# Arabic Sentiment Analysis Using Naïve Bayes and CNN-LSTM

Dima Suleiman*, Aseel Odeh and Rizik Al-Sayyed
E-mail: dima.suleiman@ju.edu.jo, odeh.aseel99@yahoo.com, r.alsayyed@ju.edu.jo
King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan

**Keywords:** NB, sentiment analysis, polarity, deep learning

**Received:** May 24, 2022

*Sentiment analysis (SA) is a useful NLP task. There are hundreds of Arabic sentiments analysis systems. However, because of the morphological nature of the Arabic languages, there are still many challenges that need more work. In this paper, two classifiers have been used: Naive Bayes and CNN-LSTM models. The experiments are conducted on Arabic tweets dataset that consists of 58k tweets written in several dialects, the same preprocessing steps have been done before fitting the models. The experimental results show that the deep Learning CNN-LSTM classifier fits better for this task which achieved an accuracy of 98% while Naive Bayes achieved 87.6%.*

*Povzetek: Dva klasifikatorja: Naivni Bayes in globoke nevronske mreže sta bila uporabljena za analizo mnenj nad arabskimi besedili.*

## 1 Introduction

Sentiment analysis (SA); also called opinion mining; is a core research subject in artificial intelligence (AI). It involves applying computational approaches for building a system to examine and classify opinions about products, comments, reviews, and tweets. Sentiment analysis is a type of natural language processing (NLP) that studies the subjective attitude of a natural texts [1-3].

The importance of sentiment analysis covers many domains, where education is one of the fields in which SA can be utilized. By understanding and finding out what students prefer most about a course, instructor, or teaching methodology, this can be considered by the respective institutions [4]. SA is also useful in some of business fields such as the analysis of a product that can be done quickly. It can be considered as a tool that analyzes the customer's responses to the new products so that it can help in making decisions in the next stages. SA can also be used in a variety of other possible domains such as health services, financial services, social and political events in elections [5].

Work on SA started in early 2000s, it started with the sentiment of movies' reviews. SA gained high attention from researchers who developed it to span many topics on social media [6]. Arabic NLP still at the beginning phases [7], in SA most of works focused on English while Arabic did not receive the required attention, due to many challenges. One of the challenges is that Arabic has a vague semantics, which makes the meaning very difficult to be grasped and analyzed. Furthermore, formal Arabic can be categorized as Classical Arabic (CA) and Modern Standard Arabic (MSA) However, Arabic speakers use informal Arabic language dialects, which defers from MSA and varies from one dialect to another in terms of vocabulary, by country.

One of the popular social platforms in the Arab world is Twitter. In March 2014, statistical reports by the Dubai school of government showed that there were more than 5.7 million Arab users on Twitter. Saudi Arabians post 40% of all Arab user's tweets, Egyptians post 17% and Kuwaitis post 10% [8]. Twitter is an attractive network source for SA, it allows people to share real-time tweets to discuss their opinions in many important fields. Therefore, we relied on the data extracted from Twitter, as it is rich of opinions in various fields and dialects.

In this paper, an approach that utilizes Deep Learning and Arabic Natural Language Processing (NLP) to classify Arabic tweets into positive and negative opinions automatically is proposed. We stand on two approaches; the first one is the Naïve Bayes classifier while the other is the combination between Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) models.

The rest of this paper is organized as follows. The related work of Arabic sentiment analysis (ASA) is covered in section 2, followed by section 3 that describes the dataset that is used in experiments. Section 4 investigates the proposed methodology. The experimental results are discussed in section 5. Finally, section 6 and section 7 discuss the conclusion and the future orientation respectively.

## 2 Related work

Arabic language has gained fewer efforts compared to other languages. However, hundreds of studies have been worked for ASA in several approaches using a lexicon-based method, machine learning classification and deep

[*] Corresponding author

learning [7] Arabic sentiment analysis (ASA) used many types of data extracted from social media networks and reviews on specific products, events and services.

In many systems, SA was modeled as a classification problem using machine learning such as [9], which used Support Vector Machine (SVM) classifier on MSA and Egyptian Dialectal Arabic tweets and indicated high performance level with accuracies over 95%. Another model [10] used four machine learning classifications. The results showed that SVM model achieved the highest accuracy of 96.06%. Al-Horaibi et al. [8] used semantic approach that classified the data into 3 classes which are: positive, negative and neutral. They translated Arabic tweets into English using Google Translate with accuracy of 60.25%. Also, [11] combined a corpus-based and lexicon-based on Arabic MSA and Saudi dialects tweets They used three hybrid classifiers where the highest accuracy was 69.9%. SentiArabic [12] is a Sentiment Analyzer for MSA, the authors developed a corpus from standard Arabic that contains 4000 sentences from news websites. It has been built using lexicon-based which achieved F-score of 76.5 and accuracy of 76.7%. Throughout the literature, it was shown that deep learning provided good results in Arabic NLP [12-16] and specifically in SA [18]. Alayba et al. [18] combined Convolutional Neural Network (CNN) and long short-term memory (LSTM). Authors used 4 datasets where the highest accuracy was achieved by using the dataset collected from twitter which was based on Ch5gram-level of 95.68%. Majazak [6] is an online Arabic sentiment analysis that used three datasets where the highest accuracy of 92% was achieved using deep learning over ArSAS dataset. Ombabi et al. [19] used CNN and LSTM then passed the output to SVM classifier which was used to make the final classification. The outstanding performance of the model with an accuracy of 90.75% was achieved. hULMonA proposed by [20] using Transformers Learning BERT. They used Hotel Arabic reviews dataset which consists of 93700 reviews and achieved an accuracy of 95.7%. Abu Kwaik et al. [21] presented SA corpus that was collected from twitter which contains 36000 tweets. They used two polarities which are: positive and negative. They used distant supervision approaches and achieved an accuracy of 86%. Various Arabic sentiment analysis classifiers were summarized in Table 1.

# 3 Dataset

Unfortunately, there is a lacking of Arabic sentiment analysis datasets compared to English. Since there is no standard Arabic corpus, we used the latest available dataset that was collected in April 2019. The dataset has been collected to provide an Arabic corpus for SA research community. It contains 58K Arabic tweets (47K training, 11K test) in multiple Arabic dialects classified into positive and negative labels It has been collected using Emojis lexicon. The reason of using tweets dataset is that it includes subjective data rather than descriptive ones [24].

# 4 Methodology

In this section, we present the used methodology which consists of two main phases (preprocessing and the model training), we utilized two classifiers for sentiments analysis: the first one is the naïve Bayes classifier while the second one is the combination between CNN and LSTM.

## 4.1 Data preprocessing

Preprocessing is an essential and critical task in NLP and text mining. It is a set of activities that process texts to make them usable for NLP and other tasks. Figure 1 shows the preprocessing for text mining. Because the collected texts may have special characters and numerical data that make noise, preprocessing data is important to reduce the size of data and to improve the efficiency of the system that will be used in. Several preprocessing steps have been used including: stop words and links removing, stemming and tokenization [25].

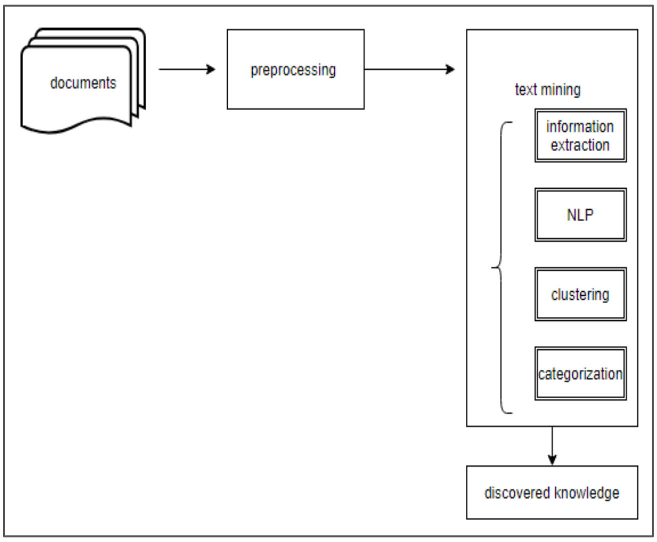


Figure 1: Preprocessing for text mining [26]

### 4.1.1 Data cleaning and removing

In this step we removed useless data in the tweets:

- Stop words: Stop words are the words that are frequently used and sometimes their use is meaningless in data analysis and data mining. Stop words are very common and highly used in Arabic like (في، من، إلى). Python NLTK library includes a set of commands that can be used to recognize and remove Arabic stop words.
- Links (URLs): URLs, as they are not natural language, they must be removed to reduce size of data and to reduce the noise.
- Hashtags: Because twitter is a framework that highly use hashtags by adding ‘#’ to link tweets to each other, it is already useful for predicting thus we only remove the hash ‘#’ sign from the tweet.

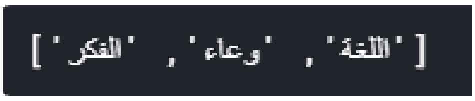
Table 1: Recently proposed sentiment analysis approaches.

| Ref | Year | Model | Dataset | Polarity | Results/Evaluation |
|---|---|---|---|---|---|
| [9] | 2015 | SVM | MIKA | Positive, negative or neutral | Accuracy=95% |
| [10] | 2016 | SVM<br>NN<br>Naïve Bayes<br>Decision Tree | Collected from Arabic reviews and comments from Facebook, Twitter, and YouTube. | Positive, negative or neutral | Accuracy=96.06% |
| [8] | 2016 | SentiWordNet facility | Arabic tweets | Positive, negative or neutral | Accuracy= 60.25% |
| [11] | 2017 | SVM | Tweets written in MSA and the Saudi Dialect | 1) positive and negative<br>positive, negative or neutral<br>positive, negative, neutral or mixed | Accuracy=69.9% |
| [12] | 2018 | SentiArabic | SentiTrain<br>SentiTest(testing)<br>PATB(testing) | Positive or Negative | F-score = 76.5<br>Accuracy = 76.7% |
| [18] | 2018 | CNN+LSTM | Main-AHS,<br>Sub-AHS,<br>Ar-Twitter,<br>ASTD | Positive or Negative | Accuracy = 95.68% |
| [6] | 2019 | “Mazajak” based on deep learning | SemEval,<br>ASTD,<br>ArSAS | 1) (SemEval) positive, negative or neutral<br>2) (ASTD) positive, negative or neutral<br>3) (ArSAS) positive, negative, neutral or mixed | Accuracy = 92% |
| [19] | 2020 | CNN+LSTM +SVM | Multi-domain sentiment corpus | Positive or Negative | Accuracy = 90.75% |
| [20] | 2019 | Transformers Learning BERT | HARD,<br>ASTD,<br>ArSenTD-Lev | (HARD) positive or negative<br>2) (ASTD) positive, negative or neutral | Accuracy = 95.7% |
| [21] | 2020 | Distant supervision approaches | ATSAD, LABR, ASTD, Shami-Senti | Positive or Negative | Accuracy = 86%. |
| [22] | 2021 | Arabic BERT tokenizer | ASTD, HARD, LABR, AJGT, ArSenTD-Lev | 1) (ASTD) positive, negative or neutral<br>2) (HARD) positive or negative<br>3) (AJGT) positive or negative | Accuracy = 96,11% |
| [23] | 2022 | deep LSTM, GRU, and CNN | Merges thirteen sets from free accessible sentiment analysis corpora | Positive, Negative or neutral | Accuracy = 95.08% |

- Diacritics: We removed diacritics because the tweets are written in dialects and MSA which rarely include diacritics.

### 4.1.2 Tokenization

This is the process of breaking a sequence of text into single words, phrases, symbols, which called tokens [25]. Figure 2 shows the sentence ”اللغة وعاء الفكر” after being tokenized. We used TweetTokenizer from NLTK python library.



```
['اللغة', 'وعاء', 'الفكر']
```

Figure 2: Tokenizer output

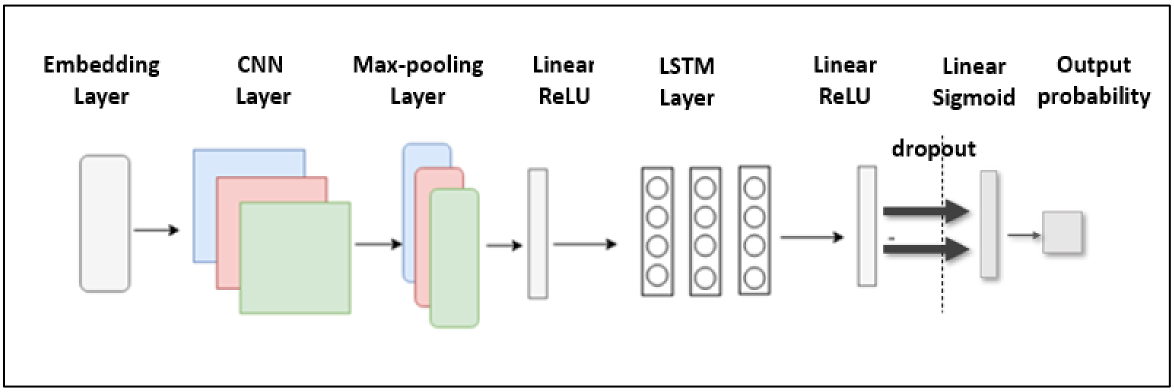


Figure 3: Overall CNN-LSTM model.

### 4.1.3 Stemming

Stemming is a conflating of the variant forms of the same word into one common representation, called the stem [25]. In Arabic natural language, the stem refers to the root of words so that the word دراسة /derasa/ which translated to (studying) and the word مدرسة /madrasa/ which is translated to (school) and the word يدرس / yadros/ which is translated to (studying) - all have the same root درس /darasa/ (study). Arabic stemming is a process that finds the lexical root of the words, by eliminating characters stuck to its root [27].

Information Science Research Institute's (ISRI) Arabic stemmer is built without using a root dictionary, ISRI stemmer provides better results than other stemmers on shorter queries [28]. In this research, we used ISRI stemmer in tweet cleaning. IRSI stemmer removes diacritics representing vowels, length three and length two prefixes, connector "و" (and ).

## 4.2 Training

In this section, we describe the two classifiers that we used for sentiment analysis:

### 4.2.1 Naïve Bayes

Naïve Bayes (NB) is a probabilistic algorithm in machine learning that is based on Bayes Theorem, used in classification tasks. Bayes theorem in mathematics is used for calculating the conditional probabilities. It assumes that the features are independent of each other conditionally and all the features have the same importance. NB is an algorithm that takes short time and short prediction time.

After cleaning the dataset, we train the model by finding the frequency of each word of tweets in each class (positive or negative), so that this helps us to compute the conditional probability according to the count of the words in each condition (class) which is represented as:

$$\mathrm{P(X|Y)} = \frac{P(Y|X)\times P(X)}{P(Y)} \qquad (1)$$

Where P(X|Y) is the probability of X (which represents the word) conditioned by Y (which represents the negative or positive class), and P(Y|X) is the probability of Y conditioned by X, P(X) is the probability of X which is the number of positive or negative examples in our work, and P(Y) is the probability of Y which refers to the number of all documents.

The result of equation (1) is used to find lambda (λ) in equation (2):

$$\lambda = \log\frac{P(pos)}{P(neg)} + \sum \log\frac{P(w|pos)}{P(w|neg)} \qquad (2)$$

The first component in equation (2) is called the log prior, that is that is the probability of the feature in the absence of any data, and the second component is the log likelihood, for a given word it registers how 'likely' is the data is. The final value of lambda is used to classify the class of the tweet if it is positive or negative.

### 4.2.2 Convolutional Neural Network & Long-Short Term Memory (CNN-LSTM)

Figure 3 shows the overall CNN-LSTM model. The proposed model consists of five layers. The first layer is the embedding layer which is used to represent the data using vectors. The output of the embedding layer is entered to the second layer which is the CNN layer. CNN layer is used to extract the useful features of the input data. These features are sequentially passed from convolutional neural network layer to the max pooling layer. After that, the results are passed to the ReLU. At this stage, LSTM layers take the output of the ReLU and passed it to the sigmoid function in order to build the text vector that is used in the prediction phase. More details about the model layers are explained in the following subsections:

**a. Embedding Layer**

Word embedding is the representation of words using vectors. This is one of the main steps that must be performed when using neural networks in natural language processing. The reason for this is the fact that the inputs for neural networks are numbers instead of text. In this research, we used Keras Embedding which is a supervised method that finds customized embeddings during model training.

Keras Embedding layer is being parameterized by specific weights. The weights are being updated during training the model based on the back-propagation algorithm. Therefore, the resultant embeddings of the words are conducted by the used loss function.

**b. Convolutional Neural Network (CNN)**

Word embedding are stacked into convolutional to extract the useful features in each region. The outputs of the CNN layer are the features that will be fed into max-pooling layer. The features at the CNN layer are generated using the following equation:

$$y = f(w \circ x_{n:n+\omega-1} + \mathrm{b}) \qquad (3)$$

Where f is ReLU function, o is the convolutional operator, w is the weight matrix, b is the bias vector and ω is the number of filters.

**c. Max-Pooling Layer**

The max-pooling layer is used to extract the most important features from the features that are generated using CNN layer. These most important features are fed into the LSTM layer. In max-pooling layer the max function is applied on the output of each filter from the CNN layer. The main reason for using CNN layer is to reduce the computations by eliminating the non-maximal values and to extract local dependency in each region of the filters.

**d. Long-Short Term Memory (LSTM)**

To capture long-term dependency in regions, we use the sequential layer which combines each region vector to text vector. After the LSTM memory cell sequentially traverses through all regions. this layer is followed by ReLU activation function and dropout Dense layers that takes the features that is bigger than the threshold to be passed to a Flatten Layer to be shaped and then passed to the final Activation Linear function.

**e. Linear Decoder**

The last layer is the output layer where the features are classified using sigmoid function which also called logistic function. Sigmoid function is used to predict the probability as an output, using the following formula:

$$\varphi(z) = \frac{1}{1+e^{-z}} \qquad (4)$$

the output of this function will be transformed between 0 and 1, the reason behind using sigmoid function is that we have two classes. In our model, we consider the values of sigmoid function that is less than 0.5 as a negative class. Otherwise, the value of sigmoid function will be used to represent the positive one.

**f. Hyper-Parameters Tuning**

Hyper-parameters are chosen depending on Trial and Error, many parameters have been set to minimize the Overfitting, the final chosen parameters are shown in Table 2.

The selection of parameters was based on many trails, some of which affected Overfitting of the model like the number of LSTM layers, and some of them affected the accuracy and results, these trails are shown in Table 4.

Table 2: CNN-LSTM model hyper-parameters

| **Parameter** | **Value** |
|---|---|
| **#LSTM cells** | 50 |
| **Recurrent dropout** | 0.2 |
| **Output dropout** | 0.4 |
| **#Filters in CNN** | 200 |
| **Filter size** | 3 |
| **Pooling size** | 2 |
| **Optimizer** | Adam |
| **Learning rate** | 0.0001 |
| **Activity regularizer** | 0.01 |

# 5 Experimental results

In this section we evaluate the Arabic Tweets using two approaches: Naïve Bayes against CNN-LSTM. Both classifiers have been evaluated using the most common measures, accuracy, recall (R) and precision (P), these measures are based on confusion matrix (CM), which is a matrix that used to describe a classification model on the test dataset. The elements of the matrix are named as follows: "True Positive" (TP) that is the number of true predictions in the positive class, "False Negative" (FN) which is the number of false predictions in the negative class, "False Positive" (FP) that is the number of false predictions of the positive class, and "True Negative" (TN) that refers to the number of true predictions in the negative class, this matrix can be written as

$$\mathrm{CM} = \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \qquad (5)$$

Classification accuracy measure is the number of correct predictions over the total number of predictions, as shown in equation (6).

$$Acc = \frac{TP + TN}{total} \qquad (6)$$

Classification recall measure (R) measures the positive class out of all positive predictions in the model, as shown in equation (7).

$$R = \frac{TP}{TP + FN} \qquad (7)$$

Classification precision measure (P) quantifies the number of positive predictions that belong to the positive class.

$$P = \frac{TP}{TP + FP} \qquad (8)$$

We used the three measures to evaluate the effectiveness of our models, the results are as follows:

## 5.1 Naïve Bayes results

NB model achieved an accuracy of 87.6%, recall of 87.6% and precision of 87.9%, as shown in Table 3:

Table 3: NB performance

| Criteria | Value |
|---|---|
| **Accuracy** | 0.876 |
| **Recall** | 0.876 |
| **Precision** | 0.879 |

The accuracy value of 87.6% is not considered satisfying in sentiment analysis, thus we used CNN-LSTM approach that is more efficient in SA.

## 5.2 CNN-LSTM results

CNN-LSTM model achieved an accuracy of 98%, precision of 93.6% ,and recall of 94.6% on Arabic Tweets testing data that contains 11k Tweets.

The accuracy and loss evaluation are shown in figures 4 and 5 respectively. The figures indicate that a small value of overfitting can be ignored.

This model gives better accuracy compared to the first one, as shown in Table 4. Sentiment analysis and most of NLP tasks fit better on algorithms that use the long dependencies and deep hidden layers.

Table 4: NB vs CNN-LSTM results

| | Naïve Bayes | CNN-LSTM |
|---|---|---|
| **Accuracy** | 87.6% | 98.0% |
| **Precision** | 87.9% | 93.6% |
| **Recall** | 87.6% | 94.6% |

All in all, we noticed that the accuracy of the proposed model (CNN-LSTM) outperformed those models presented in literature shown in Table 1. Actually, the case is expectedly repeated for the other 2 measures: the precision, and the recall. None of the measure used (accuracy, precision, and recall) reached 88% for the NB, while none of these same measures went below 93.6% for CNN-LSTM; and this a clear evidence that the CNN-LSTM performed very well.

# 6 Conclusion

In this paper, we proposed two main approaches for classifying Arabic sentiments of Twitter Arabic corpus. The corpus contains many Arabic dialects and Modern Standard Arabic tweets. The first approach was naïve Bayes classifier which achieved an accuracy of 87.6%. On the other hand, the second approach was a combination of CNN and LSTM models that achieved an accuracy of 98%. Several preprocessing steps had been conducted on the dataset such as stemming and tokenization of Arabic tweets after cleaning them by removing useless data. The experimental results showed that CNN-LSTM classifier is better than Naïve Bayes classifier in term of accuracy.

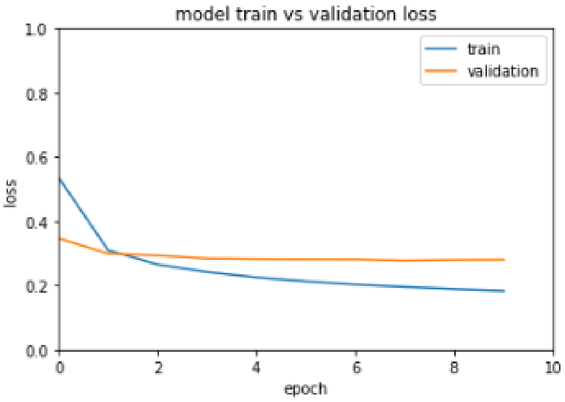


Figure 4: Train and validation loss.

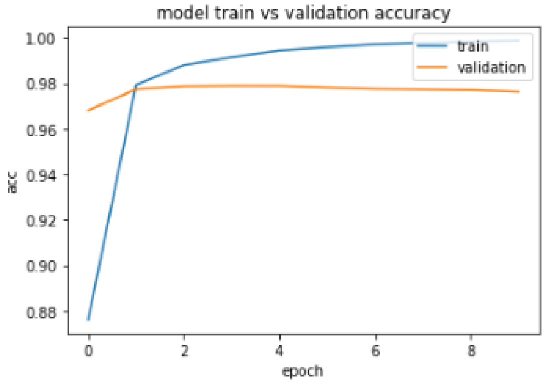


Figure 5: Train vs validation accuracy.

# 7 Future work

The future orientation is to extend our work by collecting more data and use transformers for analyzing and predicting sentiments. We aim to increase the number of classes to analyze the emotions and opinions more specifically which a field called emotional intelligence (EI). We hope to provide more works and put all of our energy for improve Arabic NLP.

Table 4: Hyper-parameters selection.

| # of layers in CNN | # of layers in LSTM | Dropout | Regularizer | Learning rate | Accuracy | Precision | Recall | Overfitting? |
|---|---|---|---|---|---|---|---|---|
| 100 | 50 | 0.4 | 0.010 | 0.0010 | 97.8 | 92.5 | 96.0 | No |
| 100 | 50 | 0.4 | 0.001 | 0.0001 | 97.7 | 92.6 | 94.6 | Yes |
| 100 | 50 | 0.4 | 0.010 | 0.0001 | 98.1 | 93.9 | 94.1 | No |
| 100 | 50 | 0.5 | 0.010 | 0.0001 | 97.9 | 94.0 | 94.1 | No |
| 300 | 50 | 0.4 | 0.010 | 0.0001 | 98.0 | 94.6 | 93.1 | Yes |
| 200 | 50 | 0.4 | 0.010 | 0.0001 | 98.0 | 93.6 | 94.6 | No |
| 200 | 100 | 0.4 | 0.010 | 0.0001 | 98.1 | 92.5 | 95.7 | Yes |
| 100 | 100 | 0.4 | 0.010 | 0.0001 | 98.0 | 93.1 | 95.0 | Yes |

# References

[1] Vinodhini, G., & Chandrasekaran, R. (2012). Sentiment Analysis and Opinion Mining : A Survey, International Journal of Advanced Research in Computer Science and Software Engineering, 2(6), 283–292.

[2] W. Etaiwi, D. Suleiman, and A. Awajan, "Deep Learning Based Techniques for Sentiment Analysis: A Survey," IJCAI, vol. 45, no. 7, Dec. 2021, https://doi.org/10.31449/inf.v45i7.3674.

[3] Hassonah, M. A., Al-Sayyed, R., Rodan, A., Ala'M, A. Z., Aljarah, I., & Faris, H. (2020). An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter. Knowledge-Based Systems, 192, 105353. https://doi.org/10.1016/j.knosys.2019.105353.

[4] Hajrizi, R., & Nuçi, K. P. (2020). *Aspect-Based Sentiment Analysis in Education Domain*. http://arxiv.org/abs/2010.01429.

[5] Nurfauzi, Y., Suyanto, S., Sukidjo, S., Munsyi, M., & Ahdhianto, E. (2020). The role of business center using sentiment analysis to foster entrepreneurial spirit in vocational high school. *Universal Journal of Educational Research*, *8*(11), 5151–5157. https://doi.org/10.13189/ujer.2020.081114

[6] Abu Farha, I., & Magdy, W. (2019). *Mazajak: An Online Arabic Sentiment Analyser*. 192–198. https://doi.org/10.18653/v1/w19-4621.

[7] Ghallab, A., Mohsen, A., & Ali, Y. (2020). Arabic Sentiment Analysis: A Systematic Literature Review. *Applied Computational Intelligence and Soft Computing*, *2020*. https://doi.org/10.1155/2020/7403128.

[8] Al-Horaibi, L., & Khan, M. B. (2016). Sentiment Analysis of Arabic Tweets Using Semantic Resources. *International Journal of Computing and Information Sciences*, *12*(2), 149–158. https://doi.org/10.21700/ijcis.2016.118.

[9] Ibrahim, H. S., Abdou, S. M., & Gheith, M. (2015). MIKA: A tagged corpus for modern standard Arabic and colloquial sentiment analysis. *2015 IEEE 2nd International Conference on Recent Trends in Information Systems, ReTIS 2015 - Proceedings*, *4*(2), 353–358. https://doi.org/10.1109/ReTIS.2015.7232904

[10] Hammad, Mustafa & Al-awadi, Mouhammd. (2016). Sentiment Analysis for Arabic Reviews in Social Networks Using Machine Learning. https://doi.org/10.1007/978-3-319-32467-8_13.

[11] Al-Twairesh, N., Al-Khalifa, H., Alsalman, A. M., & Al-Ohali, Y. (2018). Sentiment analysis of Arabic tweets: Feature engineering and a hybrid approach. *ArXiv*.

[12] Eskander, R. (2019). SentiArabic: A sentiment analyzer for standard Arabic. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, 1215–1219.

[13] D. Suleiman, W. Etaiwi, and A. Awajan, "Recurrent Neural Network Techniques: Emphasis on Use in Neural Machine Translation," IJCAI, vol. 45, no. 7, Dec. 2021, https://doi.org/10.31449/inf.v45i7.3743.

[14] D Suleiman, A Awajan and W Etaiwi, Arabic Text Keywords Extraction using Word2vec, The 2nd International Conference on new Trends in Computing Sciences (ICTCS'19), Amman, 2019. https://doi.org/10.1109/ICTCS.2019.8923034.

[15] D. Suleiman and A. Awajan, "Multilayer encoder and single-layer decoder for abstractive Arabic text summarization," KnowledgeBased Systems, vol. 237, p. 107791, Feb. 2022, https://doi.org/10.1016/j.knosys.2021.107791.

[16] D. Suleiman and A. Awajan, "DEEP LEARNING BASED ABSTRACTIVE ARABIC TEXT SUMMARIZATION USING TWO LAYERS ENCODER AND ONE LAYER DECODER," Journal of Theoretical and Applied Information Technology, Vol. 98, no. 16, p. 12, 2020.

[17] D Suleiman and A Awajan, Using Part of Speech Tagging for Improving Word2vec Model, The 2nd International Conference on new Trends in Computing Sciences (ICTCS'19), Amman, 2019

[18] Alayba, A. M., Palade, V., England, M., & Iqbal, R. (2018). A combined CNN and LSTM model for Arabic sentiment analysis. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11015 LNCS, 179–191. https://doi.org/10.1007/978-3-319-99740-7_12

[19] Ombabi, A. H., Ouarda, W., & Alimi, A. M. (2020). Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks. *Social Network Analysis and Mining*, *10*(1), 1–13. https://doi.org/10.1007/s13278-020-00668-1

[20] ElJundi, O., Antoun, W., El Droubi, N., Hajj, H., El-Hajj, W., & Shaban, K. (2019). hULMonA: The Universal Language Model in Arabic. 1, 68–77. https://doi.org/10.18653/v1/w19-4608

[21] Abu Kwaik, K., Chatzikyriakidis, S., Dobnik, S., Saad, M., & Johansson, R. (2020). An Arabic Tweets Sentiment Analysis Dataset (ATSAD) using Distant Supervision and Self Training. *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, *March*, 1–8. https://www.aclweb.org/anthology/2020.osact-1.1

[22] Chouikhi, Hasna & Chniter, Hamza & Jarray, Fethi. (2021). Arabic Sentiment Analysis Using BERT

Model. https://doi.org/10.1007/978-3-030-88113-9_50.

[23] Omara, Eslam & Mosa, Mervat & Ismail, Nabil. (2022). Applying Recurrent Networks For Arabic Sentiment Analysis. Menoufia Journal of Electronic Engineering Research. https://doi.org/31.21-28. 10.21608/mjeer.2022.218776.

[24] Bessou, S., & Aberkane, R. (2019). Subjective Sentiment Analysis for Arabic Newswire Comments. *ArXiv*, *17*(5). https://doi.org/10.6025/jdim/2019/17/5/289-295.

[25] Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., Nithya, M., Kannan, S., & Gurusamy, V. (2015). Preprocessing Techniques for Text Mining. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16.

[26] S, V., & R, J. (2016). Text Mining: open Source Tokenization Tools – An Analysis. *Advanced Computational Intelligence: An International Journal (ACII)*, *3*(1), 37–47. https://doi.org/10.5121/acii.2016.3104.

[27] A. Otair, M. (2013). Comparative Analysis of Arabic Stemming Algorithms. *International Journal of Managing Information Technology*, *5*(2), 1–12. https://doi.org/10.5121/ijmit.2013.5201.

[28] Taghva, K., Elkhoury, R., & Coombs, J. (2005). Arabic stemming without a root dictionary. *International Conference on Information Technology: Coding and Computing, ITCC*, *1*, 152–157. https://doi.org/10.1109/itcc.2005.90.