Utilizing Deep Learning in Arabic Text Classification Sentiment Analysis of Twitter

Nehad M. Ibrahim[1], Wael M.S. Yafooz[2], Abdel-Hamid M. Emara[3], Ahmed Abdel-Wahab[4]
Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia[1]
Department of Computer Science-College of Computer Science and Engineering, Taibah University, Medina 42353, Saudi Arabia[2, 3]
Department of Computers and Systems Engineering-Faculty of Engineering, Al-Azhar University, Cairo 11884, Egypt[3, 4]
Arab Open University, Riyadh, Saudi Arabia[4]

***Abstract*—The number of social media users has increased. These users share and reshare their ideas in posts and this information can be mined and used by decision-makers in different domains, who analyse and study user opinions on social media networks to improve the quality of products or study specific phenomena. During the COVID-19 pandemic, social media was used to make decisions to limit the spread of the disease using sentiment analysis. Substantial research on this topic has been done; however, there are limited Arabic textual resources on social media. This has resulted in fewer quality sentiment analyses on Arabic texts. This study proposes a model for Arabic sentiment analysis using a Twitter dataset and deep learning models with Arabic word embedding. It uses the supervised deep learning algorithms on the proposed dataset. The dataset contains 51,000 tweets, of which 8,820 are classified as positive, 37,360 neutral, and 8,820 as negative. After cleaning it will contain 31,413. The experiment has been carried out by applying the deep learning models, Convolutional Neural Network and Long Short-Term Memory while comparing the results of different machine learning techniques such as Naive Bayes and Support Vector Machine. The accuracy of the AraBERT model is 0.92% when applying the test on 3,505 tweets.**

Keywords*—Arabic sentiment analysis; machine learning; convolutional neural networks; word embedding; Arabic word2Vec; long short-term method; AraBERT

I. Introduction

Recently, sentiment analysis has been prioritized by researchers because it plays an important role in many domains. It is primarily used to study user feedback (user opinion) on a specific event, product or social phenomenon. Many studies have proposed models, approaches or novel databases to predicate and detect user opinions. These methods use machine learning classifiers, deep learning models and natural language techniques as pre-processing methods. Most of the sentiment analysis research focuses on languages other than Arabic. Recent Natural Language Processing research is now increasingly focused on using deep neural learning [1]. Some research initiatives are being launched in a competition funded by the King Abdullah University of Science and Technology (KAUST). They focus on the Arabic language and some individual research efforts.

Generally, in other languages, specifically English, the universal language has proven to be significant due to the vast amount of data contributed by users on social networks (Facebook, Twitter, etc.). In machine learning, a classification known as supervised learning is used in sentiment analysis. There are several methods used in sentiment analysis which can be categorized into binary classification, multi-classification, polarity, multilingual and aspect-based sentiment analysis. In binary classification, the classes can be represented only as positive and negative. In multi-class, there are more than two classes. Additionally, there are classifiers used in binary classification such as DT and TH, while KNN and LR are used in multi-classification. Polarity in sentiment analysis is based on a dictionary that assigns a score to each word. Multilingual sentiment analysis requires many pre-processing steps to be performed in option detection and aspect-based. It is focused on one aspect, concept or word.

To the best of our scholarly knowledge, less attention has been given to Arabic sentiment analysis and there are fewer public Arabic datasets [2]. Therefore, this paper proposes a model for Arabic sentiment analysis based on the proposed dataset. This work uses supervised deep learning algorithms. The original dataset before the cleaning process contains 51,000 tweets classified as 8,820 positive, 37,360 neutral and 8,820 negative. After cleaning, it contains 31,413 tweets classified as 4,855 positive, 21,842 neutral and 4,716 negative. This work introduces and applies deep learning methods on Arabic sentiment analysis text multi-classes with parameter optimization, and improves the process in the text pre-processing area. We apply the deep learning methods Convolution Neural Network (CNN) and Long Short-Term Memory (LSTM) to compare the results of different supervised machine learning techniques such as Naive Bayes (NB) and Support Vector Machine (SVM). The accuracy of the best CNN model is 95.8% and the accuracy of LSTM is 96.6%, which are better than the SVM and NB results, which are 82.5% and 69.4%, respectively. We used BERT pre-trained specifically in Arabic to achieve the same success that BERT achieved in English [3]. Based on a review of the literature and the high accuracy achieved in the deep learning models, the main contributions of this paper can be summarized as follows.

- Develop a model for Arabic sentiment analysis using machine learning and deep learning models.
- Explore the most recent approaches to Arabic sentiment analysis.
- Propose a novel dataset called ASAD that is publicly available.
- Perform a comparative analysis of the results of MLC and DLM.

The remainder of this paper is organized as follows: Section II presents an overview of related studies on sentiment analysis. Our research methods and materials are explained in Section III. Section IV presents the results. The conclusion is in Section V.

II. Related Studies

Scholars have not given enough attention to the Arabic Sentiment Analysis Dataset (ASAD). It [4] provides a comprehensive overview of a new Twitter-based benchmark dataset for Arabic sentiment analysis. ASAD is a massive, high-quality annotated dataset (including 95,000 tweets) with three-class sentiment labels, compared to other publicly released Arabic datasets (positive, negative and neutral) in [5], researched Twitter's sentiment analysis. Three machine learning algorithms are used, Logistic Regression, Aid Vector Classification, and NB, with two sets of characteristics. The word frequency approach, word embeddings, and machine learning classifiers can correctly identify rumour-related tweets with 84% accuracy, which classifies tweets into four categories: academic, media, government and health [6].

In [7], two new Arabic text categorization datasets are introduced. The first consists of Twitter, Facebook and YouTube posts from well-known Arabic news channels, and the second consists of tweets from popular Arabic accounts. In modern standard Arabic, the papers in the former are almost entirely written (MSA), while the tweets in the latter contain both MSA and dialectal Arabic.

In [8], Word2Vec models were collected from 10 newspapers in different Arabic countries from a broad Arabic corpus. The reports increased the accuracy of sentiment classification after applying various machine learning algorithms and convolution neural networks with different text feature choices (91%–95%).

In [9], an in-house-built dataset shows tweets and comments where three classifiers were applied, including NB, SVM and K-Nearest Neighbour, in particular. The findings show that SVM provides the highest accuracy, while KNN (K=10) provides the highest recall.

In [10], four classifiers were trained to incorporate a dataset consisting of 4,712 tweets to conduct a comparative study on the output of the classifiers, namely NB, SVM, Multinomial Logistic Regression and K-Nearest Neighbour. When running against the tweet's dataset, these algorithms revealed that SVM gives the highest F1 score (72.0), while KNN (K=2) achieved the best accuracy, equivalent to 92.0.0.

In [11] the processes of gathering Twitter data and filtering, pre-processing and annotating the Arabic text to create a large dataset of sentiment analysis in Arabic are summarized. In addition to deep learning and CNN, machine learning algorithms (NB, SVM, and Logistic Regression) were used on the health dataset in the sentiment analysis experiments. The keywords are Machine Learning, Deep Neural Networks, Arabic Language and Emotion Analysis.

Several versions of RNN and CNN classifiers using GloVe-based word embedding were introduced. All classifiers performed well, while the classifiers between 90% and 91% had the highest accuracy. Experimental findings indicate that BRAD 2.0 is rich and stable [12]. To encourage more study in the field of Arabic computational linguistics, the benchmark dataset was made available as the key contribution.

In [13] a new mix model from CNN and LSTM is proposed using vector representations of sentences and the SoftMax regression classifier to identify the sentiment tendencies in the text. In [14] a method for evolving the CNN and creating an Arabic sentiment classification system is proposed using the differential evolution (DE) algorithm. In [15] a novel architecture for Arabic word classification and understanding is proposed. This is based on CNNs and recurrent neural networks that address the difficulty of handling unstructured social media texts in low data availability. Therefore [16] attempts to identify expressions related to feelings, such as happiness, rage, anxiety and sadness. In addition, it presents the emotion classification in Arabic tweets by using the CNNs and compares them with the machine learning methods SVM, NB and Multi-Layer Perceptron (MLP).

In [17] an Arabic sentiment analysis corpus culled from Twitter is presented, consisting of 36,000 tweets categorized as positive or negative, plus 8,000 tweets manually annotated and used to assess the corpus intrinsically by comparing it to human classification and pre-trained sentiment analysis models, with an accuracy of 86%. In [18] a survey is proposed that focuses on 90 recent research papers (74% were published after 2015). In [19] supervised and unsupervised transformation methods, such as principal component analysis (PCA) and latent Dirichlet allocation (LDA), are presented. They are tested on five Arabic opinion text datasets of various domains and sizes (1.6–94,000 reviews). In the two-class classification problem, accuracy values range from 95.5%–99.8%, and for the three-class classification problem, accuracy values ranged from 92%–97.3%.

In [20] a new study is presented to develop a new model to predict an individual's awareness of the precautionary procedures. Tweets related to COVID-19 were collected from the five main regions in Saudi Arabia, and the accuracy level achieved was 85%. A systematic comparative overview of the most appropriate methods for analysing Arabic sentiment is presented in [21]. It carries out a thorough comparison of various machine learning methods for Arabic sentiment analysis, such as NB, SVM, CNN, LSTM and several recently developed language models. The model achieves F-scores of 0.69, 0.76 and 0.92. A method for extracting knowledge from Arabic text on social media in four stages – data collection, cleaning, enrichment and availability – is shown in [22]. It

offers an integrated solution for the challenges of pre-processing Arabic text on social media. This was undertaken to investigate the performance metrics as given in [23, 24, 25, 26, 27, 28, 29] and validates the proposed model for small-and large-scale datasets. Disambiguation using the deep learning techniques with the Arabic corpus is presented in [30]. An Arabic model for text clustering using word embedding and Arabic word net is presented in [31].

III. Methodology

This section describes the research methodology that was used to conduct this research. It consisted of six main interrelated phases: text retrieval; text pre-processing; tokenization and feature extraction; application of the deep learning model; model performance evaluation as shown in Fig. 1; and use of the transfer learning by applying the AraBERT Model.

The analysis concentrated on three cases of tweets, positive, negative and neutral. To perform the sentiment analysis experiment, the large number of collected tweets was necessary. A lot of noisy data was used in the total number of tweets (51,000 tweets). The model architecture is shown in Fig. 1. The first phase in this work is text retrieval; the second is text cleaning; the third is tokenization; the fourth is embedding the text using the Word2Vec corpus in [32]; the fifth is to apply deep learning models; and the last phase is to evaluate the results.

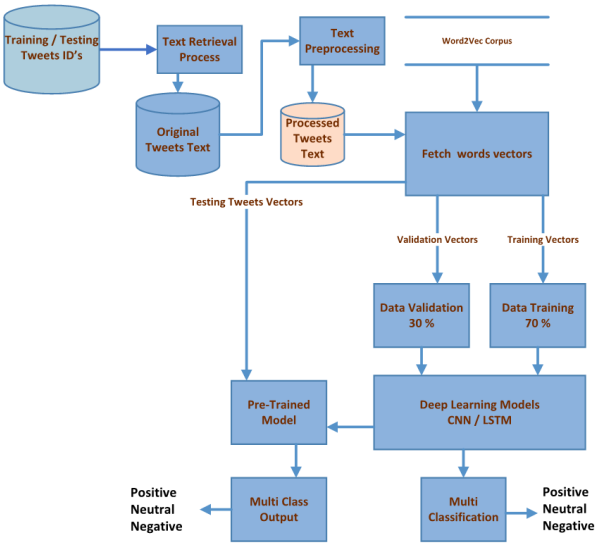

Fig. 1. Model architecture

A. Tweet Text Retrieval

This section describes the first phase, text retrieval. Data retrieval is performed using Tweepy API, and because the text characters are in Arabic, we implemented an Arabic text retrieval module using the Tweepy Twitter API library in Python.

B. Text Pre-Processing

Improving the accuracy of the text classification required enhancing the text features [22]. We added some feature selection improvements, such as noise removal. Removing the noisy characters from the text enhanced the word representation. The following steps were run on the text to remove the noisy data.

- Remove the advertisement tweets.
- Remove the retweeted tweets, which started with the segment 'RT'.
- Remove the duplicate tweets, which were retrieved more than once.

1) Normalization: Normalization is a pre-processing method of text data cleaning, to format a sequence of texts into a standard uniform [33]. It is difficult to analyse Arabic texts because of the nuances of the language, both in terms of infrastructure and conformation. Arabic has an abundance of diverse inflexions, dialects and spellings that change the meanings of the words. Using special labels, called configuration, rather than vowels, they differ according to the shape of the word. This method is necessary and useful in word processing to minimise uncommon terms and increase classification accuracy. The following steps were applied to the stored tweets in the dataset on two levels. The first level retained the emoji and the second level removed the emoji from the data collection phase.

- Remove digits, non-Arabic letters, single letters, punctuations, diacritics and special characters ($, %, &, #, .).
- Separate all words by spaces.
- Remove (ها, وا, ان, ون, ين , ات) from the end of the word.
- Remove (ال, تال, وكال, كال,وال, وتال, ولل, لل) from the beginning of the word except إله لله, الله, اللهم.
- Normalize some characters with a single character, such as (أ, إ , آ) with (ا), (ئ,ي) with (ى) and (ة) with (ه).
- Replace repeated characters (e.g.,يااااااااااااارب) with a single character (e.g., يارب).

2) Stop words removal: Stop words are words that do not affect the meaning of the sentence, and eliminating the stop words from the text helps to identify the most important words as the following [34].

- Relative pronouns (الأسماء الموصولة)
- Referral names/determiners (أسماء الإحالة / المحددات)
- Transformers (verbs, letters) ((الأفعال والحروف) المحولات)
- Verbal pronouns (الضمائر اللفظية)
- Adverbs (الاحوال)
- Interrogative pronouns (ضمير الاستفهام)
- Conditional pronouns (الضمير مشروط)
- Prepositions (حرف جر)
- Pronouns (الضمائر)

a) Pre-processing: We performed pre-processing operations on the data, such as removing the stop words, special characters, such as '@' and '#', URLs, non-Arabic characters and punctuation to create a clear analysis of the text. In addition, we used NLTK (https://www.nltk.org) word tokenization on the text data (refer Table I)

TABLE I. SAMPLES OF TWEETS BEFORE AND AFTER THE NOISE CLEANING PROCESS

#	Label	Text before cleaning	Text normalized with emoji retained	Text normalized with emoji removed	English meaning
1	Neutral	'عندي @elm مشكله لما تبي اجدد إقامة عامل حاولت أتواصل بس يفصل معي الخط من قبل مقيم'	عندي مشكله لما تبي اجدد إقامة عامل حاولت أتواصل بس يفصل معي الخط من قبل مقيم	عندي مشكله لما تبي اجدد إقامة عامل حاولت أتواصل بس يفصل معي الخط من قبل مقيم	I have a problem when I want to find the residency of a worker, I tried to contact me, but the line was separated by a resident
2	Positive	'@nas_alharbi8 والله حسب الأرقام سيكون مخيب للآمال ولكن الأهلي قدها برجاله في الملعب '	والله حسب الأرقام سيكون مخيب للآمال ولكن الأهلي قدها برجاله في الملعب	والله حسب الأرقام سيكون مخيب لآمال ولكن الأهلي قدها برجاله في الملعب	By God, according to the numbers, it will be disappointing, but Al-Ahly has led it with its men on the field
3	Neutral	'" الزعل بيغير ملامحك ، بيغير نظرة العين ، بيغير شكلك في الصور ، الزعل ممكن يطفيك تماما."'	الزعل بيغير ملامحك بيغير نظرة العين بيغير شكلك في الصور الزعل ممكن يطفيك تماما	الزعل بيغير ملامحك بيغير نظرة العين بيغير شكلك في الصور الزعل مكن يطفيك تماما	Upset changes your features, changes the look of the eyes, changes your appearance in the photos, upset you can completely eliminate you
4	Positive	'ثقي قلباً وقالباً معاكِ وفخورين فيكِ وين مارحتي وربي يوفقك ويسهلك طريقك☺♥ https://t.co/N8vVjcwD08'	ثقي قلباً وقالباً معاكِ وفخورين فيكِ وين مارحتي وربي يوفقك ويسهلك طريقك☺♥	ثقي قلبا وقالبا معاك وفخورين فيك وين مارحتي وربي يوفقك ويسهلك طريقك	Trust your heart and soul with you and be proud of you. Where have you gone, and my Lord will help you and make your way
5	Negative	'هذا المخلوق ينتمي إلى المجتمع الشيطاني ما في منه اي خير . https://t.co/IYoS64SKJZ'	هذا المخلوق ينتمي إلى المجتمع الشيطاني ما في منه اي خير	هذا المخلوق ينتمي إلى المجتمع الشيطاني ما في منه اي خير	This creature belongs to the satanic community, there is no good in it
6	Negative	'ادور لعيني النوم لو كانه سلف .. من بارحة الاولى و انا اودي و اجيب .'	ادور لعيني النوم لو كانه سلف من بارحة الاولى و انا اولادي و اجيب	ادور لعيني النوم لو كانه سلف من بارحة الاولى و انا اولادي و اجيب	I turn to my eyes to sleep if it was an ancestor from yesterday and I am my children and I answer

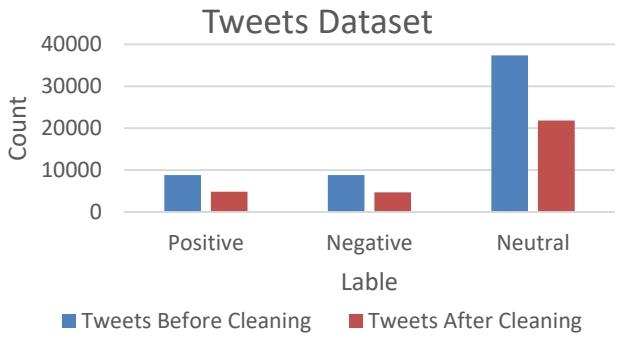

Fig. 2. Dataset statistics

Fig. 2 shows the dataset before and after cleaning. As the dataset published by [34] contains IDs and annotations only, the first step must be to retrieve the tweeter text using the authorized API object from Tweepy, using the API to retrieve all IDs from the file one by one.

3) Tokenization: Tokenization is the method used to break down text into individual tokens separated by white space. Tokenization removes all special characters, determines phrase boundaries, and processes abbreviations and numbers [35]. Due to the morphological complexity of the language, the number of tokens in Arabic can exceed four. Since Arabic words often contain many affixes and clitics, the tokenization process was preceded by a segmentation process to eliminate suffixes.

C. Word Vectors Lookup

Word embedding [36] is a language modelling and feature learning technique, where each word is mapped to a vector of real values in such a way that words have a similar representation with similar meanings. Using neural networks, value learning can be achieved. Word2Vec, which has models such as skip-gram and continuous bag of words (CBOW), is a widely used word embedding method. The likelihood of words occurring in proximity to one another is dependent on both models. Skip-gram allows a word to begin with and to anticipate the words that are likely to accompany it. By predicting a word that is likely to occur based on particular background terms, CBOW reverses that. The CBOW model

used to learn domain-specific word embeddings from large amounts of Arabic text was collected from the free online encyclopaedia Wikipedia (2,000,000 words vectors of Word2Vec) to create the corpus defined in [31].

The Word2Vec corpus was used to look up all words' vectors from the corpus. In [11] it is suggested that pre-trained word embeddings trained on very large text corpora, such as the free Word2Vec vectors trained on 100 billion Google news tokens, can provide universal features for use in natural language processing.

D. Deep Learning Models

This section presents the two types of deep learning models used in the experiments. Using deep learning in text classification is powerful for feature extraction. In this work, we improve the CNN model in [13] as shown in Fig. 3, and the LSTM model as shown in Fig. 4 with parameter optimization to improve the classification accuracy and compare it with the different ML methods (KNN, NB and SVM). Additionally, we propose the new CNN model as shown in Table III.

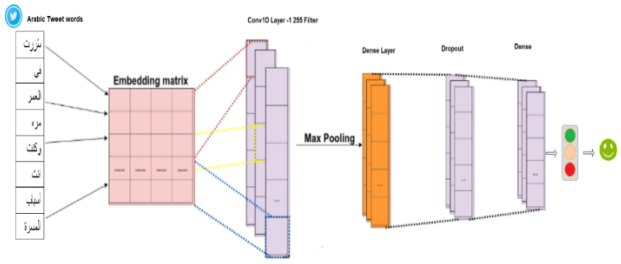

Fig. 3. CNN architecture

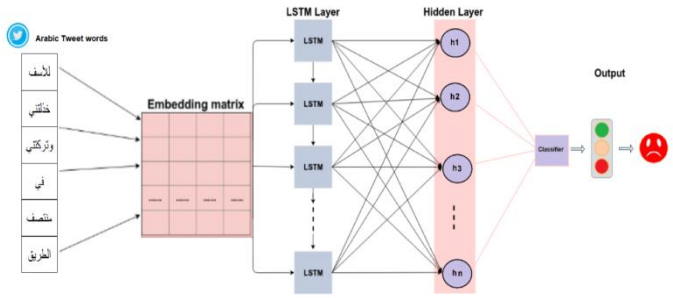

Fig. 4. LSTM architecture

IV. Results

This section presents the results of the conducted experiments. These three experiments, namely, the experiment based on the CNN model, the experiment based on the LSTM model, and the experiment based on the feature methods. These experiments were carried out based on different experiment settings.

A. Experiment Settings

This section describes the experiment settings of all three experiments. The experiment settings and hyperparameter tuning were performed to improve the accuracy of the performance model. In the first experiment, each CNN layer has various parameters such as the number of filters, kernel size, strides, padding, dropout rate, batch size and activation function. All CNN parameters are tuned and optimized to achieve high accuracy. In the second experiment, LTSM is used with the architecture. The embedding layer is used in the LSTM layers as in the CNN model. In the third experiment, the CNN and LSTM models with N-gram ranges were applied to achieve highly accurate model performance. The N-gram ranges method is applied with embedding vectors CBOW and skip-gram. All the parameters and details of the experiment are shown in Table II.

TABLE II. Experiments Settings

Parameter	Experiment 1	Experiment 2	Experiment 3
No. of layers	10	22	5
Input shape	300	300	300
Filters	64	64	64
Embedding vectors	CBOW	CBOW	CBOW
Max sequence length	300	300	300
Activation function	SoftMax	SoftMax	SoftMax
Optimization	Adam (learning rate = 0.001)	Adam (learning rate = 0.001)	Adam (learning rate = 0.001)

In these experiments, we applied the SoftMax activation function as shown in formula 1. The key benefit of using SoftMax is the performance probabilities range. The range will vary from 0 to 1, and the sum of all the odds will be equal to 1. If the SoftMax function is used for the multi-classification model it returns the probabilities of each class and the target class would have a high probability.

$$f(S)_i = \frac{e^{S_i}}{\sum_j^C e^{S_j}} \quad (1)$$

While for the loss function in all experiments a 'Categorical Cross-Entropy loss' has been utilized to train a CNN to output a probability over the C classes for each tweet, the target is three classes in the SoftMax activation function at the last layer. The mathematical formula of SoftMax activation and Cross-Entropy loss is shown in formula 2.

$$CE = -\log\left(\frac{e^{S_p}}{\sum_j^C e^{S_j}}\right) \quad (2)$$

where S_p is the CNN score for the positive class.

Before training the model, the dataset is divided into 25,130 training samples and validated on 6,283 samples.

TABLE III. LSTM ARCHITECTURE LAYERS

Layer (type)	Output Shape	Parameters #
input (Input Layer)	(None, 300)	0
embedding (Embedding)	(None, 300, 300)	17421900
LSTM	(None, 18)	22968
Batch Normalization	(None, 18)	72
Dense	(None, 3)	57
Total Parameters: 17,444,997 Trainable Parameters: 17,444,961 Non-trainable Parameters: 36		

In the third experiment, two experiments applied CNN and LSTM models to get a highly accurate model performance. The first experiment used the N-gram ranges method along with embedding vectors using CBOW. The second experiment used the N-gram ranges method to extract the required features, but with embedding vectors using skip-gram.

B. Experiments Results

This section explains the results of the experiments. Several experiments were conducted, and these can be categorized into three main experiments, namely, the experiment based on the CNN model, the experiment based on the LSTM model, and the experiment based on the feature methods.

1) Experiments based on CNN model: In this experiment, the CNN model applied two different architectures as shown in Table III, Table IV and Table V. Fig. 5 shows a scatter plot diagram for tweets labelled against the predicted labels. In addition, this figure explains the correlation with different epochs compared to the pre-processing approaches, and whether the emoji were retained or removed.

TABLE IV. CNN-1 ARCHITECTURE LAYERS

Layer (type)	Output Shape	Parameters #
input (Input Layer)	(None, 300)	0
embedding (Embedding)	(None, 300, 300)	17421900
CNN	(None, 300, 64)	96064
Global max pooling	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
batch_normalization_1	(None, 64)	256
dense_1 (Dense)	(None, 3)	195
dropout_2 (Dropout)	(None, 3)	0
Batch Normalization	(None, 3)	12
Dense	(None, 3)	12
Total Parameters: 17,518,439 Trainable Parameters: 17,518,305 Non-trainable Parameters: 134		

TABLE V. CNN-2 ARCHITECTURE LAYERS

Layer (type)	Output Shape	Parameters #
input (Input Layer)	(None, 300)	0
embedding (Embedding)	(None, 300, 300)	17421900
CNN-1	(None, 300, 64)	96064
CNN-2	(None, 300, 64)	20544
max pooling	(None, 150,64)	0
CNN-3	(None, 150, 64)	20544
CNN-4	(None, 150, 64)	20544
max pooling	(None, 75,64)	0
CNN-5	(None, 75, 64)	20544
CNN-6	(None, 75, 64)	20544
CNN-7	(None, 75, 64)	20544
max pooling	(None, 75,64)	0
CNN-8	(None, 37, 64)	20544
CNN-9	(None, 37, 64)	20544
CNN-10	(None, 37, 64)	20544
Global max pooling	(None, 75,64)	0
CNN-11	(None, 18, 64)	20544
CNN-12	(None, 18, 64)	20544
CNN-13	(None, 18, 64)	20544
Global max pooling	(None, 64)	0
Batch Normalization	(None, 64)	256
Dense	(None, 3)	195
Total params: 17,764,943 Trainable params: 17,764,815 Non-trainable params: 128		

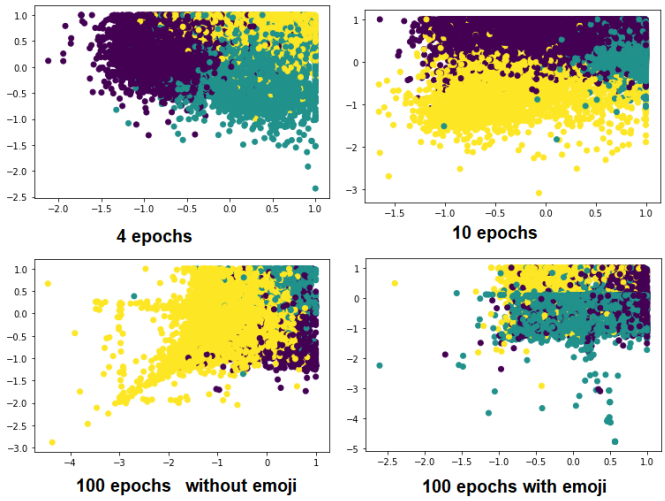

Fig. 5. Scatter plot diagram for tweets labels

2) Experiment based on the LSTM model: In this experiment, the LSTM model has been utilized. LSTM is used to enhance the memorisation of important information. In the text classification, LSTM is used in multiple word strings to identify the class to which it belongs. In this experiment, the dataset has been divided into three groups with different dataset sizes of 3,000 tweets, 15,000 tweets and 31,000 tweets, respectively. The results of this experiment are shown in Evaluation Results of Our Models' Accuracy with Different Dataset Sizes

Model	Accuracy-without emoji	Accuracy-with keeping emoji
SVM	81.79 %	82.5 %
NB	69 %	69.4 %
CNN-1	95 %	95.8 %
CNN-2	70 %	82.7 %
LSTM	96.6 %	95.5 %

TABLE VI. EVALUATION RESULTS OF OUR MODELS' ACCURACY WITH 31,000 TWEETS, WITH EMOJI RETAINED AND REMOVED

3) VI.

Model	Accuracy-without emoji	Accuracy-with keeping emoji
SVM	81.79 %	82.5 %
NB	69 %	69.4 %
CNN-1	95 %	95.8 %
CNN-2	70 %	82.7 %
LSTM	96.6 %	95.5 %

Table VII shows the results of our models' accuracy with the maximum tweets (31,000) and compares the classification accuracy when emoji are retained or removed from the content of the tweets.

TABLE VII. EVALUATION RESULTS OF OUR MODELS' ACCURACY WITH DIFFERENT DATASET SIZES

Model	Accuracy-without emoji	Accuracy-with keeping emoji
SVM	81.79 %	82.5 %
NB	69 %	69.4 %
CNN-1	95 %	95.8 %
CNN-2	70 %	82.7 %
LSTM	96.6 %	95.5 %

TABLE VIII. EVALUATION RESULTS OF OUR MODELS' ACCURACY WITH 31,000 TWEETS, WITH EMOJI RETAINED AND REMOVED

Dataset Size	SVM	NB	CNN-1(9L)	CNN-2(21L)	LSTM
Dataset-1(3000 Tweets)	81 %	23 %	93 %	89 %	90 %
Dataset-1 (15000 Tweets)	78.9 %	63 %	96 %	74.8	90 %
Dataset-1 (31000 Tweets)	82.5 %	69.4%	95.8%	83%	96.6 %

Training and testing accuracy for CNN and LSTM models is shown in Fig. 6, which also shows the accuracy of retaining or removing emoji. This comparison was implemented with different epochs, up to 100 epochs.

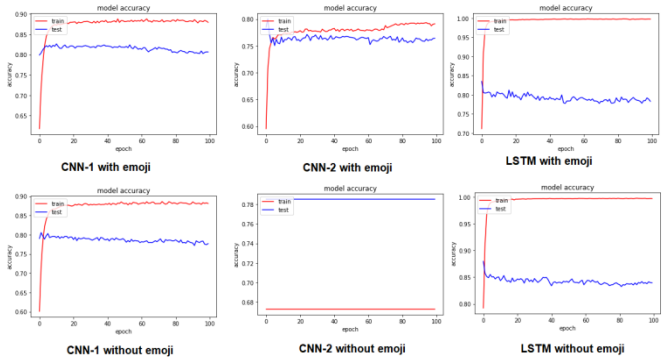

Fig. 6. Deep learning accuracy comparing retaining and removing emoji

4) Experiments based on features methods: In this experiment, two methods have been utilized to examine the accuracy of the model performance. In the first experiment, the two methods used are N-gram and CBOW. The experiments' results are shown in Table VIII.

TABLE IX. CBOW WITH N-GRAM RESULTS COMPARISON

Classifier	Without N-gram	N-gram range 1-2	N-gram range 1-3
SVM	82.22	82.64	83.2
LSTM	96.6	96.7	96.9
CNN-1	95.8	95.82	95.83
CNN-2	83	83.3	84
NB	69.4	69.7	70

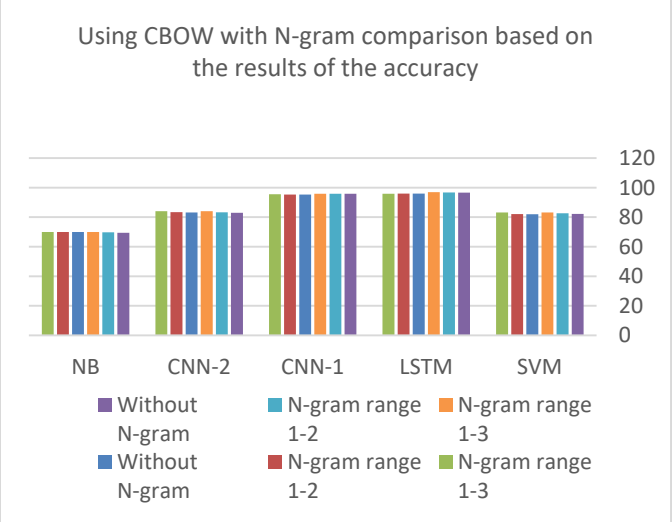

Fig. 7. CBOW and N-gram results comparison

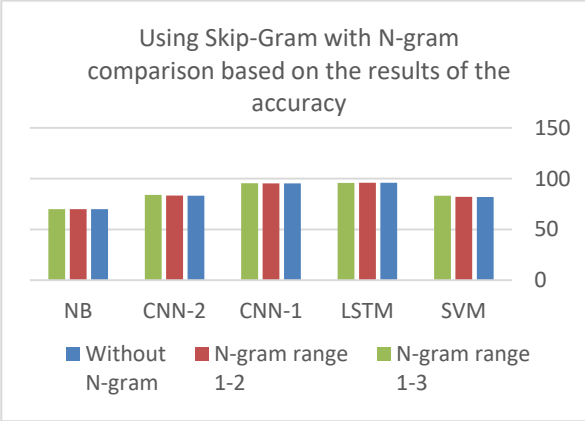

Fig. 8. Skip-gram and N-gram results comparison

In the second experiment, skip-gram and N-gram have been utilized.

TABLE X. SKIP-GRAM AND N-GRAM RESULTS COMPARISON

Classifier	Without N-gram	N-gram range 1-2	N-gram range 1-3
SVM	82	82.06	83.2
LSTM	**96**	**96**	**95.8**
CNN-1	95.3	95.3	95.5
CNN-2	83.2	83.4	84
NB	70	70	70

The accuracy of CNN-1 with emoji retained is 95.8%, and with emoji removed is 95%. The accuracy of CNN-2 with emoji retained is 82.7%, and with emoji removed is 70%. The CNN-1 model is more accurate than CNN-2. The accuracy of LSTM with emoji retained is 95.5% and 96.6% with emoji removed. When applied to SVM and NB the results are 82.5% and 69.4%, respectively. Fig. 8 shows the comparison between the deep learning accuracy with emoji retained and removed. By applying the same models with N-gram and with skip-gram, the results in Table VIII and Table IX, and represented in Fig. 7 and Fig. 8, show the LSTM model is better in both CBOW and skip-gram.

5) Experiment based on the AraBERT model: In this experiment the AraBERT for Sequence Classification is applied (Transformer-based Model for Arabic Language Understanding) with the following parameters:

TABLE XI. ARABERT PARAMETERS

Parameter	Value
Attention probs dropout prob	0.1
Hidden act	gelu
Hidden size	768
Initializer range	0.02
Intermediate size	3072
Layer norm eps	1e-12
Max position embeddings	512
Num attention heads	12
Num hidden layers	12
Transformers version	4.17.0
Vocab size	64000

After training the dataset on the AraBERT model using the parameter list as shown in Table X. The training accuracy (Fig. 9) of the AraBERT model is 0.92% when the test is applied on 3,505 tweets.

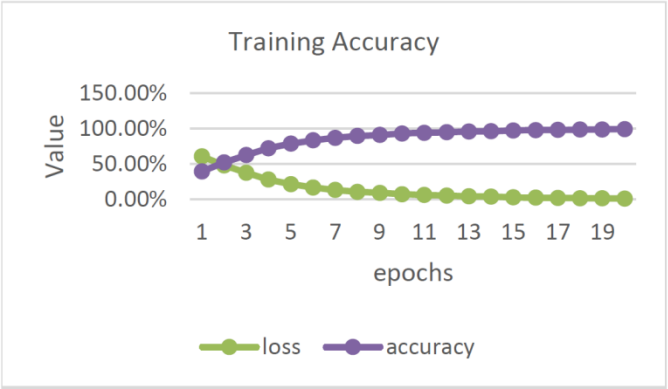

Fig. 9. Results of the training for AraBERT Model

V. CONCLUSION AND FUTURE WORKS

Results of this work show that there is improvement in CNN model accuracy by retaining emoji in text content, and LSTM is more accurate when emoji are removed. These results are summarized in Table II. The accuracy of CNN-1 with emoji retained is 95.8%, and with emoji removed is 95%. The accuracy of CNN-2 with emoji retained is 82.7%, whereas its accuracy with emoji removed is 70%. CNN-1 outperforms CNN-2 in terms of accuracy. The accuracy of LSTM when the emoji are retained is 95.5 %, and it is 96.6% when the emoji are removed. When we use SVM and NB, the outcomes are 82.5% and 69.4 %, respectively. The accuracy of the AraBERT model is 0.92%. In this work, we have shown that the LSTM architecture is the most suitable for the analysis of Arabic tweets. In the future, we can build a new system to analyse Arabic texts using the modern model GPT-3 from Open AI and apply the sentiment analysis on this dataset.

ACKNOWLEDGMENT

The authors would like to thank the Arab Open University for supporting this research paper.

REFERENCES

[1] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding," arXiv:2003.00104 [cs], Mar. 2021, Accessed: Dec. 01, 2022. [Online]. Available: http://arxiv.org/abs/2003.00104.

[2] Alhejaili, R., Alhazmi, E. S., Alsaeedi, A., & Yafooz, W. M. (2021, September). Sentiment Analysis of The Covid-19 Vaccine For Arabic Tweets Using Machine Learning. In 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) (pp. 1-5). IEEE.

[3] B. Alharbi et al., "ASAD: A Twitter-based Benchmark Arabic Sentiment Analysis Dataset," arXiv:2011.00578 [cs], Mar. 2021, Accessed: Dec. 01, 2022. [Online]. Available: https://arxiv.org/abs/2011.00578.

[4] H. Saif, Y. He, and H. Alani, "Semantic Sentiment Analysis of Twitter," in International semantic web conference, 2012, pp. 508–524.

[5] L. Alsudias and P. Rayson, "COVID-19 and Arabic Twitter : How can Arab World Governments and Public Health Organizations Learn from Social Media?," in ACL 2020 Workshop, 2020, pp. 1–9.

[6] S. A. Chowdhury, A. Abdelali, K. Darwish, J. Soon-Gyo, J. Salminen, and J. B. Jansen, "Improving Arabic text categorization using transformer training diversification," in In Proceedings of the fifth arabic natural language processing workshop, Dec. 2020, pp. 226–236.

[7] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Improving sentiment analysis in Arabic using word representation," in In 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), Mar. 2018, pp. 13–18.

[8] R. M. Duwairi and I. Qarqaz, "Arabic sentiment analysis using supervised classification," in In 2014 International Conference on Future Internet of Things and Cloud, Aug. 2014, pp. 579–583

[9] R. Ismail, M. Omar, M. Tabir, N. Mahadi, and I. Amin, "Sentiment analysis for Arabic dialect using supervised learning," in In 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Aug. 2018, pp. 1–6.

[10] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Arabic language sentiment analysis on health services," 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), Apr. 2017, doi: 10.1109/asar.2017.8067771.

[11] A. Elnagar, L. Lulu, and O. Einea, "An Annotated Huge Dataset for Standard and Colloquial Arabic Reviews for Subjective Sentiment Analysis," Procedia Computer Science, vol. 142, pp. 182–189, 2018, doi: 10.1016/j.procs.2018.10.474.

[12] Y. Zhang, J. Zheng, Y. Jiang, G. Huang, and R. Chen, "A Text Sentiment Classification Modeling Method Based on Coordinated CNN-LSTM-Attention Model," Chinese Journal of Electronics, vol. 28, no. 1, pp. 120–126, Jan. 2019, doi: 10.1049/cje.2018.11.004.

[13] A. Dahou, M. A. Elaziz, J. Zhou, and S. Xiong, "Arabic Sentiment Classification Using Convolutional Neural Network and Differential Evolution Algorithm," Computational Intelligence and Neuroscience, vol. 2019, pp. 1–16, Feb. 2019, doi: 10.1155/2019/2537689.

[14] M. Beseiso and H. Elmousalami, "Subword Attentive Model for Arabic Sentiment Analysis," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 19, no. 2, pp. 1–17, Mar. 2020, doi: 10.1145/3360016.

[15] M. Baali and N. Ghneim, "Emotion analysis of Arabic tweets using deep learning approach," Journal of Big Data, vol. 6, no. 1, Oct. 2019, doi: 10.1186/s40537-019-0252-x.

[16] K. Abu Kwaik, S. Chatzikyriakidis, S. Dobnik, M. Saad, and R. Johansson, "An Arabic Tweets Sentiment Analysis Dataset (ATSAD) using Distant Supervision and Self Training," ACLWeb, May 01, 2020. https://www.aclweb.org/anthology/2020.osact-1.1 (accessed Dec. 01, 2022).

[17] I. Guellil, H. Saâdane, F. Azouaou, B. Gueni, and D. Nouvel, "Arabic natural language processing: An overview," Journal of King Saud University - Computer and Information Sciences, Feb. 2019, doi: 10.1016/j.jksuci.2019.02.006.

[18] R. M. K. Saeed, S. Rady, and T. F. Gharib, "Optimizing Sentiment Classification for Arabic Opinion Texts," Cognitive Computation, vol. 13, no. 1, pp. 164–178, Jan. 2021, doi: 10.1007/s12559-020-09771-z.

[19] S. S. Aljameel et al., "A Sentiment Analysis Approach to Predict an Individual's Awareness of the Precautionary Procedures to Prevent COVID-19 Outbreaks in Saudi Arabia," International Journal of Environmental Research and Public Health, vol. 18, no. 1, p. 218, Dec. 2020, doi: 10.3390/ijerph18010218.

[20] I. Abu Farha and W. Magdy, "A comparative study of effective approaches for Arabic sentiment analysis," Information Processing & Management, vol. 58, no. 2, p. 102438, Mar. 2021, doi: 10.1016/j.ipm.2020.102438.

[21] M. O. Hegazi, Y. Al-Dossari, A. Al-Yahy, A. Al-Sumari, and A. Hilal, "Preprocessing Arabic text on social media," Heliyon, vol. 7, no. 2, p. e06191, Feb. 2021, doi: 10.1016/j.heliyon.2021.e06191.

[22] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning Word Vectors for 157 Languages," arXiv preprint arXiv:1802.06893, Aug. 2018.

[23] A. Brahmi and A. E. Abdelkader, "Arabic texts analysis for topic modeling evaluation," Inf. Retr. Boston., vol. 15, no. 1, pp. 33–53, 2012, doi: 10.1007/s10791-011-9171-y.

[24] Z. Kaoudja, B. Khaldi, and M. L. Kherfi, "Arabic artistic script style identification using texture descriptors," CCSSP 2020 - 1st Int. Conf. Commun. Control Syst. Signal Process., pp. 113–118, 2020, doi: 10.1109/CCSSP49278.2020.9151569.

[25] L. Srinivasan and C. Nalini, "An improved framework for authorship identification in online messages," Cluster Comput., vol. 22, no. s5, pp. 12101–12110, 2019, doi: 10.1007/s10586-017-1563-3.

[26] M. Martinc and S. Pollak, "Combining n -grams and deep convolutional features for language variety classification," vol. 2013, pp. 607–632, 2019, doi: 10.1017/S1351324919000299.

[27] T. K. Mustafa, A. A. Abdul Razzaq, and E. A. Al-Zubaidi, "Authorship Arabic Text Detection According to Style of Writing by Using (SABA) Method," Asian J. Appl. Sci., vol. 5, no. 2, pp. 483–490, 2017, doi: 10.24203/ajas.v5i2.4750.

[28] M. H. Altakrori, F. Iqbal, B. C. M. Fung, S. H. H. Ding, and A. Tubaishat, "Arabic authorship attribution: An extensive study on twitter posts," ACM Trans. Asian Low-Resource Lang. Inf. Process., vol. 18, no. 1, 2018, doi: 10.1145/3236391.

[29] Alhujaili, R. F., & Yafooz, W. M. (2022, May). Sentiment Analysis for YouTube Educational Videos Using Machine and Deep Learning Approaches. In 2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI) (pp. 238-244). IEEE.

[30] N. M. A. Rahman, S. A. Nouh, and R. H. A. Alez, "A Language Model for Arabic Texts Disambiguation using Deep Learning," vol. 6, no. 2, pp. 1–16, 2019.

[31] N. M. Abdel and R. Ibrahim, "A New Model for Arabic Text Clustering by Word Embedding and Arabic Word Net," Saudi J. Eng. Technol., vol. 6272, pp. 401–406, 2019, doi: 10.36348/SJEAT.2019.v04i10.001.

[32] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "ASA: A framework for Arabic sentiment analysis," Journal of Information Science, p. 016555151984951, May 2019, doi: 10.1177/0165551519849516.

[33] O. Oueslati, E. Cambria, M. B. HajHmida, and H. Ounelli, "A review of sentiment analysis research in Arabic language," Future Generation Computer Systems, vol. 112, pp. 408–430, Nov. 2020, doi: 10.1016/j.future.2020.05.034.

[34] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow, "A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios," arXiv:2010.12309 [cs], Apr. 2021, Accessed: Dec. 01, 2022. [Online]. Available: http://arxiv.org/abs/2010.12309

[35] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in neural information processing systems, 2013, vol. 26.

[36] N. M. Ibrahim, "Text Mining using Deep Learning Article Review," vol. 9, no. 9, pp. 1916–1933, 2018.