Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.062478

ARTICLE

Enhancing Arabic Sentiment Analysis with Pre-Trained CAMeLBERT: A Case Study on Noisy Texts

Fay Aljomah, Lama Aldhafeeri, Maha Alfadel, Sultanh Alshahrani, Qaisar Abbas* and Sarah Alhumoud*

College of Computer and Information Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, 11432, Saudi Arabia

*Corresponding Authors: Qaisar Abbas. Email: qaabbas@imamu.edu.sa; Sarah Alhumoud. Email: sohumoud@imamu.edu.sa

Received: 19 December 2024; Accepted: 22 May 2025; Published: 30 July 2025

ABSTRACT: Dialectal Arabic text classification (DA-TC) provides a mechanism for performing sentiment analysis on recent Arabic social media leading to many challenges owing to the natural morphology of the Arabic language and its wide range of dialect variations. The availability of annotated datasets is limited, and preprocessing of the noisy content is even more challenging, sometimes resulting in the removal of important cues of sentiment from the input. To overcome such problems, this study investigates the applicability of using transfer learning based on pre-trained transformer models to classify sentiment in Arabic texts with high accuracy. Specifically, it uses the CAMeLBERT model finetuned for the Multi-Domain Arabic Resources for Sentiment Analysis (MARSA) dataset containing more than 56,000 manually annotated tweets annotated across political, social, sports, and technology domains. The proposed method avoids extensive use of preprocessing and shows that raw data provides better results because they tend to retain more linguistic features. The fine-tuned CAMeLBERT model produces state-of-the-art accuracy of 92%, precision of 91.7%, recall of 92.3%, and F1-score of 91.5%, outperforming standard machine learning models and ensemble-based/deep learning techniques. Our performance comparisons against other pre-trained models, namely AraBERTv02-twitter and MARBERT, show that transformer-based architectures are consistently the best suited when dealing with noisy Arabic texts. This work leads to a strong remedy for the problems in Arabic sentiment analysis and provides recommendations on easy tuning of the pre-trained models to adapt to challenging linguistic features and domain-specific tasks.

KEYWORDS: Artificial intelligence; deep learning; machine learning; BERT; CAMeLBERT; natural language processing; sentiment analysis; transformer

1 Introduction

In recent years, social media platforms have become essential arenas for public conversation, allowing individuals to express opinions and experiences in real time. These opinions can be on various fields, such as events, news, services, and products. That makes social media a rich source of sentiment data that could aid in sensing public trends [1,2]. Sentiment Analysis is a branch of natural language processing that categorizes the popularity of public opinion based on understanding the feelings or the purpose behind each sentence. SA can also be applied to various fields, such as business. It contributes to improving services by analyzing the opinions of its customers, and it is beneficial in politics, where it assesses public reactions to candidates and policies and helps in decision-making [3].

A significant amount of research has been conducted to enhance the accuracy of sentiment analysis methods, ranging from simple linear models to more complex deep neural network models [4]. Recently, advancements in Deep Learning (DL) and neural networks have greatly enhanced the accuracy of sentiment analysis models. One of the most impactful advancements is using Transformers in models like BERT, revolutionizing sentiment analysis [5]. Additionally, Convolutional Neural Networks (CNN) have been effectively applied to text classification tasks, helping to identify local patterns in texts that signal specific sentiments [6]. Further, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have proven highly effective for processing sequential data, such as sentences or paragraphs, making them ideal for capturing sentiment across longer texts [7]. In addition to deep learning models, traditional Machine Learning (ML) algorithms like Support Vector Machines (SVM) and Naïve Bayes classifiers still play an essential role, especially when paired with advanced feature extraction techniques like TF-IDF and word embeddings such as Word2Vec and GloVe [8]. In summary, while traditional machine learning models can be implemented well in specific techniques, deep learning models surpass ML when dealing with larger datasets and more complicated text [9].

With the advancement in the field of sentiment analysis, it has become feasible to comprehend written texts better and deduce the emotional meanings they convey [10]. However, much of the research and studies have primarily focused on commonly spoken languages such as English, making applications in Arabic content limited and insufficient [5]. Additionally, compared to English sentiment analysis, there have been relatively few studies on Arabic sentiment analysis [10]. The Arabic language is intricate due to its ambiguity and rich morphological system. This complexity, along with the scarcity of resources, annotated corpora, and the diversity of dialects, poses challenges for advancements in Arabic sentiment analysis research [4]. The Arabic language can be characterized into three primary forms: Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA). CA is utilized in religious texts, including the Qur'an and classical books. On the other hand, MSA operates with less formal vocabulary and is primarily used in formal written and spoken contexts such as news, education, and literature. Lastly, DA is primarily used in everyday communication and demonstrates significant regional variation, with over 30 distinct dialects [11].

This study aims to find suitable techniques for determining the sentiments of X's tweets (formerly known as Twitter tweets, herein defined as Twitter/X) while addressing the challenge of noisy text. Noisy text includes elements such as spelling errors, abbreviations, symbols, or slang, which make accurate classification difficult. The study utilizes the Computational Approaches to Modeling Language (CAMeL) BERT model [12]. CAMeLBERT is a deep learning model based on the BERT architecture specifically fine-tuned for Arabic. CAMeLBERT was developed as part of the CAMeL Tools project, introduced in [13], to address the linguistic challenges unique to Arabic, such as morphological richness, dialectal variations, and orthographic inconsistencies. The model has been pre-trained on CA, MSA, and various Arabic dialects, e.g., Egyptian, Levantine, and Gulf Arabic making it well-suited for sentiment analysis tasks. CAMeLBERT is employed in this research to evaluate its effectiveness in sentiment classification. The contributions of this study will contrast the results from the CAMeLBERT model with models from the study in [14] that use DL and ML models for Arabic sentiment analysis, examining their performance in analyzing Arabic. After achieving the highest accuracy with the CAMeLBERT model, the same approach was applied to the Arabertv02-twitter and MARBERT models to compare the results.

The sections in this study are organized as follows: In Section 2, the literature review on related works, which briefly describes the approach used. In Section 3, the methodology presents the dataset, cleaning and preprocessing steps, the proposed methodology on the models, the development environment, evaluation metrics, and the experimental design. In Section 4, they discuss the study results, and benchmarks. Section 5 concludes with a summary of the study and future work.

2 Literature Review

In this study, it attempts to focus on the Arabic language domain. In contrast, a significant amount of sentiment analysis research has focused on the English language. Arabic sentiment analysis has received some attention, though not with the same capacity. To aim to bridge this gap through reviewing existing literature and analyzing the methodologies employed in Arabic sentiment analysis. This study has covered 24 related works, encompassing both Arabic and English studies, and categorized them into three main sections. The first section covers research that applied both ML and DL models on sentiment analysis, including two Arabic studies and one English study, notably in addition to the benchmark study in [14]. The second section focused on machine learning models and consisted of three Arabic and two English studies. Lastly, the research section is on deep learning models, including ten studies on Arabic sentiment analysis and five on English sentiment analysis. Table 1 provides a summary of the related works in Arabic sentiment analysis.

This study benchmarked with Wazrah and Alhumoud [14] which explores sentiment classification of Gulf dialect tweets using both traditional machine learning SVM and deep learning models. Including Stacked Gated Recurrent Units (SGRU), Stacked Bidirectional GRU (SBi-GRU), and AraBERT. Utilized MARSA dataset, which is preprocessed using the Automatic Sentiment Refinement (ASR) technique to discard noisy words. The authors compare different word embeddings like AraVec, FastText, and ArabicNews, with AraVec performing the best their results show that deep learning models, particularly the SBi-GRU with five layers and the SGRU with six layers, achieve higher accuracy than SVM. The best performance comes from an ensemble model combining AraBERT, SGRU, and SBi-GRU, reaching over 90% accuracy. The ensemble approach, coupled with the ASR technique, significantly improves sentiment classification, demonstrating that combining deep learning models offers superior performance for Arabic sentiment analysis compared to traditional ML models. Moreover, the research employed both ML and DL. Alayba et al. [12] used Word2Vec with seven ML algorithms and CNNs to classify Arabic health-related tweets, improving sentiment classification accuracy to 92% on the main dataset and 95% on a subset. Alghamdi [15] employs several ML and DL models: CNN, LSTM, BERT-MINI, Logistic Regression, Random Forest, KNN, Naïve Bayes, SVM, and XgBoost. The dataset was collected from Twitter/X of Arabic tweets over six years related to the Hajj event before, during, and after. However, BERT-MINI outperformed the best DL models, while Random Forest surpassed the best ML models.

Bahja et al. [16] presented a two-stage classifier for analyzing COVID-19 Arabic tweets, achieving F1-scores of 0.85 for relevance and 0.79 for theme classification using MNB, 0.83–0.69 using SVC, and, 0.79–0.73 using Decision Tree classifier. Al-Twairesh and Al-Negheimish [17] used three Arabic datasets SemEval2017, AraSenTi, and ASTD to train their SVM model. But before that, they worked on the input representation level by utilizing their technique of combining the surface features and generic word embeddings first. Then, the SVM classifier uses these combined feature sets to perform sentiment classification. In addition, that was the highest performance with 80.38, 74.84, and 79.77, respectively. Louati et al. [18] focused on sentiment classification regarding student experiences in a tweet-COVID-19 with a real dataset provided by the deanship of quality at Prince Sattam bin Abdulaziz University (PSAU). After the preprocessing steps, they used a machine learning SVM model to classify the reviews and evaluate the result against the pre-trained CAMeLBERT model. Interestingly, the CAMeLBERT model classified 70.48% of the reviews as positive, and the result was close to the SVM model, which classified 69.62% as positive.

Ref. [19] introduced an unsupervised self-labeling framework for Arabic sentiment classification, improving accuracy by 5% TF/IDF and 2% AraBERT on the LAMD dataset, and reaching 69.50% TF/IDF and 85.38% AraBERT accuracy on the MARSA dataset.

Baali and Ghneim [20] used a Deep Convolutional Neural Network (CNN) for emotion detection of Arabic tweets, were section the emotion into four categories: joy, anger, sadness, and fear, for a dataset

provided by SemEval. Then compare the result with three machine learning algorithms Naïve Bayes, SVM, and Multilayer Perceptron. The approach achieved 99.82% validation accuracy. Khalil et al. [21] proposed to use a Bidirectional Long Short Term Memory (BiLSTM) using the SemEval2018 Task1 dataset. Then the model has been compared with SVM and Random Forest developed in the same dataset. The model achieves the best validation accuracy. Abu Farha and Magdy [22] re-implemented various approaches for Arabic sentiment analysis and conducted tests on well-known datasets like SemEval, ASTD, and ArSAS. The study found that BiLSTM and CNN-LSTM were the most effective traditional models, achieving F-scores of 0.63, 0.72, and 0.89 for BiLSTM and 0.63, 0.72, and 0.90 for CNN-LSTM on the SemEval, ASTD, and ArSAS datasets, respectively. However, transformer-based models, particularly BERT, outperformed all others, achieving F-scores of 0.69, 0.76, and 0.92 on these datasets. Baniata and Kang [23] explored three different Arabic datasets related to hotel and book reviews, HARD, BRAD, and LABR to employ their model starting by utilizing some techniques like Multi-task learning, Multi-head attention and a Mixture of expert mechanisms to engage their Switch-Transformer Sentiment Analysis (ST-SA) model. Also, the model's F1-score among the HARD, BRAD, and LABR datasets was 83.50%, 67.89%, and 82.71%, respectively, with the best performance compared to other models with the same dataset.

Al-Khalifa et al. [24] collected the data from Twitter/X using Python libraries Snscrape and Tweepy based on specific keywords related to ChatGPT. They preprocessed their dataset with several cleaning steps, such as removing punctuation, links, diacritics, emojis, and duplicates. The ArabiTools is a pre-trained model utilized to classify the tweets, and the BERTopic model used is for topic identification due to its better performance with Arabic text. It successfully identified topics related to regional discussions, controversies, scams, and sector-specific dialogues. Shah et al. [25] developed a Modified Switch Transformer (MST), which includes two mechanisms: probabilistic projections and Variational Enmesh Experts Routing. Applying the model resulted in three text classification tasks: sarcasm detection, dialect classification, and sentiment classification, resulting in the best classification regarding F1-score with 0.67. Zahidi et al. [26] conducted a comparative approach between using Word2Vec and FastText word embedding models with LSTM networks for sentiment analysis in Arabic tweets. A dataset containing 52,155 tweets was used. The results showed that the LSTM with FastText model outperformed the LSTM with Word2Vec model, achieving an accuracy of 84.14%, compared to 82.14% for the LSTM with Word2Vec.

Table 1: Summary of related works of Arabic sentiment analysis. Pos: Positive, Neg: Negative, Neu: Neutral

Study	Publish year	Models	Dataset source	Dataset size	Domains	Labels	Evaluation metrics	Results
[15]	2024	CNN, LSTM BERT-MINI Logistic regression Random forest, KNN Naïve Bayes SVM, XGBoost	The paper itself	80,000 tweets	Tweets related to Hajj	Pos, Neg, and Neu	F1-score	BERT-MINI: 93%
[23]	2024	ST-SA	HARD, BRAD, and LABR	409,562 510,598 63,257 reviews	Hotel and books reviews	Highly Pos, Pos, Neu, Neg and Highly Neg	F-Score	HARD: 83.50% BRAD: 67.89% LABR: 82.71%
[12]	2018	CNN	AHS	2026 tweets	Tweetshealth services	Pos and Neg	Accuracy	CNN: 95%
[26]	2023	MST	ArSarcasm	10,547 tweets	Tweets related to politics, sports, and entertainment	Pos, Neg, and Neu	F1-Score	MST: 67%
[18]	2023	SVM	The paper itself	1870 reviews	PSAU course reviews	Pos, Neg, and Neu	Accuracy	SVM: 84.7%

(Continued)

Table 1 (continued)

Study	Publish year	Models	Dataset source	Dataset size	Domains	Labels	Evaluation metrics	Results
[22]	2021	Naive Bayes SVM CNN BiLSTM AraBERT	SemEval ASTD	**41,196 tweets**	Tweets covering trending topics, product reviews, and controversial subjects	Pos, Neg, and Neu	F1-score	AraBERT: SemEval: 69% ASTD: 76%
[16]	2020	SVC, MNB and DTC	The paper itself	365,498 tweets	Tweets covering Covid-19	Safety, Worry, and Irony	F1-score	MNB: 85%
[20]	2019	CNN	SemEval for the EI-oc task	5600 tweets	Generic tweets	Joy, Anger, Sadness, and Fear	Recall	CNN: 99.82%
[17]	2019	SVM	SemEval 2017 AraSenTi	5971 11,112 2479 tweets	Generic tweets	Pos and Neg	F-Score	SemEval 2017: 80.38% AraSenTi: 74.84% ASTD: 79.77% ASTD: 79.77%

Furthermore, there are several related works on English sentiment analysis. Roy and Ojha in the study [27] trained and compared the performance of BERT, attention-based BiLSTM, and CNNs based on a SemEval-2016 dataset. The best model accuracy compared to the rest two models was the BERT model. Phan et al. [28] a feature ensemble model was proposed for sentiment analysis of tweets containing fuzzy sentiment. The model was tested on two datasets (DB1 and DB2). The proposed model using CNN achieved an accuracy of 81% and an F1-score of 0.81, while on the DB2 dataset, it achieved an accuracy of 73% and an F1-score of 0.76. The difference in results may be related to the nature of the datasets.

3 Methodology

This section outlines the comprehensive methodology of the study, organized into six key subsections. First, the dataset used is introduced, followed by a description of the cleaning and preprocessing techniques applied. Next, the study's modeling approach is explained, detailing the structure and algorithms. The development environment is then discussed, including the tools and platforms utilized. The evaluation metrics used to assess model performance are presented, and finally, the experimental design is described, outlining the procedures for conducting the experiments.

3.1 Dataset

In this study, we employed the MARSA dataset, which is "the largest sentiment annotated corpus for Dialectal Arabic in the Gulf region, consisting of 61,353 manually labeled tweets that contain a total of 840 K tokens. The annotators collected from trending hashtags in four domains: political, social, sports, and technology to create a multi-domain corpus" [11]. Leveraging the MARSA dataset, which comprises 61,353 tweets, each represented by a manually labeled feature, helps us to attain insightful and comprehensive sentiment analyses in this study, thus significantly enhancing the ability to understand and interpret the nuances and complexities of emotions expressed in colloquial Arabic across different domains. The data has seven labels, after considering the main three labels: Positive, Negative, and Neutral which will reduce the dataset to the 56,662 as shown in Table 2. Sample is shown in Table 3.

As illustrated in Table 2, the dataset consists of four domains related to politics, society, sports, and technology corresponding with its size. Each label presented as: Positive, representing the text contains good feelings or an opinion of satisfaction, enthusiasm, or compliment. Negative, meaning that the text conveys

feelings or opinions of dissatisfaction, such as aggression, anger, and blame. Neutral, meaning that it is neither positive nor negative, a text that does not carry feelings or a specific opinion (e.g., news or an advertisement).

Table 2: Dataset overview

Domain	Label	No. of tweets	Size	Total tweets
Political	Positive	1785	342 KB	9629
	Negative	3829	767 KB	
	Neutral	4015	709 KB	
Social	Positive	2896	552 KB	15,595
	Negative	5234	1.05 MB	
	Neutral	7465	1.28 MB	
Sport	Positive	12,202	1.94 MB	26,486
	Negative	8258	1.4 MB	
	Neutral	6026	828 KB	
Tech	Positive	331	34.8 KB	4952
	Negative	3407	556 KB	
	Neutral	1214	182 KB	

Table 3: Sample of MARSA dataset

Label	Tweets	English translation
Pos	الهلال_التعاون الف مبروووووك يازعيم هاردلك اخواننا نادي التعاون	Al Hilal_Al Taawon, congratulations Al-Za'eem. Hard luck, our brothers, Al Taawon Club
Pos	الهلال_التعاون الف مبروك للزعما وشكرا جماهير الزعيم الحاضره	Al-Hilal_Al-Taawoun, congratulations to the Al-Za'eem, and thanks to the Al-Za'eem' fans was present
Neu	جوله الطلاب في الاستديوهات	A student trip of the studios
Neu	كلمه_لشركات_الاتصالات وش تحب تقول لشركات الاتصالات موبايلي الاتصالات_السعوديه زين	Word_telecommunications_companies What would you like to say to the telecommunications companies Mobily Saudi_Telecom Zain
Neg	اذاكر اختباري وبعد كل درس افتح الهاشتاق ترا مليت تعليق_الدراسه	I am studying for my test after every lesson I open the hashtag I'm bored study_suspension
Neg	مقالي اليوم خمسه اسباب لماذا فرض رسوم_الاراضي_البيضا ضد مصلحه المواطن البسيط والوطن	My article today Five reasons why the imposition of White_land_fees is against the interests of the common citizen and the country
Sarcasm	موبايلي على رسايلها مابقي الا تشتغل خطابه	Mobily on her messages is almost working as a matchmaking agency
Sarcasm	انترنت_السعوديه_ضعيف الانترنت عندنا يعلمك الصبر	Saudi_Internet_weak here is teaching you the patience

3.2 Data Cleaning and Preprocessing

Preprocessing is the first step, to ensure consistency with [14] that utilized the same dataset, we adopt the same identical cleaning and preprocessing techniques. This approach allows for a direct performance comparison of the methodology using CAMeLBERT with the models presented in prior work. The cleaning

steps for refining the dataset and preparing it for sentiment analysis, as shown in Algorithm 1. First, duplicate tweets are identified and removed to prevent redundant data from influencing the results. Next, HTML entities embedded within tweets, such as '&' or '<', are decoded using the Beautiful Soup library. This conversion transforms them into readable characters. Unrelated content such as URLs, special characters, emojis, and usernames (which start with '@') are also cleaned out since they do not contribute meaningfully to the sentiment of the text. To maintain focus exclusively on Arabic content, digits, and non-Arabic words are removed. Hashtags, which often repeat sentiment-relevant terms, are also removed to reduce redundancy in sentiment prediction.

Algorithm 1: Data cleaning and preprocessing

procedure textCleaner (Dataset *df*)
    Remove duplicate texts
    Parse HTML and extract plain text
    Remove special characters, emojis, and usernames '@'mentions, URLs, 'www' links, Hashtags, Arabic and English numbers, underscores, and English letters using regex
    Tokenize text into words and filter out words with length ≤1
    Join filtered words into a single string
    return Cleaned dataset
end procedure

Additionally, stopwords common function words that do not contribute to sentiment are removed. In [14], two methods were tested for removing stopwords. Arabic stopword lists were manually collected using three different approaches to eliminate common function words that do not contribute to sentiment. However, all three approaches led to degraded model accuracy. To improve this, the Automatic Sentiment Refinement (ASR) algorithm was introduced, which is used to enhance the accuracy of sentiment analysis by removing irrelevant words that do not contribute to accurately identifying sentiments. The algorithm works by analyzing texts to identify words that appear with similar frequency across different sentiment categories, such as positive, negative, and neutral. The process begins by analyzing word frequency within labeled texts, measuring the occurrence of each word in each category. If a word is found to occur at nearly the same frequency across all categories, it is classified as neutral or non-influential for classification. These words are then removed from the texts as they do not contribute to distinguishing between sentiments. Based on these findings as shown in Algorithm 2, adopting the ASR method, as it is more efficient and less time-consuming.

Algorithm 2: Data cleaning and preprocessing

procedure PREPROCESSTEXT (Dataset *df*)
    Extract columns: x ← *df.text*; y ← *df.label*
    Initialize Count Vectorizer and fit it on *x*
    Compute term frequency matrices
    for label ∈ {0, 1, 2} do
        term_freq[label] ← Sum occurrences of words in class *label*
    end for

(Continued)

Algorithm 2 (continued)

Construct DataFrame with word frequencies
Compute sotal occurrences per word
    Sort words by total frequency and select top 86,024
    Initialize empty list *final_labels*
    for each word in dataset do
    if (negative − positive ≥ 5) and negative − neutral ≥ 5) then
        Label as "negative"
    else if (positive − negative ≥ 5) and (positive-neutral ≥ 5) then
        Label as "positive"
    else if (neutral − positive ≥ 5) and (neutral − negative ≥ 5) then
        Label as "neutral"
    else if word appears only in one category *then*
        Assign corresponding label
    else
        Label as "null"
    end if
  end for
  Collect "null" words as noise
  Store null words in a file for future use
  Load mull words and apply stopword removal
  for each text entry in dataset do
    Remove words in the null list
  end for
  return Cleaned dataset
end procedure

Furthermore, for the preprocessing, certain Arabic characters are normalized for consistency. For instance, various forms of the letter "Alef" (ا, ٱ, أ) are unified into a single form "ا". For this step, CAMeL tools [13] were used to enhance the results. Tokenization is then applied to split tweets into individual tokens, such as sub-words or words. For example, the phrase "التحليل اللغوي" is tokenized into ["," "ال", "تحليل", غوي], the English equivalent would be: "linguistic analysis" tokenized into ["lingu", "istic", "anal", "ysis"], which helps in handling Arabic's complex morphology. This process is performed using the WordPunctTokenizer from the NLTK library, which splits each tweet into tokens based on punctuation.

3.3 CAMeLBERT-DA SA Model

The CAMeLBERT model is built upon the original BERT architecture [13]. Which employs a transformer-based structure comprising multiple encoders [29]. While BERT is a general-purpose model, CAMeLBERT is specifically tailored for the Arabic language, incorporating modifications that enable it to better capture the unique linguistic features and complexities inherent to Arabic. Moreover, CAMeLBERT is a collection of eight pre-trained BERT models for NLP on Arabic texts, with different size and variant models of MSA, DA, CA and model combined with a mixture of these three [30]. Since this study utilizes the MARSA dataset, which focuses on Dialectal Arabic [11], we have selected CAMeLBERT-DA (bert-base-arabic-camelbert-da) model. This model has been pre-trained specifically on DA data. The dataset size is

approximately 54 GB, comprising 5.8 billion words [30]. To evaluate the accuracy of the pre-trained model, three approaches were employed as shown in Algorithm 1 of the study architecture.

The first approach evaluates the model's performance without prior training, while the second approach involves training the model and then conducting evaluations. The third approach includes preprocessing and fine-tuning, following the evaluation of the model. In the first approach, a subset of 7933 tweets is extracted from the original dataset prior to any preprocessing. This subset was equal in size to the testing data used in the third approach, and the model's performance was evaluated directly without any training. In the second approach, 80% of the dataset was allocated for training and 20% for testing on the CAMeLBERT model. The same methodology was applied to the AraBERTv2-twitter and MARBERT models for comparison to ensure a comprehensive benchmark. Lastly, in the third approach, the entire dataset underwent preprocessing, as detailed in the preprocessing section, incorporating some enhancements. After the preprocessing, the dataset is reduced to 40,755 tweets. The data was then split into training 80% and testing 20% sets. The model was trained on the training set and subsequently tested on the testing set. Finally, the model's performance was compared between the three approaches as illustrate in Fig. 1.

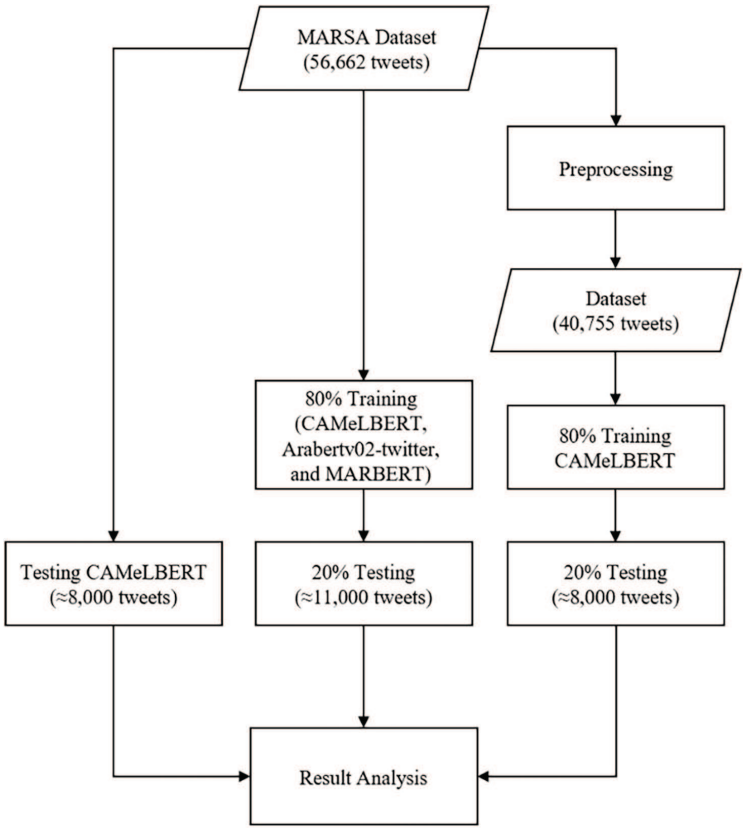

Figure 1: Proposed system architecture

The proposed methodology is examined around the use of pre-trained transformer models for Arabic sentiment analysis. Here, we provided a deeper dive into the framework and explain how the components contribute to the process.

The dataset D consists of N samples, where each sample is a pair of (x_i, y_i) then we can defined as:

$$D = \{(x_i, y_i), | \, i = 1, 2, 3, \ldots, N\} \tag{1}$$

where, the parameter x_i is the i-th tweet, and the y_i parameter is the sentimental lable of the tweet $y_i \in \{positive, negative, neutral\}$. Afterwards, the tweet x_i is preprocessed to standardize text format. This step is performed based on the normalization by applying transform variant forms of Arabic characters (e.g., أ, إ, آ → ا).

$$x_{it} = transform - variant\ (x_i) \tag{2}$$

In addition, we have alsp performed a stopword removal and noise cleaning step by removing URLs, mentions, and other irrelevant tokens.

$$x_{itr} = removal - stopwords - noise\ (x_{it}) \tag{3}$$

Each tweet x_{itr} is split into tokens using WordPiece tokenization. This ensures sub-word representations as:

$$x_{itr} \rightarrow \{t1, t2, t3, \ldots, tk\} \tag{4}$$

A transformer model like CAMeLBERT is used to generate contextual embeddings for each token:

$$h_t = CAMeLBERT\ (t_t) \tag{5}$$

where, the parameter h_t is the embedding vector for token in the above equation. These embeddings are aggregated to form a sentence-level representation by using the following equation as:

$$h_x = Aggregat\ (h_{t1}, h_{t2}, h_{t3}, \ldots, h_{tk}) \tag{6}$$

The aggregated representation h_x is passed through a fully connected layer with weights W and bias b, followed by a softmax activation:

$$\check{y}_i = argmax\ \sigma\ (W \times h_x + b) \tag{7}$$

where, the $\check{y}_i$ is the predicted sentiment and sigma (σ) is the softmax function. To train the model, the Cross-Entropy Loss L is used. This measures the difference between the true label y_i, and the predicted probability $\check{y}_i$ belong to class $\{positive, negative, neutral\}$. This loss function is calculated as:

$$L = -\frac{1}{N} \sum_{i=1}^{n} \sum_{c}^{C} Y_i \log\ (\check{y}_i) \tag{8}$$

The model parameters W and b are optimized using the AdamW optimizer. The objective is to minimize the loss function as defined in the above equation.

3.4 Development Environment

The development environment for this study used Google Colab to provide an accessible and sharable cloud-based platform while leveraging powerful GPU resources, and 12.7 GB of RAM. By utilizing Python3

programming language, exploiting the Hugging Face Transformers library for implementing the CAMeL-BERT (CAMeL-Lab/bert-base-arabic-camelbert-da) model along with the other models. Furthermore, data preprocessing, model training, and evaluation metrics are configured with essential libraries such as pandas, transformers, and Scikit-learn. The complete code for this study is available on GitHub for reference [31].

3.5 Evaluation Metrics

This study employed the evaluation metrics to assess the CAMeLBERT model's performance in the sentiment analysis task with the dataset. These metrics calculated are Accuracy (Eq. (1)), Precision (Eq. (2)), Recall (Eq. (3)), and F1-score (Eq. (4)); the metrics are expressed using well-defined mathematical equations [32].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

where, *TP*, *TN*, *FP*, and *FN* refer to True Positive, True Negative, False Positive, and False Negative, respectively.

3.6 Experimental Design

In this section, the study adjusted several key hyperparameters to optimize the performance of the CAMeLBERT model for sentiment classification along with the AraBERTv2-twitter and MARBERT models. Utilized the TrainingArguments class from the Transformers library to fine-tune these hyperparameters. The model was trained using Cross-Entropy Loss (Eq. (5)) [33]. Which is appropriate for the classification task. It is used to find the optimal weights during training, by computing the difference between actual outcomes and predicted outcomes. A lower cross-entropy loss indicates better model performance and accuracy.

With the Regularization technique the optimizer used was AdamW; the Weight decay was set at 0.01 to prevent overfitting by penalizing large weights [34]. The learning rate was set to 3e−5 and the batch size for training and evaluation was set to 16, balancing efficient memory usage and maintaining effective learning rates. These key hyperparameters, combined with the optimizer and loss function, were carefully tuned after multiple trials to achieve the best accuracy and F1-scores on the validation set, which were then validated on the test data. Table 4 presents the experimental design of CAMeLBERT, AraBERTv2-twitter and MARBERT models.

Table 4: Experimental design

Hyperparameter	Value
Weight decay	0.01
Learning rate	3e−5
Training and evaluation batch size	16
Number of epochs	1

In the methodology section, the process of cleaning and preprocessing the dataset is described first. Then, the model used in this study, development environment, evaluation metrics, and experimental design are described. Following the methodology section, the results and discussion section will present the performance of the CAMeLBERT model, along with interpretations of the findings. It will also address the study benchmark results.

4 Result and Discussion

In this section, we present the experimental with CAMeLBERT with the three approaches and benchmark.

4.1 CAMeLBERT

After evaluating the performance of the pre-trained CAMeLBERT model without any additional training or preprocessing, the model was tested on 7933 tweets extracted from a large dataset of 56,662 tweets. This sample was randomly selected, and it captured a similar distribution of sentiment labels (positive, negative, neutral). Testing on this sample without any further fine-tuning or preprocessing, the model achieved an accuracy of 62%, indicating moderate alignment with the sentiments of the dataset. This result demonstrates that while the pre-trained CAMeLBERT model captures some relevant features, further fine-tuning is required to improve accuracy due to enhance the model evaluation. Additionally, by fine-tuning the model without applying any data preprocessing using 80% of the entire dataset with 45,330 tweets and tested on a sample of 11,332 tweets, the model achieved an impressive accuracy of 92%, highlighting the effectiveness of training with a domain specific dataset. The main reason why accuracy improves with raw data is that for CAMeLBERT model was pre-trained on large, noisy datasets, enabling it to handle informal, unstructured text, such as slang and misspellings. Preprocessing steps like tokenization, lemmatization, and stopword removal can alter the text's structure, making it harder for the models to recognize patterns they were trained on. Additionally, informal texts like tweets, hashtags and emojis carry important sentimental information, and removing them during preprocessing can remove valuable context, lowering model performance. The training and testing on raw texts were initially part of an exploratory experiment. When the promising results (accuracy of 92%) were observed, it prompted the inclusion of these findings in the study to highlight the value of raw data when using pre-trained models.

Subsequently, CAMeLBERT model was fine-tuned after preprocessing which consists of 40,755 tweets, using 80% for training, while the remaining 20% of the dataset with total 8151 tweets was used for testing. After fine-tuning on this split, the model achieved an accuracy of 89%. Several experiments were conducted to adjust hyperparameters, such as the number of epochs and batch size, but these changes did not significantly affect the model's accuracy. However, tuning the learning rate had a noticeable impact: with a learning rate of 10e−5, the accuracy was 88%, while lowering it to 3e−5 resulted in an improved accuracy of 89%. This highlights the sensitivity of the model's performance to learning rate adjustments.

Three different confusion matrices illustrate the model's predictive performance for testing the model, fine-tuning with and without data preprocessing. The first confusion matrix in Fig. 2 represents the results of the pre-trained CAMeLBERT model tested on the initial sample of 7933 tweets, achieving a moderate accuracy of 62%. This matrix shows notable misclassifications, especially between the negative and neutral classes, indicating the model's initial limitations in distinguishing certain sentiment categories.

As illustrate in Fig. 3, the confusion matrix after model fine tuning without data preprocessing demonstrates a clear diagonal line, indicating that most predictions align with the true labels, highlights a marked reduction in misclassifications, demonstrating the effectiveness of fine tuning in enhancing the model's ability to correctly classify sentiment labels.

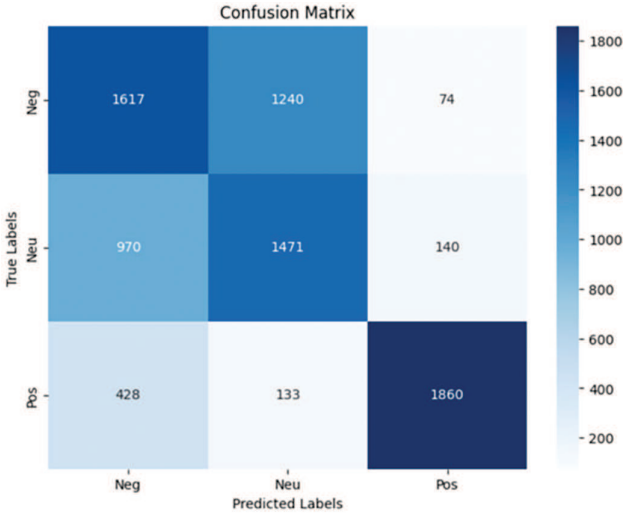

Figure 2: Confusion matrix for model testing

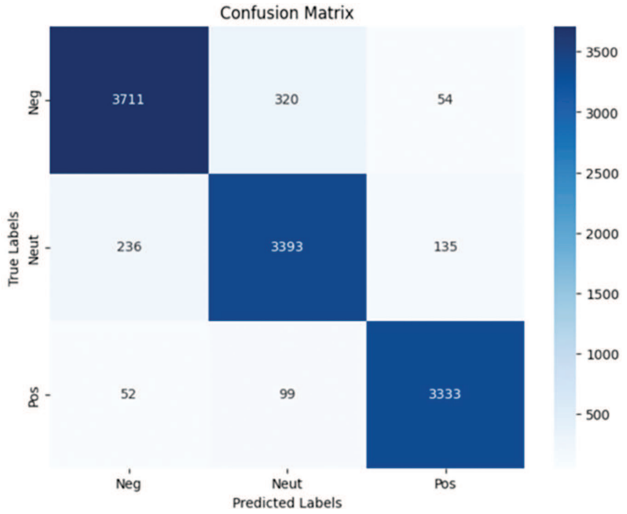

Figure 3: Confusion matrix after fine-tuning without data preprocessing

In contrast, the performance after fine-tuning with data preprocessing shows additional insights. The confusion matrix in Fig. 4 demonstrates the model's improved classification ability compared with Fig. 2, and highlights areas for potential refinement, particularly the minor misclassifications observed between the negative and neutral classes.

As shown in Fig. 3, the confusion matrix for fine-tuning without preprocessing reveals some misclassification between Negative and Neutral classes. This indicates that while the model captures positive sentiment effectively, it struggles with the subtle differences between negative and neutral tones. For example:

- Post: 'دا مطر سعبول جاهم اذا القصيم _في_الدراسه_تعليق...' (Suspension of studies in Al-Qassim: they just got a few drops of rain. . .), True Label: Negative, Predicted Label: Neutral
- Post: 'بجده الاهليه العالميه مدارس ايقاف ملكيه_اوامر...' (Royal orders to suspend the private and international schools in Jeddah. . .), True Label: Neutral, Predicted Label: Negative

Training CAMeLBERT took 1292.28 s per epoch, while evaluation and prediction took 130.97 s.

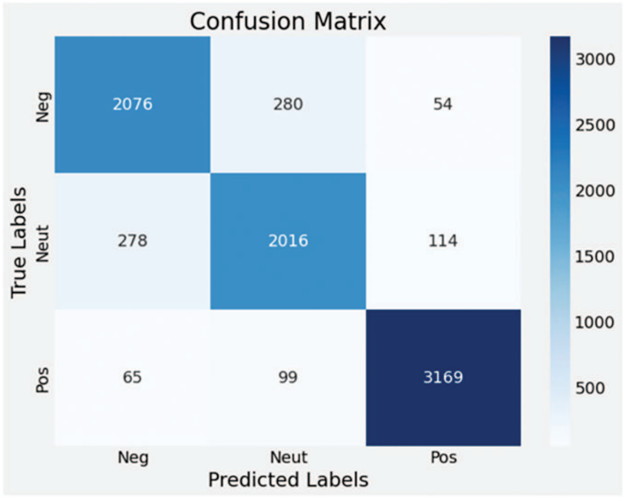

Figure 4: Confusion matrix after fine-tuning with data preprocessing

4.2 MARBERT

The MARBERT model achieved a test accuracy of 92.54%, The training took 1360.57 s per epoch, and evaluation and prediction took 135.60 s. As shown in the confusion matrix Fig. 5, most predictions were accurate, but some misclassifications occurred, particularly between Negative and Neutral classes. For example:

- Post: "وما خلص الترم والله الدراسه يعلقون ما المفروض..." (They shouldn't suspend classes; the term is almost over. . .), True Label: Negative, Predicted Label: Neutral
- Post: "ابن وسلامه المواطن عزيزي لسلامتك الدراسه تعليق..." (Study suspension for your safety and your child's safety. . .), True Label: Neutral, Predicted Label: Negative

These examples indicate challenges in distinguishing subtle sentiment differences, despite the overall strong performance.

4.3 AraBERTv02-Twitter

The AraBERTv02-Twitter model achieved a test accuracy of 92.92% with a weighted F1-score of 0.93 across all classes. The training process took 1326.00 s per epoch, and evaluation and prediction took 127.95 s. As shown in the confusion matrix Fig. 6, the model performed well overall, but some misclassifications were observed, particularly between Neutral and Negative classes. For example:

- Post: "ال من و يعتبر قصر يمتلك عامل و يرزقنا يرزقه الله..." (May God bless him and us; a worker owns a palace and is considered one of the. . .), True Label: Neutral, Predicted Label: Positive
- Post: "12 الاتصالات تغطيه خارج حقل ومحافظه ساعه _ الس..." (12 h, and Haql Governorate is out of telecommunications coverage. . .), True Label: Negative, Predicted Label: Neutral

These examples illustrate the model's difficulty in distinguishing between subtle Neutral and Positive sentiments, as well as handling contextually complex Negative expressions. Despite these challenges, the model demonstrated robust performance across all sentiment classes.

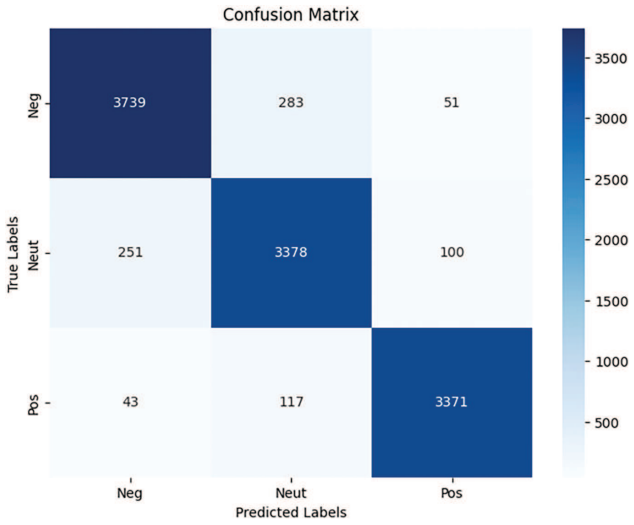

Figure 5: MARBERT confusion matrix

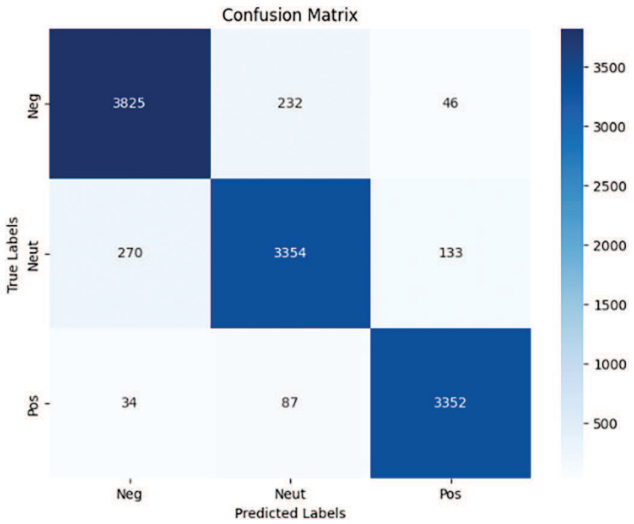

Figure 6: AraBERTv02-Twitter Confusion matrix

4.4 Benchmark

The results were benchmarked against several models presented in the study [14] by Al Wazrah and Alhumoud as illustrate in Fig. 7. The models compared include LSTM, SVM, AraBERT, SGRU, SBi-GRU, and an ensemble method combining the top-performing models. The LSTM model achieved an accuracy of 82.05% using SG word embedding, while the SVM showed performance 77.92%, AraBERT reached 85.41% accuracy, the SGRU model, with 6 layers, achieved 82.08%, demonstrating the effectiveness of GRUs for Arabic sentiment analysis, while the 5-layer SBi-GRU scored 81.59%, with advantages in recall and F1-scores. The ensemble method, combining AraBERT, SGRU, and SBi-GRU, achieved the highest accuracy of 90.21%. In comparison, the fine-tuned CAMeLBERT model achieved 92%, placing it as highly competitive with the ensemble method and outperforming individual models like SGRU, SBi-GRU, and LSTM, approximating AraBERTv02-twitter, and MARBERT accuracy. These results suggest that fine-tuned transformer-based models, such as CAMeLBERT, are highly effective for sentiment analysis of Arabic.

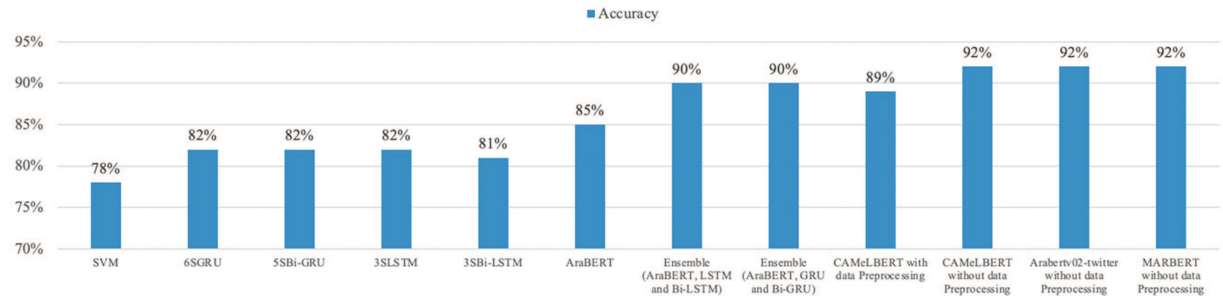

Figure 7: Accuracy benchmark result

4.5 Limitation

This study faced some challenges during model implementation. With Hyperparameter Tuning, the performance of the CAMeLBERT model can be sensitive to hyperparameters. Finding an optimal set of hyperparameters can require multiple training iterations and be time-consuming. Also, the training time depends on the dataset's size and the model's complexity; training CAMeLBERT can take a significant amount of time, which may delay research timelines. These are the limitations of the study that were faced during the implementation.

The section conducted the study results along with comparison, enumerating the limitations faced in the implementation. The last section concludes the study and potential future work.

5 Conclusion

This study comprehensively analyzes sentiment in Arabic Twitter/X's tweets using the pre-trained CAMeLBERT model. The model was evaluated on a large dataset, demonstrating its ability to adapt to the complex linguistic structure, such as the difficulty of morphology and the word's meaning for every diversity of dialects of the Arabic language. The fine-tuned CAMeLBERT model significant improvement in performance on Arabic sentiment analysis. When trained and tested directly on the raw dataset, it achieved an accuracy of 92%, compared to 89% when trained and tested on the preprocessed dataset. In its pre-trained state, without training, the model achieved only 62% accuracy. This highlights the benefit of fine-tuning on raw data for better performance, as preprocessing steps slightly reduced accuracy, the model is pre-trained on noisy, unstructured text, enabling it to handle informal language, while preprocessing maybe disrupts syntactic and semantic patterns and removes key sentiment signals.

Additionally, other models, including AraBERTv02-twitter and MARBERT, were also trained without preprocessing, achieving an accuracy of 92%. A comprehensive benchmarking was performed using CAMeLBERT, AraBERTv02-twitter, and MARBERT, comparing their results with findings from other studies to assess their performance and reliability. This shows that CAMeLBERT competes closely with state-of-the-art models, such as SGRU and ensemble approaches. It highlights the importance of fine-tuning pre-trained models for domain-specific tasks, particularly in complex languages like Arabic.

In future work, further improvements could involve exploring hybrid or ensemble approaches to enhance performance and reliability, such as combining CAMeLBERT with other models for more accurate results. Additionally, explore a wider range of hyperparameters and preprocessing methods could be tested to validate conclusions.

Acknowledgement: The authors express their gratitude to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) for funding and supporting this work through the Graduate Students Research Support Program.

Funding Statement: This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-DDRSP2504).

Author Contributions: Conceptualization: Lama Aldhafeeri, Fay Aljomah, Maha Alfadel, Sultanh Alshahrani; Methodology: Lama Aldhafeeri, Fay Aljomah, Maha Alfadel; Implementation: Lama Aldhafeeri, Fay Aljomah; Validation: Lama Aldhafeeri, Fay Aljomah, Maha Alfadel, Sultanh Alshahrani; Writing—original draft preparation: Lama Aldhafeeri, Fay Aljomah, Maha Alfadel, Sultanh Alshahrani; Writing—review and editing: Fay Aljomah, Sultanh Alshahrani; Visualization: Lama Aldhafeeri, Fay Aljomah, Maha Alfadel, Sultanh Alshahrani; Supervision: Qaisar Abbas, Sarah Alhumoud; Project administration: Lama Aldhafeeri. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available in Section 3, Paragraph A. For further reference, see [14].

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Abdul Sattar K, Obeidat Q, Akour M. Towards harnessing based learning algorithms for tweets sentiment analysis. In: Proceedings of the 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT); 2020 Dec 20–21; Sakheer, Bahrain. doi:10.1109/3ICT51146.2020.9311990.
2. Duwairi RM, Marji R, Sha'ban N, Rushaidat S. Sentiment analysis in Arabic tweets. In: Proceedings of the 2014 5th International Conference on Information and Communication Systems (ICICS); 2014 Apr 1–3; Irbid, Jordan. doi:10.1109/IACS.2014.6841964.
3. Chifu AG, Fournier S. Sentiment difficulty in aspect-based sentiment analysis. Math. 2023;11(22):4647. doi:10.3390/math11224647.
4. Heikal M, Torki M, El-Makky N. Sentiment analysis of Arabic tweets using deep learning. Procedia Comput Sci. 2018;142:114–22. doi:10.1016/j.procs.2018.10.466.
5. Alshaikh KA, Almatrafi OA, Abushark YB. BERT-based model for aspect-based sentiment analysis for analyzing Arabic open-ended survey responses: a case study. IEEE Access. 2024;12(3):2288–302. doi:10.1109/ACCESS.2023.3348342.
6. Kim H, Jeong YS. Sentiment classification using convolutional neural networks. Appl Sci. 2019;9(11):2347. doi:10.3390/app9112347.
7. Yin Q. Three-class text sentiment analysis based on LSTM. arXiv:2412.17347v1. 2024.
8. Al Sallab A, Hajj H, Badaro G, Baly R, El Hajj W, Bashir Shaban K. Deep learning models for sentiment analysis in Arabic. In: Proceedings of the Second Workshop on Arabic Natural Language Processing; 2015 Jul 30; Beijing, China. doi:10.18653/v1/W15-3202.
9. Caroppo A, Leone A, Siciliano P. Comparison between deep learning models and traditional machine learning approaches for facial expression recognition in ageing adults. J Comput Sci Technol. 2020;35(5):1127–46. doi:10.1007/s11390-020-9665-4.
10. Oueslati O, Cambria E, HajHmida MB, Ounelli H. A review of sentiment analysis research in Arabic language. Future Gener Comput Syst. 2020;112(4):408–30. doi:10.1016/j.future.2020.05.034.
11. Alowisheq A, Al-Twairesh N, Altuwaijri M, Almoammar A, Alsuwailem A, Albuhairi T, et al. MARSA: multi-domain Arabic resources for sentiment analysis. IEEE Access. 2021;9:142718–28. doi:10.1109/ACCESS.2021.3120746.

12. Alayba AM, Palade V, England M, Iqbal R. Improving sentiment analysis in Arabic using word representation. In: Proceedings of the 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR); 2018 Mar 12–14; London, UK. doi:10.1109/ASAR.2018.8480191.
13. Obeid O, Zalmout N, Khalifa S, Taji D, Oudah M, Alhafni B, et al. CAMeL tools: an open source python toolkit for Arabic natural language processing. In: Calzolari N, Béchet F, Blache P, Choukri K, Cieri C, Declerck T, et al., editors. Proceedings of the Twelfth Language Resources and Evaluation Conference; 2020 May 11–16; Marseille, France.
14. Wazrah AA, Alhumoud S. Sentiment analysis using stacked gated recurrent unit for Arabic tweets. IEEE Access. 2021;9:137176–87. doi:10.1109/ACCESS.2021.3114313.
15. Alghamdi HM. Unveiling sentiments: a comprehensive analysis of Arabic Hajj-related tweets from 2017–2022 utilizing advanced AI models. Big Data Cogn Comput. 2024;8(1):5. doi:10.3390/bdcc8010005.
16. Bahja M, Hammad R, Amin Kuhail M. Capturing public concerns about coronavirus using Arabic tweets: an NLP-driven approach. In: Proceedings of the 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC); 2020 Dec 7–10; Leicester, UK. doi:10.1109/UCC48980.2020.00049.
17. Al-Twairesh N, Al-Negheimish H. Surface and deep features ensemble for sentiment analysis of Arabic tweets. IEEE Access. 2019;7:84122–31. doi:10.1109/ACCESS.2019.2924314.
18. Louati A, Louati H, Kariri E, Alaskar F, Alotaibi A. Sentiment analysis of Arabic course reviews of a Saudi university using support vector machine. Appl Sci. 2023;13(23):12539. doi:10.3390/app132312539.
19. Alqahtani Y, Al-Twairesh N, Alsanad A. Improving sentiment domain adaptation for Arabic using an unsupervised self-labeling framework. Inf Process Manag. 2023;60(3):103338. doi:10.1016/j.ipm.2023.103338.
20. Baali M, Ghneim N. Emotion analysis of Arabic tweets using deep learning approach. J Big Data. 2019;6(1):89. doi:10.1186/s40537-019-0252-x.
21. Khalil EAH, Houby EMFE, Mohamed HK. Deep learning for emotion analysis in Arabic tweets. J Big Data. 2021;8(1):136. doi:10.1186/s40537-021-00523-w.
22. Abu Farha I, Magdy W. A comparative study of effective approaches for Arabic sentiment analysis. Inf Process Manag. 2021;58(2):102438. doi:10.1016/j.ipm.2020.102438.
23. Baniata LH, Kang S. Switch-transformer sentiment analysis model for Arabic dialects that utilizes a mixture of experts mechanism. Math. 2024;12(2):242. doi:10.3390/math12020242.
24. Al-Khalifa S, Alhumaidhi F, Alotaibi H, Al-Khalifa HS. ChatGPT across Arabic twitter: a study of topics, sentiments, and sarcasm. Data. 2023;8(11):171. doi:10.3390/data8110171.
25. Shah SMAH, Shah SFH, Ullah A, Rizwan A, Atteia G, Alabdulhafith M. Arabic sentiment analysis and sarcasm detection using probabilistic projections-based variational switch transformer. IEEE Access. 2023;11:67865–81. doi:10.1109/ACCESS.2023.3289715.
26. Zahidi Y, Al-Amrani Y, El Younoussi Y. Improving Arabic sentiment analysis using LSTM based on word embedding models. Vietnam J Comput Sci World Sci. 2023;10(3):391–407. doi:10.1142/S2196888823500069.
27. Roy A, Ojha M. Twitter sentiment analysis using deep learning models. In: Proceedings of the 2020 IEEE 17th India Council International Conference (INDICON); 2020 Dec 10–13; New Delhi, India. doi:10.1109/INDICON49873.2020.9342279.
28. Phan HT, Tran VC, Nguyen NT, Hwang D. Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model. IEEE Access. 2020;8:14630–41. doi:10.1109/ACCESS.2019.2963702.
29. Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805v2. 2019.
30. Inoue G, Alhafni B, Baimukan N, Bouamor H, Habash N. The interplay of variant, size, and task type in Arabic pre-trained language models. arXiv:2103.06678v2. 2021.
31. Aldhafeeri L. CAMelBERT [Internet]. [cited 2024 Oct 19]. Available from: https://github.com/Lama-Aldhafeeri/CAMelBERT.

32. Almaliki M, Almars AM, Gad I, Atlam ES. ABMM: Arabic BERT-mini model for hate-speech detection on social media. Electronics. 2023;12(4):1048. doi:10.3390/electronics12041048.
33. Bishop CM. Pattern recognition and machine learning. Berlin/Heidelberg, Germany: Springer; 2006. 778 p.
34. Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv:1711.05101v3. 2019.