

Sentiment Detection on Sudanese Political Tweets Using Deep Learning Approach

Fatima Salih Ibrahim, Eltyeb Elsamani, Shazali Siddig

Abstract—Sentiment detection of Arabic tweets is interesting research topic and it enables scholars to analyze huge resources of shared opinions in social media websites such as Facebook and tweeter. It is one of the more complex natural language processing tasks due to the informal noisy contents and the rich morphology of Arabic language. Many studies have been investigated for Arabic sentiment analysis. However, most of these works ignore analyzing political Arabic tweets, specifically the Sudanese Arabic dialect. In this paper, a deep learning-based approach is proposed for political Arabic Sentiment Analysis (PASA) of tweets. Our approach employed word embedding with convolutional neural network and long short-term memory network techniques to represent the tweets and extract the feature vectors. After that, the feature vectors are fed to the classifier to detect the sentiments. We conducted a number of experiments using a set of performance evaluation metrics on a political twitter dataset to test the proposed PASA approach. Experimental results showed that the proposed approach outperformed the baseline methods in terms of precision, recall and F-score metrics.

Index Terms— Sudanese revolution, Arabic tweets, convolution neural networks, machine learning, word embedding.

I. INTRODUCTION

Nowadays, Sentiment Analysis (SA) is usually studied as research area, and it defined as the application of text mining, natural language processing (NLP), and computational linguistics techniques for extracting people's sentiments or feelings about an event, product, or others [1, 2]. In general, as presented in [3] SA imposes identifying four components namely entity, its aspect, opinion/sentiment holder, and his opinion. Further, the extracted sentiments can be classified to either subjective or objective. The subjective text also can be classified to positive or negative sentiments.

Most previous SA studies have been carried out on natural languages, such as Turkish, English and Arabic [4-6]. Dealing with NLP in Arabic text is still at the beginning stages and challenging task due to its structure complexity, history, and different cultures [7].

In the literature, a large number of proposed techniques are used to address the SA task. Most of them are intended to

accomplish SA in English, which is the science language [8]. These techniques are either the rule-based or machine learning (ML) techniques. The rule-based approach extracts the sentiment words and calculates its polarities based on manual design rules [9]. On the contrary, ML models utilized annotated data with statistical and semantic features to predict the classes/labels of unseen data [10]. These techniques have been adapted to others' languages, such as Arabic [8] and achieved good classification performance, but the feature engineering process cannot generalize well in all domains.

Moreover, deep learning techniques have been used recently by some studies to detect sentiments of Arabic reviews [11-15]. However, most of the previous studies applied deep learning in product review domain and have not been applied in political Arabic domain. As a result, in this paper, a deep learning approach is proposed to detect sentiment in political tweets, specifically Sudanese political Arabic tweets extracted during Sudanese revolution in 2018. The proposed approach combines convolutional neural network and recurrent neural network with word representation technique to capture the local n-gram features and dependency of the words in sentences. Further, due to lack of political Arabic dataset, we build our own dataset to test the performance of the proposed approach and compare the result with classical ML methods.

The rest of this paper is organized as follows. Section 2 shows related work and section 3 presents the proposed deep learning approach. Experimental setup and results are showed in Section 4. Section 5. draws up the conclusion.

II. RELATED WORKS

The proliferation of social media, blogs, comments and reactions of online users makes SA very interesting research field. SA has become an important field, since it aids for identifying and extracting valuable information from various sources of data. Generally, the SA task can be accomplished using four main techniques namely rule based, ML-based method, lexical-based method and deep learning techniques.

Rule based method is time-consuming and very difficult to form the desired rules for sentiment detection. On the other hand, ML and lexicon-based models have been successfully employed to classify the sentiments of Arabic and English text. For example, Mishra et. al. [16] developed unsupervised models based on lexicons and estimated a sentiment with regards to the Digital India mission by calculating the ratio of occurrence of two types of keywords with respect to one another.

Fatima Salih Ibrahim, Department of Computer Science, Shaqra University, Shaqra, Saudi Arabia

Eltyeb Elsamani, Department of Computer Science, University of Alneelain/ Faculty of Computer Science and Information Technology, Khartoum, Sudan

Shazali Siddig, Department of Computer Science, University of Alneelain/ Faculty of Computer Science and Information Technology, Khartoum, Sudan

Marchetti-Bowick et al.[17]proposed a supervised learning model by using distant supervision and train a classifier on the political tweets for USA election. In [18], The authors developed a SA model based on naïve bays algorithm with set of features extracted from Arabic lexicon and text. A. Jain and P. Dandannavar [19]proposed a hybrid approach by combining a lexical-based approach with a learning-based for SA.

There are very few studies have examined the application of deep learning techniques in SA of Arabic. Ahmed Oussous et al.[15]proposed to use CNN and LSTM for Arabic SA. They compared their results with traditional ML classifiers such as Support Vector Machines (SVM) and the results proved the efficiency of deep learning models. Al-Azani & El-Alfy[13]used various deep learning models based on CNN and LSTM for Arabic sentiment detection on microblogs. They utilized continuous bag of words and skip gram models to represent the text and their experiment showed that LSTM performed better than CNN. Alayba et al. [20]combined CNN with Arabic Health Twitter Lexicons to classify sentiment in Arabic. They used Word2Vec as the embedding technique, and achieved promising results. However, most of these works focused on English whereas Arabic did not receive much attention until recently. Also, these approaches ignore extracting political Arabic tweets, and we believe it is difficult to analyze such tweets due to many challenges of the Arabic language.

III. PROPOSED APPROACH

The aim of this research is to develop a PASA framework that detects the sentiments of political Arabic tweets. This is achieved by first implementing a deep learning model for Arabic text and then comparing various ML techniques with the proposed approach. In NLP, a standard approach for data modeling is to convert a sequence of text as a feature vector using neural network methods such as CNN and LSTM. CNN has frequently been used for syntactic and semantic representations of the text in many NLP tasks and has attained excellent classification performance than classical NLP approaches. Generally, in CNN algorithm, the input size is bigger than the filter size. As a result, the output interacts with a narrow window of the input and highlights the local lexical connections of the n-gram. LSTM is useful for sequence modeling and can successfully remember long-sequence dependencies but may fail to extract the local n-gram context. Moreover, it has been found that using a vector from either CNN or LSTM to encode an entire sequence is not sufficient to capture all the significant information from a text. Therefore, we propose to combine CNN and LSTM to exploit the benefits of both methods.

The overall architecture of our proposed framework comprises of three main phases: data preparation, feature extraction, and classification phase. The data preparation phase includes collecting, preprocessing, and labeling the political Arabic tweets. The feature extraction phase employed word embedding with CNN and LSTM to construct the feature vectors. The classification phase employed softmax function to detect the sentiments of the political Arabic tweets.

Fig1 shows the architecture of our model. First, the input sentence is represented as a matrix by using Word2Vec

technique. Then, local n-gram features are extracted by using the convolutional and max-pooling layers. LSTM is then used to compose the features for producing the results. The architecture can extract both local information and long-distance dependency across tweets. The following is the detailed description of each part of the proposed PASA architecture.

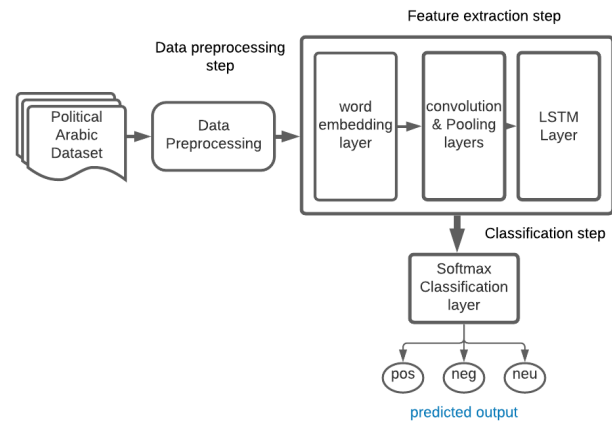


Fig. 1. The architecture of the proposed PASA model

A. Data preparation step

In SA of tweets process, the normal step is to collect tweets by specifying a keyword to retrieve all tweets that are similar to the entered keyword. Tweets can be collected from many sources; in our paper, we utilized Twitter's API to collect political Arabic tweets related to Sudanese revolution that was started on December 19, 2018. The dataset was collected using hash tags related to the Sudanese revolution, such as # (مدن السودان تنتفض # تسقط بس # اعتصام القيادة). The collected dataset is highly imbalanced: 1822 positive, 243 negative and 420 neutral tweets as shown in Table 1.

Table 1: Distribution of the collected Tweets

Class Name	No of Tweets
Neutral	420
Negative (Military)	243
Positive (Civilian)	1822
Total	2485

Four annotators (graduate students with computer science and political science background) were used to label the data. In this step, we gave the annotators clear instructions to label the dataset into positive (we call it civilian), negative (military) and neutral classes as shown in Table 1.

After that, we preprocess the data to be suitable for machine and deep learning classifiers. Further, the preprocessing can be useful for data quality improving and reducing the amount of trivial noise (i.e., deleting tweets that are irrelevant to events). In this paper, we cleaned the data by removing irrelevant information, such as URLs, special characters, duplicated characters, and non-Arabic letters. Also, we remove stop words such as prepositions and articles that occur frequently and may not be useful in discrimination between data classes/labels. Finally, the words were reduced to their stem from where structural information is used to match different variants of words.

B. Feature extraction

After cleaning and preprocessing the data, we transform our tweets' data into feature vectors, which would be used as input to the classifiers. We use Word2Vec with CNN and LSTM to extract the feature vectors. The detailed description of the feature extraction process is presented using the following layers.

Embedding layer. The input is a tweet (sentence) S which is a sequence of words w_1, w_2, \dots, w_d where each word is drawn from a finite-size vocabulary V . Each word is converted into a low-dimensional vector via a lookup table: $x_i = E_w(w_i)$, where $W \in \mathcal{R}^{d_w \times |V|}$ is the word embedding matrix and d_w is each word embedding dimension. As a result, the input tweet sentence is represented as a matrix where each column corresponds to a word embedding.

Convolutional layer. In this layer, we use a set of m filters to extract local n-gram features from the embedding matrix. The feature map output z_i^l can be produced by using a sliding window of size w , and a filter $F_l (1 \leq l \leq m)$ as follows:

$$z_i^l = f(X_{i:i+w-1} * W^l + b^l) \quad (1)$$

Where $*$ denotes a convolution operation, $W \in \mathbb{R}^{w \times d}$ is the weight matrix, b is a bias, and $X_{i:i+w-1}$ denotes the word vectors. The results of the filter F_l will be $z^l \in \mathbb{R}^d$, where z_i^l is the i -th element of z^l . Rectified Linear Units was used as activation function.

Max-Pooling layer. We apply max-pooling operation for each filter to capture the most important features. The max value is extracted from each row of z^l , which will generate the next representation vector. To prevent the overfitting problem in CNN, a dropout strategy was applied.

Bi-LSTM layer. Single direction LSTM is insufficient and may not be able to exploit the contextual information from the subsequent words. Therefore, the Bi-LSTM was used to utilize both prior and future contexts by processing the sequence on both forward and backward directions. The bellow equations illustration how LSTM estimates the hidden states h_t and outputs o_t .

$$f_t = \sigma(W_f \cdot z_t + U_f \cdot h_{t-1} + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot z_t + U_i \cdot h_{t-1} + b_i) \quad (3)$$

$$\tilde{c}_t = \tanh(W_c \cdot z_t + U_c \cdot h_{t-1} + b_c) \quad (4)$$

$$c_t = i_t \otimes \tilde{c}_t + f_t \circ c_{t-1} \quad (5)$$

$$o_t = \sigma(W_o \cdot z_t + U_o \cdot h_{t-1} + b_o) \quad (6)$$

$$h_t = o_t \otimes \tanh(\tilde{c}_t) \quad (7)$$

Where z_t and h_t are the input to Bi-LSTM and output vectors at time t , respectively. $W_f, W_i, W_o, U_i, U_f,$ and U_o are weight matrices and $b_i, b_f, b_c,$ and b_o are bias vectors. The symbol \otimes represents element-wise multiplication and σ denotes the sigmoid function.

C. Classification phase (output layer)

We fed the outputs of the previous layer to the fully connected layer. The softmax function generates a distribution over the class labels for tweet sentiment classification. This layer produces the final classification results and is defined as:

$$p(y|x) = \text{softmax}(W_d \cdot o + b_d) \quad (8)$$

Where o is the final representation word vector, y is the predicted label of the tweet sentiment, and W_d and b_d respectively denote the weights and bias.

IV. EXPERIMENTAL SETTINGS AND RESULTS ANALYSIS

This section presents the experimental settings and configurations of the proposed model on our dataset. The experiments are conducted on the Keras framework running on Python

A. Experimental settings

In all experiments, we applied 10-fold cross validation. Word embedding dim was set to 300. We used Adam optimizer to train our model and the losses were calculated by using categorical cross entropy. For the CNN, the adopted convolutional layer used multiple filters of width (3, 4, 5), number of filters was set to 300, and ReLus as activation function, dropout was set to 0.5 to minimize the overfitting problem. Padding was set to zero. For the LSTM settings, the hidden state dimensionality was set to 128, and sigmoid function was used as an activation function. The number of epochs was set to 50 in the entire architecture of the model and the batch size set to 64. For other baselines (SVM, Random forest, ensemble learning, and naïve bays (NB), the dataset was split by 80%, 10%, and 10 % for training, development, and test sets respectively and we used grid search to select best parameters.

B. Baseline models and evaluation metrics

SVM. SVM is a ML model derived from statistical learning theory with the goal of reducing the computational complexity and test error. SVM is a supervised learning algorithm for binary classification purpose and other similar tasks. The SVM works by identifying a separating hyper-plane with optimality between the labels of a training data.

Random Forest Classifier (RFC). A random forest is a ML model and is a combination of tree-structured classifiers. This algorithm creates many random trees on the subset of data and combines the results of all trees. One of the advantages of an RFC in classification systems is to accurately give best performance compared to simple decision tree models.

Multinomial Naïve bays. Naïve Bayes is a classification model based on Bayes' Theorem with an hypothesis of independence among predictors models. This algorithm assumes that the presence of a particular variable in a specific class is unrelated to the presence of any other features. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability.

K-Nearest Neighbor (K-NN). K-NN is defined as an instant-based learning model that classified objects based on

nearest feature space in the training data. In this model, the training dataset can be converted into multi-dimensional feature space. Then these features can be divided into regions based on the class of the training dataset. For the classification or prediction, A feature space point will be assigned to a specific class if it is the most frequent class among the k-nearest training data.

Ensemble methods. Ensemble methods (EM) aim to combine the predictions of multiple machine-learning classifiers to obtain a classification model with better performance. The advantages of combining classifiers are to reduce the variance and bias of classification and remove the dependency of results on features of a single training dataset. An ensemble model normally works by training different ML classifiers on a dataset and each classifier makes predictions separately. The predictions of the best classifiers are then combined in an ensemble model to make a final prediction result.

Evaluation metrics. We employed the confusion matrix to visualize the algorithms' performance. The components of the confusion matrix can be used for finding the values of some evaluation metrics such as precision, recall, and F-score as formalized in the following.

Precision is the ratio of total number of the examples correctly labeled as positive to the total number of positively classified example.

$$Precision = \frac{TP}{(TP + FP)} \times 100 \quad (1)$$

Recall is the ratio of total number of the positive labeled examples to the total examples that are truly positive.

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (2)$$

F-score is described as the harmonic mean or the weighted average for both the precision and recall obtained.

$$F - score = 2 \times Precision \times Recall / (Precision + Recall) \quad (3)$$

where, TP (true positive) is described as the examples, which are positive and have been classified as positive by the classifiers, FN (false negative) represents positive examples, but classifier classifies them as negative. True Negative (TN) represents the examples that are negative and classified as negative examples by the classifiers, false positive (FP) represents negative examples, but classifier classifies them as positive.

C. Experimental results and discussion

As mentioned earlier, the proposed approach employs Word2Vec with CNN and LSTM to generate feature vectors. Generally, the proposed approach outperforms the baseline models on the dataset. From Table 2, we observe that, the proposed approach can accurately classify opinions with at least 92 % F-score. Moreover, our model achieves better results than traditional ML classifiers with an improvement of 9% F-score on the dataset. Specifically, our model improves results by 4% in terms of F-score (92%) comparing with the CNN F-score (88%). We further noted that compared with the traditional classifiers the proposed approach is higher in its performance and reported an increase in F-score by 9%. Furthermore, the proposed model outweighs classical classifiers, LSTM, and CNN in all metrics. Our model achieves an F-score of 92%, which is equal to 4% increase over the CNN. Moreover, our approach attains a higher than

6% increase of F-score over the rest of the baseline approaches.

Moreover, comparing the baseline ML approaches with each other, SVM with Word2Vec features leads to best results than MultinomialNB. In addition, Random Forest gives better results than NB, KNN in terms of F-score. The ensemble learning model achieves 83% F-score, which is a superior classification performance compared to other baseline ML methods. Additionally, comparing the baseline methods with each other, on our task, SVM with word embedding features (Table 2) leads to better results than SVM and NB with TF-IDF features (Fig2). Similarly, as presented in table2 ensemble model with Word2Vec features leads to best results than all classical ML methods with TF-IDF syntactic features as presented in Fig2.

Table2: Classifiers performance using Word2Vec

Method	Precision	Recall	F-score
SVM	0.85	0.81	0.82
MultinomialNB	0.55	0.82	0.61
K-nearest Neighbors	0.84	0.76	0.78
Random Forest	0.92	0.62	0.78
Ensemble learning	0.87	0.79	0.83
LSTM	0.89	0.81	0.85
CNN	0.92	0.83	0.88
Proposed approach	0.95	0.88	0.92

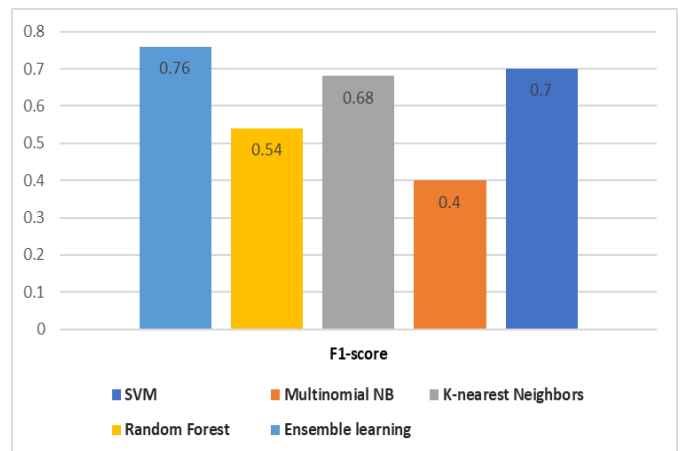


Fig2. Shows classifiers performance using TF-IDF feature.

Moreover, a number of experiments were conducted by using the dropout strategy to see its impact on the classification performance of our proposed approach. Table 3 shows the effect of dropout on the results of our model. We noted that using dropout strategy improves the classification performance of the model by 2 % F-score.

Table 3: The effect of dropout strategy on the performance of our approach

Option	Precision	Recall	F-score
Non-dropout	0.92	0.86	0.90
Dropout	0.95	0.88	0.92

Finally, the results of this paper approve that the proposed approach produces more effective classification results than traditional classifiers. Also, we believe that representing Arabic tweets using word embeddings is suitable for capturing semantic information of the sentence unlike the

TF-IDF feature, which is purely reliant on the statistics to represent the tweets.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a deep learning model for detecting political sentiment of Arabic tweets by exploiting Word2Vec, CNN and LSTM algorithms. The proposed model has three steps namely data preprocessing, feature extraction, and classification phase. We present a network architecture to represent the tweets and improve the classification performance. Further, the proposed model was evaluated by using political Arabic dataset and the results show that our model outperforms the classical ML techniques' approaches. Further research will be investigating the benefits of imbalance techniques with our model for improving the classification performance.

ACKNOWLEDGMENT

First author collected the dataset and labeled it and also conducted the experiments and wrote the paper. The second and third authors guided the first author and helped in the whole process from preparing the dataset and conducting experiments to writing the paper. All authors have approved the final version.

REFERENCES

1. Liu, B.J.H.o.n.l.p., *Sentiment analysis and subjectivity*. 2010. 2(2010): p. 627-666.
2. Medhat, W., A. Hassan, and H.J.A.S.e.j. Korashy, *Sentiment analysis algorithms and applications: A survey*. 2014. 5(4): p. 1093-1113.
3. Liu, B.J.S.l.o.h.l.t., *Sentiment analysis and opinion mining*. 2012. 5(1): p. 1-167.
4. Mahmoud, A. and T. Elghazaly, *Using twitter to monitor political sentiment for Arabic slang*, in *Intelligent Natural Language Processing: Trends and Applications*. 2018, Springer. p. 53-66.
5. Kaya, M., G. Fidan, and I.H. Toroslu, *Sentiment analysis of turkish political news*. in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*. 2012. IEEE.
6. Basile, P. and N. Novielli, *Uniba: Sentiment analysis of English tweets combining micro-blogging, lexicon and semantic features*. in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. 2015.
7. Abdulla, N.A., et al. *Arabic sentiment analysis: Lexicon-based and corpus-based*. in *2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AECT)*. 2013. IEEE.
8. Korayem, M., D. Crandall, and M. Abdul-Mageed, *Subjectivity and sentiment analysis of arabic: A survey*. in *International conference on advanced machine learning technologies and applications*. 2012. Springer.
9. Yang, C.-S. and H.-P. Shih, *A Rule-Based Approach For Effective Sentiment Analysis*. in *PACIS*. 2012.
10. Kawade, D.R., K.S.J.I.J.o.E. Oza, and Technology, *Sentiment analysis: machine learning approach*. 2017. 9(3): p. 2183-2186.
11. Heikal, M., M. Toriki, and N.J.P.C.S. El-Makky, *Sentiment analysis of Arabic Tweets using deep learning*. 2018. 142: p. 114-122.
12. Mohammed, A., R.J.S.N.A. Kora, and Mining, *Deep learning approaches for Arabic sentiment analysis*. 2019. 9(1): p. 1-12.
13. Al-Azani, S. and E.-S.M. El-Alfy, *Hybrid deep learning for sentiment polarity determination of arabic microblogs*. in *International Conference on Neural Information Processing*. 2017. Springer.
14. Altaher, A.J.I.J.o.A. and A. Sciences, *Hybrid approach for sentiment analysis of Arabic tweets based on deep learning model and features weighting*. 2017. 4(8): p. 43-49.

15. Oussous, A., et al., *ASA: A framework for Arabic sentiment analysis*. 2020. 46(4): p. 544-559.
16. Mishra, D., M. Venugopalan, and D.J.P.C.S. Gupta, *Context specific Lexicon for Hindi reviews*. 2016. 93: p. 554-563.
17. Marchetti-Bowick, M. and N. Chambers, *Learning for microblogs with distant supervision: Political forecasting with twitter*. 2012, MICROSOFT CORP SAN FRANCISCO CA.
18. El-Beltagy, S.R., et al. *Combining lexical features and a supervised learning approach for Arabic sentiment analysis*. in *International Conference on Intelligent Text Processing and Computational Linguistics*. 2016. Springer.
19. Dandannavar, P. and S.J.I.S.P. Mangalwede, *Sentiment Analysis of Real World Big Data—A Review of General Approaches*. 2017. 10: p. 185-192.
20. Alayba, A.M., et al. *Arabic language sentiment analysis on health services*. in *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*. 2017. IEEE.



Fatima Salih received her bachelor degree in Information Technology from Computer Man College, Sudan. She received her masters' degree in Computer Science from University of Gezira, Sudan. Currently, she is a lecturer at Shaqra University, Saudi Arabia. Her research interest includes machine learning and deep learning.



Shazali Siddig received his bachelor degree in Computer Science and his master in Information Technology from University of AL-Neelain, and received his PhD in Information Technology from AL-Neelain University, Sudan. Currently, he is Assistant professor and researcher at Data Science Group at the Department of Information Technology, AL-Neelain University. His research interest includes machine learning, database systems, data science.



Eltyeb Elsamani received his bachelor degree in Computer Science from University of Al-neelain, master in Computer Science from Al-neelain University, and PhD from University of Al-neelain, Sudan. Currently he is an associated professor in faculty of Computer Science and Information Technology, University of Al-neelain Sudan. His research interest includes machine learning, text analysis, database systems, and sentiment analysis.