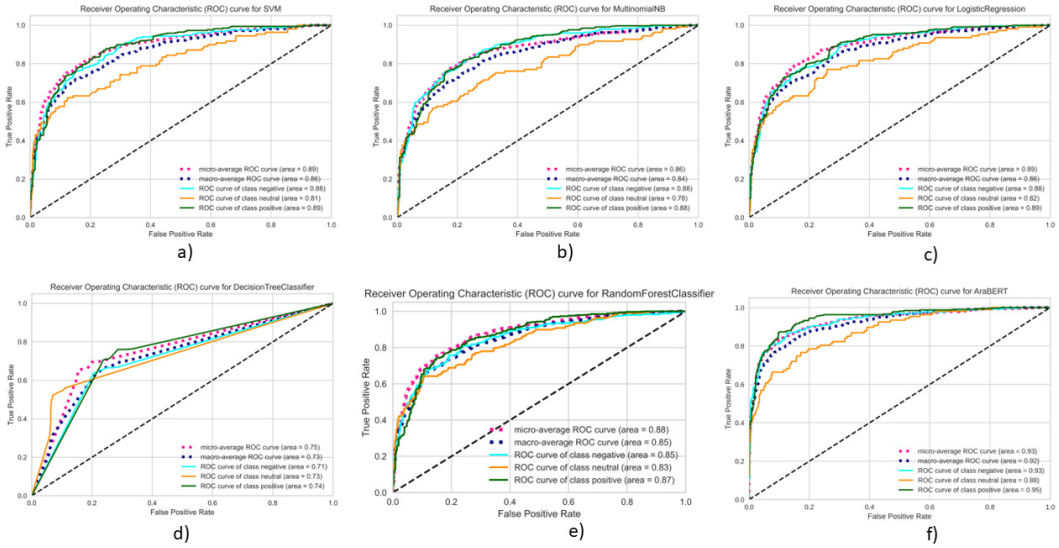

Fig. 4. ROC Curve for the imbalanced dataset: a) SVM, b) MultinomialNB, c) LR, d) DT, e) RF, f) AraBERT.

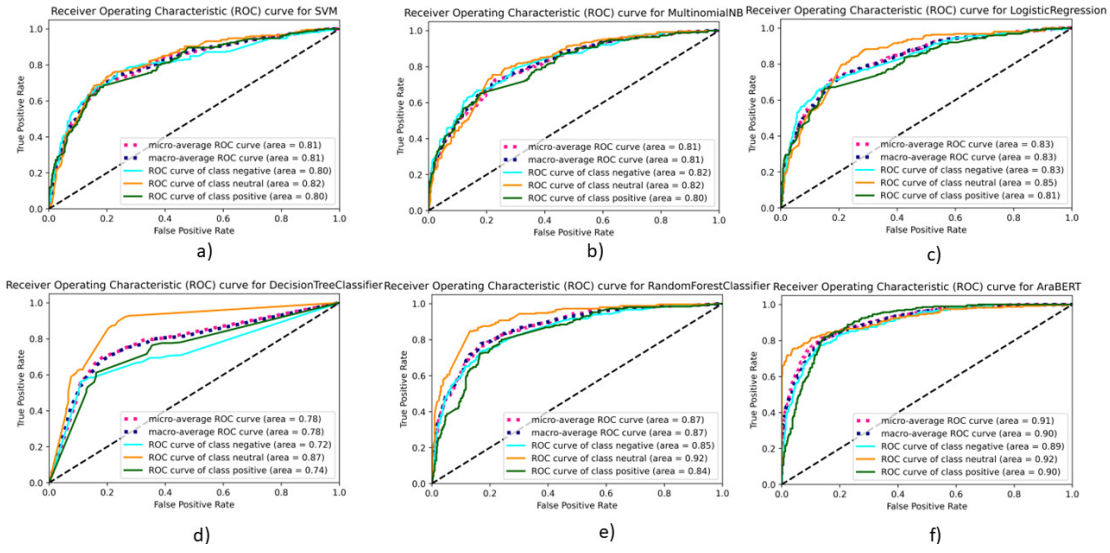

Fig. 5. ROC Curve for the balanced dataset: a) SVM, b) MultinomialNB, c) LR, d) DT, e) RF, f) AraBERT.

TABLE VII. CLASSIFICATION REPORT OF ARABERT

	Imbalanced Data			Balanced Data		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Neutral	0.74	0.47	0.57	0.87	0.77	0.82
Negative	0.80	0.85	0.82	0.77	0.80	0.78
Positive	0.85	0.90	0.87	0.79	0.84	0.81
Accuracy	0.82			0.80		

TABLE VIII. THE RESULTS OF THE BALANCED DATASET.

	Accuracy	**Precision**	**Recall**	**F1-score**	**AUC**
MultinomialNB	0.65	0.65	0.65	0.65	0.81
SVM	0.66	0.66	0.66	0.66	0.81
LR	0.69	0.70	0.69	0.68	0.83
DT	0.67	0.67	0.67	0.66	0.78
RF	0.72	0.73	0.72	0.72	0.87
AraBERT	0.80	0.81	0.80	0.80	0.92

It is a commonly held misconception that the highest accuracy equates to the best performance in a classification problem. However, accuracy alone is not always a suitable metric for evaluating the performance of a model, particularly in the case of imbalanced datasets. The classification reports presented in Tables II-VI demonstrate the results obtained from both balanced and imbalanced datasets. Although the highest accuracy was achieved in the imbalanced dataset, a closer examination of the precision, recall, and F1 scores reveals a bias towards the positive class, with the values of the neutral class being comparatively lower. This is attributed to the variance disparity between classes in the dataset. The ROC curve in Figure 4 further highlights this phenomenon, as it illustrates that the area under the curve for the neutral class is significantly lower than that of the other classes. However, the ROC curve in Figure 5, which represents the balanced dataset, showcases a convergent area under the curve for all classifier models, except for the decision tree and random forest. In this scenario, the area of the neutral class is the highest. Moreover, the balanced dataset distributes the values of precision, recall, and F1 evenly among the neutral, negative, and positive classes. Hence, the results obtained from the balanced dataset, as depicted in Table VIII, are adopted as the preferred.

Several machine learning algorithms and a pre-trained transformer were evaluated for their effectiveness in Arabic SA on both imbalanced and balanced datasets. Among the machine learning algorithms, the RF classifier achieved the highest performance with an accuracy of 72%. The use of the AraBert model led to a marked improvement in the SA results, as indicated by an accuracy of 80%.

V. CONCLUSION AND FUTURE WORK

This study conducted a comprehensive comparison of various machine learning models, including SVM, multinomial NB, LR, DT, and RF classifiers, alongside the AraBERT pre-trained model for Arabic sentiment analysis. Our analysis, based on a dataset collected from students at the University of Jeddah, comprising over 3,400 samples categorized into positive, negative, and neutral sentiments, revealed that AraBERT outperformed the other models.

As part of our future research endeavors, we aim to expand our dataset by gathering student evaluations from a diverse array of universities, encompassing different Arabic dialects. Furthermore, we are interested in exploring the incorporation of advanced Deep Learning Models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to further enhance the accuracy and applicability of SA in the Arabic language. These initiatives will help us advance our understanding of sentiment analysis in the Arabic context and contribute to the broader field of natural language processing.

REFERENCES

[1] H. Kennedy, "Perspectives on Sentiment Analysis," *Journal of Broadcasting & Electronic Media*, vol. 56, no. 4, pp. 435–450, Oct. 2012, https://doi.org/10.1080/08838151.2012.732141.

[2] F. S. Dolianiti, D. Iakovakis, S. B. Dias, S. Hadjileontiadou, J. A. Diniz, and L. Hadjileontiadis, "Sentiment Analysis Techniques and Applications in Education: A Survey," in *International Conference on Technology and Innovation in Learning, Teaching and Education*, Thessaloniki, Greece, Jun. 2018, pp. 412–427, https://doi.org/10.1007/978-3-030-20954-4_31.

[3] M. Hilario, D. Esenarro, I. Petrlik, and C. Rodriguez, "Systematic Literature Review of Sentiment Analysis Techniques," *Journal of Contemporary Issues in Business and Government*, vol. 27, no. 1, pp. 506–517, 2021.

[4] T. Alqurashi, "Arabic Sentiment Analysis for Twitter Data: A Systematic Literature Review," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, pp. 10292–10300, Apr. 2023, https://doi.org/10.48084/etasr.5662.

[5] N. Boudad, R. Faizi, R. Oulad Haj Thami, and R. Chiheb, "Sentiment analysis in Arabic: A review of the literature," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 2479–2490, Dec. 2018, https://doi.org/10.1016/j.asej.2017.04.007.

[6] H. Rahab, M. Djoudi, and A. Zitouni, "Sentiment Analysis of Arabic Documents: Main Challenges and Recent Advances," in *Natural Language Processing for Global and Local Business*, Hershey, PA, USA: IGI Global, 2021, pp. 307–331.

[7] H. AlSalman, "An Improved Approach for Sentiment Analysis of Arabic Tweets in Twitter Social Media," in *3rd International Conference on Computer Applications & Information Security*, Riyadh, Saudi Arabia, Mar. 2020, https://doi.org/10.1109/ICCAIS48893.2020.9096850.

[8] H. Newman and D. Joyner, "Sentiment Analysis of Student Evaluations of Teaching," in *International Conference on Artificial Intelligence in Education*, London, UK, Jun. 2018, pp. 246–250, https://doi.org/10.1007/978-3-319-93846-2_45.

[9] D. F. Sengkey, A. Jacobus, and F. J. Manoppo, "Implementing Support Vector Machine Sentiment Analysis to Students' Opinion toward Lecturer in an Indonesian Public University," *Journal of Sustainable Engineering: Proceedings Series*, vol. 1, no. 2, pp. 194–198, Sep. 2019, https://doi.org/10.35793/joseps.v1i2.27.

[10] M. A. Kausar, S. O. Fageeri, and A. Soosaimanickam, "Sentiment Classification based on Machine Learning Approaches in Amazon Product Reviews," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10849–10855, Jun. 2023, https://doi.org/10.48084/etasr.5854.

[11] D. Goularas and S. Kamis, "Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data," in *International Conference on Deep Learning and Machine Learning in Emerging Applications*, Istanbul, Turkey, Aug. 2019, pp. 12–17, https://doi.org/10.1109/Deep-ML.2019.00011.

[12] M. H. Munna, M. R. I. Rifat, and A. S. M. Badrudduza, "Sentiment Analysis and Product Review Classification in E-commerce Platform," in *23rd International Conference on Computer and Information Technology*, Dhaka, Bangladesh, Dec. 2020, https://doi.org/10.1109/ICCIT51783.2020.9392710.

[13] D. Elangovan and V. Subedha, "Adaptive Particle Grey Wolf Optimizer with Deep Learning-based Sentiment Analysis on Online Product Reviews," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10989–10993, Jun. 2023, https://doi.org/10.48084/etasr.5787.

[14] A. Q. Al-Bayati, A. S. Al-Araji, and S. H. Ameen, "Arabic Sentiment Analysis (ASA) Using Deep Learning Approach," *Journal of Engineering*, vol. 26, no. 6, pp. 85–93, Jun. 2020, https://doi.org/10.31026/j.eng.2020.06.07.

[15] A. Al-Hassan and H. Al-Dossari, "Detection of hate speech in Arabic tweets using deep learning," *Multimedia Systems*, vol. 28, no. 6, pp. 1963–1974, Dec. 2022, https://doi.org/10.1007/s00530-020-00742-w.

[16] A. Onan, "Mining opinions from instructor evaluation reviews: A deep learning approach," *Computer Applications in Engineering Education*, vol. 28, no. 1, pp. 117–138, 2020, https://doi.org/10.1002/cae.22179.

[17] A. Alshutayri *et al.*, "Evaluating sentiment analysis for Arabic Tweets using machine learning and deep learning," *Romanian Journal of Information Technology and Automatic Control*, vol. 32, no. 4, pp. 7–18, 2022, https://doi.org/10.33436/v32i4y202201.

[18] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding." arXiv, Mar. 07, 2021, https://doi.org/10.48550/arXiv.2003.00104.

[19] H. Chouikhi, H. Chniter, and F. Jarray, "Arabic Sentiment Analysis Using BERT Model," in *International Conference on Computational Collective Intelligence*, Rhodes, Greece, Oct. 2021, pp. 621–632, https://doi.org/10.1007/978-3-030-88113-9_50.

[20] R. A. Alsuhemi and S. M. Zarbah, "Machine Learning and AraBERT Models for Arabic Online Reviews Sentiment Analysis," *Romanian Journal of Information Technology and Automatic Control*, pp. 1–14, 2022.

[21] H. El Moubtahij, H. Abdelali, and E. B. Tazi, "AraBERT transformer model for Arabic comments and reviews analysis," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 1, pp. 379–387, Mar. 2022, https://doi.org/10.11591/ijai.v11.i1.pp379-387.

[22] "Resampling strategies for imbalanced datasets." https://kaggle.com/code/rafjaa/resampling-strategies-for-imbalanced-datasets.

[23] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," in *European Conference on Machine Learning*, Chemnitz, Germany, Apr. 1998, pp. 137–142, https://doi.org/10.1007/BFb0026683.

[24] V. Kecman, "Support Vector Machines – An Introduction," in *Support Vector Machines: Theory and Applications*, L. Wang, Ed. Berlin, Heidelberg: Springer, 2005, pp. 1–47.

[25] D. Berrar, "Bayes' Theorem and Naive Bayes Classifier," in *Encyclopedia of Bioinformatics and Computational Biology*, Amsterdam, Netherlands: Elsevier, 2018, pp. 403–412.

[26] S. T. Indra, L. Wikarsa, and R. Turang, "Using logistic regression method to classify tweets into the selected topics," in *International Conference on Advanced Computer Science and Information Systems*, Malang, Indonesia, Oct. 2016, https://doi.org/10.1109/ICACSIS.2016.7872727.

[27] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 1, pp. 20–28, Mar. 2021, https://doi.org/10.38094/jastt20165.

[28] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, "A comparison of random forest variable selection methods for classification prediction modeling," *Expert Systems with Applications*, vol. 134, pp. 93–101, Nov. 2019, https://doi.org/10.1016/j.eswa.2019.05.028.

[29] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, https://doi.org/10.1023/A:1010933404324.