Ingénierie des Systèmes d'Information
Vol. 28, No.1, February, 2023, pp. 67-75
Journal homepage: http://iieta.org/journals/isi

A Deep Neural Network Optimized by a Genetic Algorithm to Improve Arabic Sentiment Classification

Omar Al-Harbi[1*], Ahmed Hamed[1], Malek Alzoubi[2]

[1] Computer Dept., Applied College, Jazan University, Jazan 45142, Saudi Arabia
[2] Information Technology & Security Dept., College of Computer Science & IT, Jazan University, Jazan 45142, Saudi Arabia

Corresponding Author Email: oalharbi@Jazanu.edu.sa

https://doi.org/10.18280/isi.280107

Received: 17 November 2022
Accepted: 30 January 2023

Keywords:
sentiment classification, Arabic sentiment analysis, deep learning, genetic algorithm

ABSTRACT

Deep learning has improved the state-of-the-art in sentiment analysis for various languages, including Arabic. One aspect that can affect the performance of deep learning-based sentiment classification is the optimization method used for training the neural network. The conventional optimization method is carried out by a backpropagation (BP) algorithm that relies on gradient descent to find the minimum of a cost function. However, BP has the tendency to converge into local minima instead of global minima since neural networks generate complex error surfaces for even simple problems. In this study, for the purpose of improving the Arabic sentiment classification, we propose to use a genetic algorithm (GA) to train a deep neural network (DNN). GA is a meta-heuristic optimization algorithm inspired by the theory of natural evolution. The algorithm is expected to improve the classifier's performance due to its capability to reach optimal or near-optimal solutions. The proposed method uses Arabic sentiment lexicons to extract various features considering different aspects for text representation. The effectiveness of the proposed method is evaluated by analyzing its performance, versus a DNN trained with BP algorithm. The experimental results show that the proposed method can present better F1-measure of 90.7% for Arabic sentiment classification than traditional BP-based DNN.

1. INTRODUCTION

Sentiment analysis refers to the process of automatically identifying opinions expressed in texts about different objects of interest and classifying their sentiment polarization (positive or negative) [1]. With the massive growth of user-generated content on Web and social media, the automatic classification for opinions has become a crucial element that helps to make decisions in a variety of domains such as politics, commerce, education, and health. Positive or negative opinions can be in different forms of textual information such as tweets, comments, articles, or reviews. Generally, the sentiment classification is performed at three levels of granularity: document, sentence, and aspect.

Research on sentiment analysis has achieved considerable progress in English language. However, Arabic sentiment analysis research has not developed significantly despite the increasing expansion of Arabic language usage on the Internet [2]. That can be attributed to the complex linguistic nature and diverse dialects of the Arabic language [3, 4]. Additionally, the lack of Arabic linguistic resources and shortage of natural language processing (NLP) tools [5]. Early studies in the field of sentiment analysis presented many attempts to address various tasks based on two types of approaches, namely, semantic orientation and machine learning. The former leverages on existing predefined language lexicons that contain polarized and scored terms such as SentiWordNet [6] and WordNet [7]. While the latter uses learning algorithms to learn from either labeled or unlabeled data to generate models that classify, or cluster a given input data. Recently, machine learning-based techniques have been widely used to address the sentiment analysis task, due to their efficiency and competence which resulted in remarkable success [8]. In machine learning, sentiment analysis can be seen as a polarity classification problem that can be addressed by adapting different learning models including deep learning models.

Deep learning is a machine learning model, which is based on artificial neural network. In the last few years, deep learning has attained a keen interest from researchers in sentiment analysis because of its ability to handle the complex problems efficiently [9]. Deep learning technique along with the availability of large datasets has made essential evolution in the sentiment analysis field. Different deep learning structures like deep neural network (DNN), recurrent neural network (RNN), and convolutional neural network (CNN) have yielded significant results and outperformed the traditional machine learning algorithms. With respect to the Arabic language, the literature shows that little research on deep learning-based sentiment analysis has been introduced, such as the studies [10-13]. In this study, we mainly focus on DNN as a sentiment classifier for the Arabic language.

The typical architecture of DNN is composed of multiple hidden layers and interconnected processing nodes that enable it to discover semantic representations of texts automatically from data [14]. In DNN, the learning process is performed by a feed-forward algorithm through which the input data is fed to processing nodes in a layer to calculate activation levels and then pass the outcome to other layers up to the output layer. Another training algorithm called back-propagation (BP) is required for the optimization process, which is responsible for minimizing the cost function by finding an optimal set of weights. BP is the most commonly used optimization algorithm

for training neural networks [15]. The key to BP is using a gradient descent algorithm for minimizing the observed error with respect to the weights for given data inputs by moving backwards through the network. However, using BP may not give the best performance since neural networks generate complex error surfaces for even simple problems with the presence of multiple local minima and global minima and the tendency to converging into local minima instead of global minima [16-18].

This problem has induced many researchers to explore other methods for training the networks. One method is the use of meta-heuristic algorithms for training the networks include but are not limited to the genetic algorithm (GA). GA is a search-based optimization algorithm inspired by the natural selection mechanism for selecting a set of high-quality solution candidates to a problem [19]. This algorithm has the ability to reliably solve problems that are complex, continuous, discrete, and non-differentiable [18]. Consequently, it can always reach the near-optimum or global minima [20]. GA has recently attracted much attention from researchers, and it has been successfully applied in diverse domains such as image classification [21], economy [22], commerce [23], and healthcare [24].

Based on what mentioned above, it is expected that integrating GA and DNN would avoid the limitations and combine the advantages associated with each of these techniques and results in better performance for sentiment classification. Therefore, in this paper, to improve Arabic sentiment classification on reviews, the GA is proposed for optimizing the DNN. To the best of our knowledge, using GA with the neural network, in general, for Arabic sentiment analysis, was limited only for deciding the optimal subset features or network configurations. However, we extend the usage of GA to train a deep network by searching the optimal or near-optimal connection weights between the nodes instead of BP algorithm. With GA, we can formulate the training process as the evolution of all weights generated in the DNN by using genetic operators such as selection, crossover, and mutation through a number of generations.

To extract important hidden information from reviews, which may lead to better classification, we consider different aspects rather than seeing a review as a bag of words. Thus, for text representation and feature extraction, we define an extensive set of 20 features that can be grouped into three categories, namely, sentiment-based, linguistic-based, and structural features. To this end, we employ sentiment-based lexicons, and predefined lists of modifiers, stop-words, negation words, compound words, emoticons, etc. To evaluate our work, we compare the performance of the proposed method with that of typical DNN with BP training algorithm that uses a gradient descent optimizer.

The remaining of this paper is organized into four sections: Section 2 discusses the related work; Section 3 presents the methodology; Section 4 describes the evaluation method and summarizes the results; Section 5 presents the conclusion of this work.

2. RELATED WORK

In the literature on artificial neural network, usage of GA has been found in different ways; network architecture optimization, connection weight optimization, and feature selection. The problem of deciding the optimum configurations of a network architecture has been studied in sentiment analysis for different languages. For instance, the authors [25] compared particle swarm optimization (PSO) and GA with differential evolution (DE) algorithm in searching for the optimal configurations of CNN architecture to classify Arabic texts. The network hyper-parameters include filter sizes, number of filters per convolution filter size, number of neurons, initialization mode, and dropout rate. A word embedding matrix with two dimensions was used to represent features. As reported, the DE algorithm yielded the highest average accuracies and the shortest computation time compared to the other algorithms on five different datasets. Similarly, Ishaq et al. [26] adopted GA to optimize the architecture of CNN for aspect-based sentiment analysis of English texts. Their proposed method outperformed other machine learning algorithms such as SVM, maximum entropy, random forest, stabilized discriminant analysis, decision tree and generalized linear model in terms of accuracy, precision, recall, and F-measure.

GA also has been shown to perform efficiently in searching for optimal initial weights of a network. Yin et al. [27] employed GA to optimize the initial weights and thresholds of BP-based neural network networks for public opinion prediction on Chinese texts. They built a network of one hidden layer to compute the output that will be optimized based on the BP algorithm. The Metropolis acceptance criterion was implemented with the aim of improving the local searching ability of GA. Likewise, Ye et al. [28] combined GA and simulated annealing algorithm to optimize the initial weights to be trained by a BP-based neural network of one hidden layer to predict opinion trends in Chinese texts.

On the other hand, Alboaneen et al. [29] implemented three meta-heuristic algorithms for optimizing the weights of a multi-layer perceptron network instead of the BP algorithm. The meta-heuristic algorithms include GA, glowworm swarm optimization (GSO), and biogeography-based optimization (BBO). The network with only one hidden layer was built for classifying the sentiment of English tweets. They employed mutual information (MI) for selecting the optimal subset of features. Experimental results showed superiority in performance for GSO over the other algorithms. For the same purpose of weights optimization, GA also has been integrated with different machine learning classifiers such as the work [30], where the authors combined GA with PSO and decision tree algorithm to classify the sentiment of English tweets. The model they suggest showed better performance versus SVM and K-nearest neighbor (K-NN). Another hybrid model for the same language is proposed [31] based on GA and Naïve Bayes (NB) to classify movie reviews. They used an arcing classifier to combine the two classification algorithms. The proposed hybrid NB-GA method is shown to be superior to individual approaches.

Furthermore, the merits of using GA have been investigated in semantic orientation-based sentiment analysis. As an instance, the work [32] proposed word and document co-clustering for sentiment analysis based on GA. The authors used co-clustering to reduce the search space and group related sentiment words into clusters. Then, they considered the problem of determining the weight of the sentiment words for each cluster as an optimization problem addressed by a GA. For classification, they built a decision list to compute the sentiment based on selected weights. The experiments on the dataset of Russian language showed that the effectiveness of the proposed method is higher than other classifiers such as SVM.

GA also has been found in the literature as feature selection method to improve the performance of various machine learning classifiers. For example, in the work [33], GA was used to select the optimal subset features to improve sentiment classification of SVM and NB on Arabic texts. Abbasi et al. [34] also proposed entropy weighted GA for feature selection with a SVM classifier on texts of multiple languages, including Arabic. They concluded that using GA resulted in a considerable improvement in accuracy. For the same purpose, Zhu et al. [35] evaluated the performance of a conditional random forest (CRF) model for classifying sentiments into positive or negative with GA used for selecting the optimal subset of features. On the other hand, the work [36] used GA for the task of features subsets generation with a fitness function based on correlation criteria. The authors used SVM for the classification task on Arabic texts.

Heikal et al. [37] use an ensemble model of CNN and LSTM to predict the sentiment of Arabic tweets for the sentence level. The model is trained on top pre-trained word vectors developed [38]. They evaluated the model on ASTD dataset [39] and achieved an F1-score of around 64% and an accuracy of around 65%. Another work applies CNN and LSTM to sentiment analysis of Arabic tweets described [12]. They designed a system to identify the sentiment's class and intensity as a score between 0 and 1. The authors used word and document embedding vectors to represent the tweets. They translated the tweets into English to benefit from the available preprocessing tools. As they highlighted, the step of the translation led to degrading the overall performance. Although the system includes some preprocessing steps, it excludes some other important normalizing processes such as removing diacritics, punctuations and repeating characters.

Based on the literature review, it was found that the usage of GA in Arabic sentiment analysis is limited to selecting optimal features subset and optimize network architecture. Differently, this work proposes a method the basis of which is to utilize GA instead of BP to train the DNN through optimizing the connection weights. Applying GA to weights optimizing in DNN is simulated by an evolutionary process in which genetic operators such as selection, crossover, and mutation have to be developed. The proposed method also uses Arabic sentiment lexicon and linguistic knowledge to extract features that efficiently represent the reviews and avoid the problem of high dimensional representation that is computationally expensive.

3. PROPOSED METHOD

The proposed method for Arabic sentiment classification is composed of three steps, namely, data preprocessing, feature extraction, and sentiment classification based on DNN and GA.

3.1 Data preprocessing

The preprocessing step consists of data cleansing and normalization. The process of data cleansing includes removing misspellings, repeated letters, diacritics, double spaces, symbols, numerals, English words, and elongation. Afterwards, a normalization process is applied to particular letters, for example, the letters (أ, إ, آ) were converted to (ا), the letters (ى, ئ) were converted to (ي), the letter (ة) was converted to (ه), and finally the letter (ؤ) was converted to (و).

3.2 Feature extraction

Feature extraction is a fundamental process prior to applying a learning classification algorithm. In this step, the data is transformed into dimensions of features representing the information embedded into the reviews. The efficacy of identifying features plays an essential role in achieving high classification performance. In this work, we extracted 20 features that can be grouped into three different sets, namely, sentiment-based features, linguistic-based features, and structural-based features. For assessing the proposed features, we performed a process to measure the correlation, which showed that features are highly correlated with the polarity class, yet uncorrelated to each other. Table 1 presents the details of the features extracted for the purpose of this work. For F1-F7 in sentiment-based features, we adopted the publicly available resources introduced [40] which contain 3400 labeled sentimental words and 580 sentiment-carrying compound phrases. Negation is also considered in this study; we employed the rule-based algorithm presented [41] to extract F10-F12 in linguistic-based features. The algorithm detects negation words such as (مش، مو، لا) and then tags the opinionated words that might be affected within a predefined window length of words. The rules were crafted based on observing many cases of negation, simple linguistic knowledge, and sentiment lexicon. Furthermore, we manually built lists for the remaining features; including emoticons, stop words, modifiers, and shifters.

Table 1. Details of the proposed features

Group name	Feature	Description	Representation	Example
Sentiment-based Features	F1	Frequency of subjective words normalized by F19	real number	زاكي، بتخزي (Delicious, awful)
	F2	Frequency of positive words normalized by F1	real number	زاكي Delicious
	F3	Frequency of negative words normalized by F1	real number	بتخزي Awful
	F4	Frequency of neutral words normalized by F19	real number	الأسعار Prices
	F5	Presence of positive compound phrase	Binary(0, 1)	سعرو فيه It is worth the price
	F6	Presence of negative compound phrases	Binary(0, 1)	قول وفعل Saying and doing بطلعو روحك They get your soul out
	F7	Polarity of last sentence	Multi (positive, negative,	منك لله Leave it up to God -

			neutral)	
	F8	Presence of positive emoticon	Binary(0, 1)	☺
	F9	Presence of negative emoticon	Binary(0, 1)	☹
Linguistic-based Features	F10	Frequency of negation words normalized by F19	real number	لا، مش، مو No, not
	F11	Frequency of negated positive words normalized by F2	real number	مش حلو It is not cool
	F12	Frequency of negated negative words normalized by F3	real number	مو غلط It is not shameful
	F13	Frequency of shifter normalized by F19	Integer	لكن، لاكن، بس But
	F14	Frequency of modifier for positive words normalized by F2	Binary(0, 1)	كثيرا، قليلا، للغاية، اكتير، بالمره، شوي
	F15	Frequency of modifier for negative words normalized by F3	Binary(0, 1)	Much, slightly, extremely, at all, a bit
	F16	Frequency of stop-words normalized by F19	real number	أنت، إذا You, if
	F17	Frequency of question normalized by F20	Integer	؟
	F18	Frequency of exclamation normalized by F20	Integer	!
Structural Features	F19	Length of review	Integer	-
	F20	Number of sentences per a review	Integer	-

3.3 Data preprocessing

This section introduces the methods that are used for the sentiment classification on Arabic reviews. The methods include the typical DNN and proposed GA-based training for DNN.

3.3.1 DNN

DNN is an extension of ANN, which is a biologically inspired machine learning model that imitates the functioning of the human brain for learning from observational data. The typical architecture of ANN consists of three major layers: an input layer, a hidden layer, and an output layer. On the other hand, the conventional structure of DNN differs in the number of hidden layers used for the learning process, as shown in Figure 1.

The number of hidden layers L determines how deep the structure of DNN. Each layer K^{th} is comprised of multiple nodes N that are fully connected with other nodes in the next layer. The connection between i^{th} node in input layer and j^{th} node in the first hidden layer is represented by w_{ij}^{k}. Thus, for input variables x_i in a training example, the first node j_1 in the first layer k can be computed as:

$$a_{j_1}^{1} = f\left(\sum_{i=1}^{m} w_{i,j_1}^{1}.x_i + b_j^{1}\right) \quad (1)$$

While the nodes j_k in the subsequent hidden layers k, where $k \geq 2$ can be calculated as:

$$a_{j_k}^{k} = f\left(\sum_{j_{k-1}}^{N} w_{j_{k-1},j_k}^{k}.a_{j_{k-1}}^{k-1} + b_{j_k}^{k}\right) \quad (2)$$

where, b denotes the bias term and $f(x)$ represents an activation function. In this work, we investigated two activation functions: sigmoid and Relu. Then, an optimization process is performed to find the optimal set of connection weights. As mentioned earlier, BP is the most commonly used optimization algorithm for training artificial neural networks. BP begins with quantifying the error output of the network through a cost function. For illustration purpose, let us assume that a Quadratic function was used which can be written as:

$$C = \frac{1}{2m}\sum_{x}(y(x) - a(x))^2 \quad (3)$$

where, m is the total number of training examples, $y(x)$ is the desired output, and $a(x)$ is the actual output. The aim of the optimization process is to minimize the error output between $a(x)$ and $y(x)$, where the connecting weights are adjusted iteratively until the error is the least. This is usually achieved by a gradient descent algorithm that starts from the last layer backward to the first layer. The key of using gradient descent algorithm is to compute the partial derivative of the cost function with respect to the weight and bias.

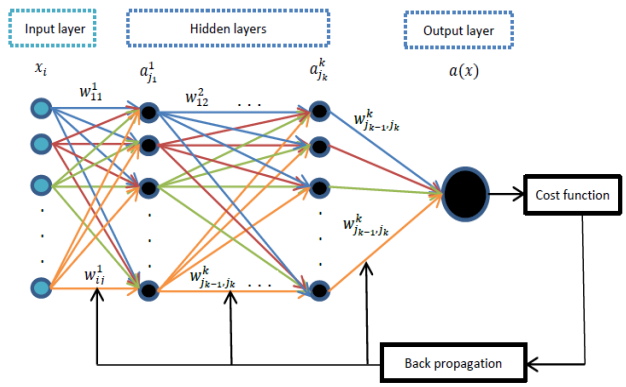

Figure 1. The structure of DNN

In this study, the BP training method is used only for the purposes of comparison with the proposed method in which GA was employed as the optimization method. The structural attributes (or hyper-parameters) of the DNN are determined, including the number of hidden layers and the number of nodes and activation functions for each layer. The choice of those hyper-parameters is important as they affect the performance of DNN by avoiding problems such as over-fitting and under-

fitting. There is no rule-of-thumb for choosing the hyper-parameters as it is a problem-dependent; therefore, they were selected based on a trial-and-error approach.

3.3.2 GA-based DNN

To change the default, adjust the template as follows.GA is a global optimization technique that is inspired by Darwin's theory of natural evolution. The algorithm mimics the process of natural selection where the solutions are evolved across a number of generations to reproduce the best solutions. Each solution called an individual that is represented as a chromosome, and a group of solutions called a population. A chromosome composed of genes that can be represented in different codes. In the beginning, GA works on the randomly initialized population of chromosomes where each one is evaluated using a fitness function which determines its competency. Accordingly, the chromosomes with the highest fitness values will be selected to continue through three iterative genetic operators; selection, crossover, and mutation in order to produce offspring with higher quality. For the purpose of this study, we formulate the DNN training process as the evolution of all connection weights w_{ij}^{k} using the genetic operators and without computing gradient information. The architecture of the GA-based DNN is illustrated in Figure 2.

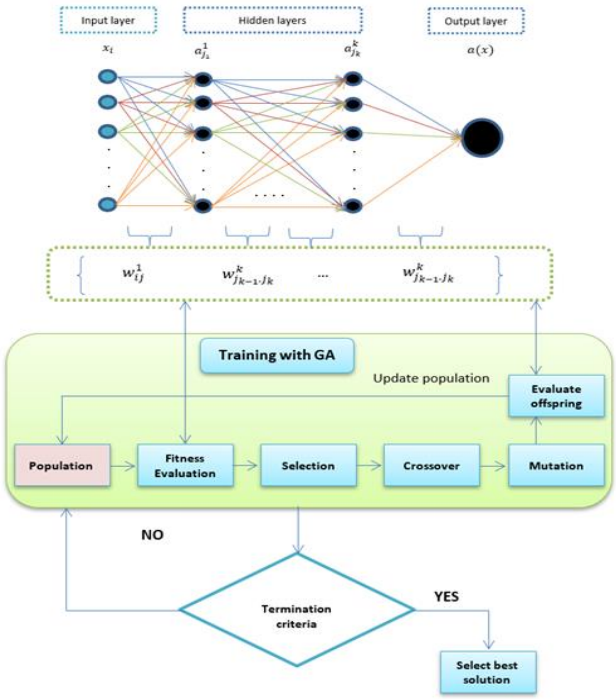

Figure 2. The architecture of GA-based DNN

In general, the first step in GA is randomly generating a population *P* of chromosomes *c*, which evolve across a number of generations. All the randomly initialized connection weights and biases in all layers will be considered as one chromosome $c_1 = \left(w_{ij}^{1}, w_{j_{k-1},j_k}^{k}, b_{j_k}^{k}\right)$ and encoded as real values. To decide which chromosomes have better survival chance, the DNN accepts the population of chromosomes, and through a fitness function, they are evaluated in terms of their accuracy of predicting the class label. The accuracy function measures the ratio between the correctly classified samples and the total number of samples based on the DNN prediction results. Then, three genetic operators will be iteratively applied until the termination criteria are met, as follows:

(1) Selection: After the fitness values of the chromosomes have been calculated, the best chromosomes with the highest accuracy values (they are known as parents) are selected into what so-called mating pool. The intuition is that choosing the chromosomes with high-quality genes has a bigger chance to survive, and it is expected to reproduce new chromosomes (offspring) with better quality than the parents.

(2) Crossover: With respect to crossover, the selected parents into the mating pool will undergo an operation in which some genes from couple parents are exchanged to produce offspring that have the parent's properties. There are several methods to apply the crossover operator. In this work, we adopted the single-point method in which a point that split genes into two halves is picked, and then the genes to the right and left of the point are exchanged between the two parent chromosomes resulting in two offspring.

(3) Mutation After all the preceding steps, the loop comes to its end with a mutation process. This operator is responsible for generating genetic diversity that avoids local minima problem. The output offspring of crossover operation will be mutated through stochastically selecting a percentage of genes from each offspring and altering their values. The output of this operator is a new offspring (chromosome) with new properties. The fitness value of the offspring is evaluated using DNN, to be a replacement for its parent chromosomes in the population if its fitness value is better; otherwise, the parent chromosome is kept. The mutation has several forms based on the chromosome representation. Since the real value encoding is used in this work, we used a uniform random value selected between [-1, 1] to replace the value of the chosen gene.

The genetic operation explained above forms one generation or iteration of the GA-based training. Then the evolving process iteratively continues until the predetermined generations are finished, or the optimal or near-optimal solution has been found. Then the best solution (optimal connection weights) is returned to the DNN to be used for sentiment classification. Finally, to ensure optimization with optimal solutions, the hyper-parameters of the GA must be assigned appropriately. In this work, we used the trail-and-error approach to choose the values of those hyper-parameters, which include population size, number of generations, number of parents to mate, and mutation rate.

4. EXPERIMENTS AND RESULTS

This section presents the experiment conducted to evaluate the performance of the proposed model for Arabic sentiment classification. Additionally, we introduce the dataset, hyper-parameter settings, and evaluation metrics. It is worth mentioning that the experiments were implemented using Python 3.6.5. The genetic algorithm and neural network were developed based on NumPy library. The BP-based DNN was built using Keras library. More details about the experiments are provided in the following subsections.

4.1 Dataset

For the purpose of this work, we used a publicly available dataset introduced [42] for Jordanian dialect. The dataset is annotated on the document level, and it considers only two polarity classes, which are positive and negative. To balance the dataset, we randomly selected 2000 reviews, of which 1000 were positive, and 1000 were negative. The data consists of

MSA and colloquial Jordanian reviews about various domains (restaurants, shopping, fashion, education, entertainment, hotels, motors, and tourism).

4.2 Evaluation metrics

For classification performance evaluation, we initially split the dataset into training, validation and testing sets by percentage 80%, 10%, and 10%, respectively. Then, the performance is quantified using the following evaluation metrics: Precision (P), Recall (R), and F1-score (F1); see Eqns. (2), (3) and (4). The precision calculates the accuracy of the classifier in regard to the specific predicted class. The recall shows the percentage of the correct predicted classes among the actual class in the data. The F1-score represents an overall measure of a model's accuracy that calculates the weighted average of precision and recall.

$$P = \frac{TP}{TP + FP} \tag{4}$$

$$R = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{6}$$

where, *TP* indicates a true positive which means the number of the inputs in data test that have been classified as positive when they are really belong to the positive class. *TN* indicates a true negative which means the number of the inputs in data test that have been classified as negative when they are really belong to the negative class. *FP* indicate a false positive which means the number of the inputs in data test that have been classified as positive when they are really belong to the negative class. *FN* indicates a false negative which means the number of the inputs in data test that have been classified as negative when they are really belong to the positive class.

4.3 Hyper-parameters configuration

This section illustrates the hyper-parameters configured experimentally for DNN and GA. The process of selecting the optimal hyper-parameters is a challenging task, and it varies based on the characteristics of the research problem. To this end, we performed several trials to choose the hyper-parameters with which the classifiers yielded the best performance results. It is worth mentioning that all the data points have been normalized before being passed to the network. According to the results obtained experimentally, a neural network was constructed with five layers of size 128, 64, 32, 18, and 18 nodes, respectively. The dense hidden layers are trained using a Relu function. Then, the weights are passed to an output layer with a sigmoid function to give the final classification probability.

In addition, to ensure optimization with superior selection results, the hyper-parameters of the GA need to be tuned appropriately. Similar to DNN, we used the trial-and-error approach to choose the values of those hyper-parameters, which include population size, number of generations, number of parents to mate, and mutation percentage. Thus, knowing that the bigger size of the population may lead to finding the optimal solution, we ended up with assigning 200 to population size. The number of parents determines how many chromosomes with the highest accuracy will be selected parents into the mating pool. This work takes two parents to undergo the crossover operation. Since the changes in mutation are random, the percentage should be small to avoid instability. For this work, the mutation percentage of 0.2 was set. Finally, the number of generations was set to 400.

4.4 Results

In this section, we present the experimental results of applying the proposed method to Arabic sentiment classification. The proposed method was evaluated through a comparison with BP-based DNN based on precision, recall, and F-measure over the same dataset. It is necessary to mention that BP uses the gradient descent algorithm for optimization with a learning rate of 0.001, and binary cross-entropy function as a cost function. For the network structure, it was constructed with the same topology used when GA is applied. Furthermore, we carried out an experiment that compares the proposed features with Unigram model when BP is applied to DNN with using the same hyper-parameters to evaluate the effectiveness of the proposed features.

Figure 3 presents the value of accuracy during the training process based on GA, where the accuracy was calculated using the fitness function through each generation based on the training set. It can be seen that evolution has converged after 150 with an accuracy value of 88.7%. Whereas, when BP was used for training the network with the proposed features and Unigram, the training accuracy was 85.5% and 86.1%, as shown in Figure 4 and Figure 5, respectively. As we can see, in comparison with BP, the slope of GA is steeper. This indicates that the convergence to the optimal solution in GA was faster than BP. For example, at the 50th iteration, the accuracy of BP was less than 75%, whereas it was 85% in GA. We also can notice that the accuracy when the Unigram model was used is slightly higher than the proposed features. Nevertheless, using the proposed features avoided degrading the convergence performance since it is computationally inexpensive to search through a small search space. Furthermore, this study mainly concerns the merits of using GA to train a neural network for sentiment classification, regardless of the nature of features that are used.

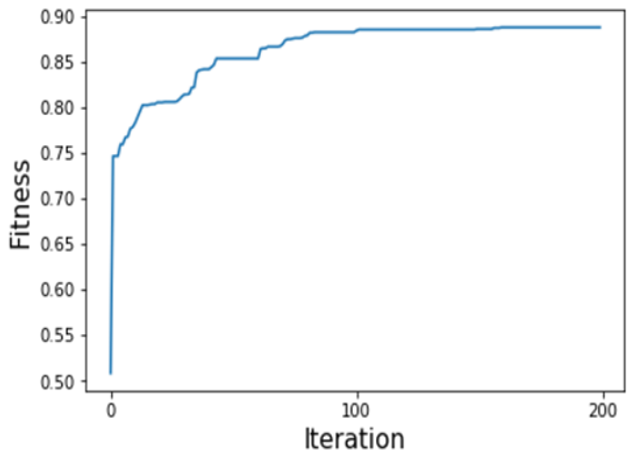

Figure 3. Accuracy of proposed method during training process

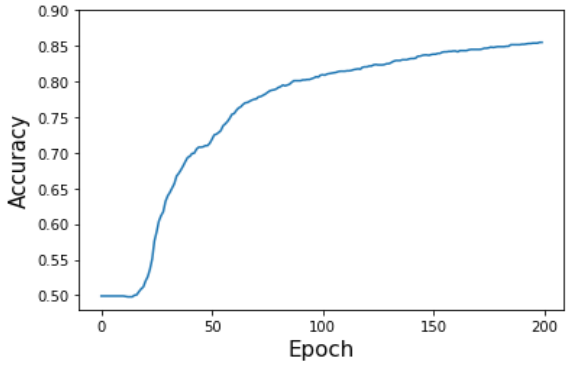

Figure 4. Accuracy of BP-DNN during training process based on proposed features

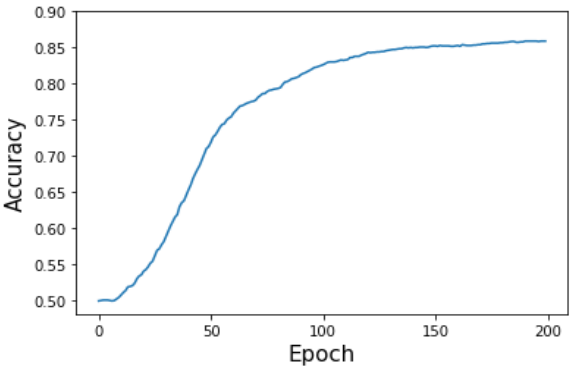

Figure 5. Accuracy of BP-DNN during training process based on Unigram

After the best solution has evolved, and the network dynamically learned during the evolution process, the optimal or near-optimal weights obtained were applied to the validation set to tune the parameters to their best values. Next, the efficiency of the proposed method was tested on the reviews in the test set. The summary of the results obtained is tabulated in Table 2. Noticeably, GA outperformed BP algorithm in optimizing the DNN either with the proposed feature or Unigram, where the overall classification performance significantly increased. For example, the overall precision, recall, and F-measure when the proposed features are used have significantly improved with points of 3%, 9.6%, and 6.2%, respectively. This is mainly because of the capability of GA to overcome the problems associated with BP. The results indicate that using GA to optimize DNN is adequate to show improvement in the sentiment classification performance. We also can notice that although the proposed features achieved lower value in precision and F-measure of BP-based DNN, they have been shown to be more effective than Unigram representation in the recall.

Table 2. The results of the proposed method on the test set

Model	Precision	Recall	F1-Measure
BP-DNN Ngrams	91.6	79.8	85.3
BP-DNN features	86.5	82.3	84.5
Proposed model	89.5	91.9	90.7

5. CONCLUSION

This paper presents a method that uses GA to train a DNN for improving the performance of Arabic sentiment classification. The proposed method optimizes the connection weights using an evolutionary process in which genetic operators such as selection, crossover, and mutation are utilized. Also, Arabic sentiment lexicon and linguistic knowledge to extract features are involved in extracting features for text representation. The results show that the proposed method has the capability to avoid the limitations of BP, and combine the advantages associated with each of GA and DNN, which resulted in a good performance for sentiment classification. Additionally, it shows that this method has the ability to yield considerable improvement and converge fast compared to BP-based DNN. Nevertheless, there is a room for improvement as the proposed features have insignificant contribution compared with Unigram model. Further work is also required to explore the effectiveness of the proposed method with deeper layers and diverse configurations on different Arabic benchmark datasets. Additionally, to extend the work so that it can be applied to other Arabic sentiment analysis tasks such as aspect-based sentiment analysis.

REFERENCES

[1] Zhang, L., Liu, B. (2016). Sentiment analysis and opinion mining. In: Sammut, C., Webb, G. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7502-7_907-1

[2] Oueslati, O., Cambria, E., HajHmida, M.B., Ounelli, H. (2020). A review of sentiment analysis research in Arabic language. Future Generation Computer Systems, 112: 408-430. https://doi.org/10.1016/J.FUTURE.2020.05.034

[3] Al-Sughaiyer, I.A., Al-Kharashi, I.A. (2004). Arabic morphological analysis techniques: A comprehensive survey. Journal of the American Society for Information Science and Technology 55(3): 189-213. https://doi.org/10.1002/asi.10368

[4] Boudad, N., Faizi, R., Oulad Haj Thami, R., Chiheb, R. (2018). Sentiment analysis in Arabic: A review of the literature. Ain Shams Engineering Journal, 9(4): 2479-2490. https://doi.org/10.1016/J.ASEJ.2017.04.007

[5] El-Beltagy, S.R., Ali, A. (2013). Open issues in the sentiment analysis of Arabic social media: A case study. In 2013 9th International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, pp. 215-220. https://doi.org/10.1109/Innovations.2013.6544421

[6] Baccianella, S., Esuli, A., Sebastiani, F. (2010). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. LREC, 10(2010): 2200-2204.

[7] Miller, G.A. (1995). WordNet: A lexical database for English. Commun. ACM, 38(11): 39-41. https://doi.org/10.1145/219717.219748

[8] Yue, L., Chen, W., Li, X., Zuo, W., Yin, M. (2019). A survey of sentiment analysis in social media. Knowledge and Information Systems, 60(2): 617-663. https://doi.org/10.1007/s10115-018-1236-4

[9] Kim, Y. (2014). Convolutional neural networks for sentence classification. pp. 1746-1751. https://arxiv.org/abs/1408.5882.

[10] Sallab, A.A. al., Hajj, H.M., Badaro, G., Baly, R., El-Hajj,

W., Shaban, K.B. (2015). Deep learning models for sentiment analysis in Arabic. the Second Workshop on Arabic Natural Language Processing, pp. 9-17.
[11] Abdelhade, N., Soliman, T.H.A., Ibrahim, H.M. (2018). Detecting twitter users' opinions of Arabic comments during various time episodes via deep neural network. In: Hassanien, A., Shaalan, K., Gaber, T., Tolba, M. (eds) Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017. AISI 2017. Advances in Intelligent Systems and Computing, vol 639. Springer, Cham. https://doi.org/10.1007/978-3-319-64861-3_22
[12] Abdullah, M., Hadzikadicy, M., Shaikhz, S. (2018). SEDAT: sentiment and emotion detection in Arabic text using CNN-LSTM deep learning. In 2018 17th IEEE international conference on machine learning and applications (ICMLA), Orlando, FL, USA, pp. 835-840. https://doi.org/10.1109/ICMLA.2018.00134
[13] Al-Smadi, M., Talafha, B., Al-Ayyoub, M., Jararweh, Y. (2019). Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews. International Journal of Machine Learning and Cybernetics 10(8): 2163-2175. https://doi.org/10.1007/s13042-018-0799-4
[14] Tang, D., Qin, B., Liu, T. (2015). Deep learning for sentiment analysis: Successful approaches and future challenges. WIREs Data Mining and Knowledge Discovery, 5(6): 292-303. https://doi.org/10.1002/widm.1171
[15] Haykin, S. (2010). Neural Networks and Learning Machines. New York: Prentice Hall.
[16] Gupta, J.N.D., Sexton, R.S. (1999). Comparing backpropagation with a genetic algorithm for neural network training. Omega, 27(6): 679-684. https://doi.org/10.1016/S0305-0483(99)00027-4
[17] Valian, E., Mohanna, S., Tavakoli, S. (2011). Improved cuckoo search algorithm for feed forward neural network training. International Journal of Artificial Intelligence & Applications 2(3): 36-43. https://doi.org/10.5121/ijaia.2011.2304
[18] Chiroma, H., Noor, A.S., Abdulkareem, S., Abubakar, A.I., Hermawan, A., Qin, H., Hamza, M.F., Herawan, T. (2017). Neural networks optimization through genetic algorithm searches: A review. Applied Mathematics & Information Sciences, 11: 1543-1564. http://dx.doi.org/10.18576/amis/110602
[19] Weise, T. (2009). Global optimization algorithms-theory and application. Self-Published Thomas Weise, 361.
[20] Du, K.L., Swamy, M.N.S. (2016). Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature. 1st ed., Birkhäuser Basel. https://doi.org/10.1007/978-3-319-41192-7
[21] Esfahanian, P., Akhavan, M.R. (2019). GACNN: Training deep convolutional neural networks with genetic algorithm. ArXiv abs/1909.13354.
[22] Gjylapi, D., Proko, E., Shehu, A. (2016). Genetic algorithm neural network model vs backpropagation neural network model for GDP forecasting. In RTA-CSIT, pp. 23-29.
[23] Nezamoddini, N., Gholami, A., Aqlan, F. (2020). A risk-based optimization framework for integrated supply chains using genetic algorithm and artificial neural networks. International Journal of Production Economics, 225: 107569. https://doi.org/10.1016/J.IJPE.2019.107569
[24] Revathi, J., Anitha, J., Hemanth, D.J. (2020). Training feedforward neural network using genetic algorithm to diagnose left ventricular hypertrophy. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(3): 1285-1291. http://doi.org/10.12928/telkomnika.v18i3.15225
[25] Dahou, A., Elaziz, M.E.A., Zhou, J., Xiong, S. (2019). Arabic sentiment classification using convolutional neural network and differential evolution algorithm. Computational Intelligence and Neuroscience, 2019: 2537689. https://doi.org/10.1155/2019/2537689
[26] Ishaq, A., Asghar, S., Gillani, S.A. (2020). Aspect-based sentiment analysis using a hybridized approach based on CNN and GA. IEEE Access, 8: 135499-135512. https://doi.org/10.1109/ACCESS.2020.3011802
[27] Yin, F.M., Xu, H.H., Gao, H.H., Bian, M.J. (2019). Research on weibo public opinion prediction using improved genetic algorithm based BP neural networks. Journal of Computer Science, 30(3): 82-101. https://doi.org/10.3966/199115992019063003007
[28] Ye, X., Yang, K. (2015). Optimizing neural networks for public opinion trends prediction. In 2015 11th International Conference on Natural Computation (ICNC), Zhangjiajie, China, pp. 31-36. https://doi.org/10.1109/ICNC.2015.7377961
[29] Alboaneen, D.A., Tianfield, H., Zhang, Y. (2017). Sentiment analysis via multi-layer perceptron trained by meta-heuristic optimisation. In 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, pp. 4630-4635. https://doi.org/10.1109/BigData.2017.8258507
[30] Nagarajan, S.M., Gandhi, U.D. (2019). Classifying streaming of Twitter data based on sentiment analysis using hybridization. Neural Computing and Applications, 31: 1425-1433. https://doi.org/10.1007/s00521-018-3476-3
[31] Govindarajan, M. (2013). Sentiment analysis of movie reviews using hybrid method of naive bayes and genetic algorithm. International Journal of Advanced Computer Research, 3(4): 139-145.
[32] Kotelnikov, E.V., Pletneva, M.V. (2016). Text sentiment classification based on a genetic algorithm and word and document co-clustering. Journal of Computer and Systems Sciences International, 55: 106-114. https://doi.org/10.1134/S1064230715060106
[33] Aliane, A.A., Aliane, H., Ziane, M., Bensaou, N. (2016). A genetic algorithm feature selection based approach for Arabic Sentiment Classification. In 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, pp. 1-6. https://doi.org/10.1109/AICCSA.2016.7945661
[34] Abbasi, A., Chen, H., Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. ACM Transactions on Information Systems (TOIS), 26(3): 1-34. https://doi.org/10.1145/1361684.1361685
[35] Zhu, J., Wang, H., Mao, J. (2010). Sentiment classification using genetic algorithm and conditional random fields. In 2010 2nd IEEE International Conference on Information Management and Engineering, Chengdu, China, pp. 193-196. https://doi.org/10.1109/ICIME.2010.5478084
[36] Ziani, A., Azizi, N., Zenakhra, D., Cheriguene, S.,

Aldwairi, M. (2019). Combining RSS-SVM with genetic algorithm for Arabic opinions analysis. International Journal of Intelligent Systems Technologies and Applications, 18(1-2): 152-178. https://doi.org/10.1504/IJISTA.2019.097754

[37] Heikal, M., Torki, M., El-Makky, N. (2018). Sentiment analysis of Arabic tweets using deep learning. Procedia Computer Science, 142: 114-122. https://doi.org/10.1016/J.PROCS.2018.10.466

[38] Soliman, A.B., Eissa, K., El-Beltagy, S.R. (2017). Aravec: A set of Arabic word embedding models for use in Arabic nlp. Procedia Computer Science, 117: 256-265. https://doi.org/10.1016/J.PROCS.2017.10.117

[39] Nabil, M., Aly, M., Atiya, A. (2015). ASTD: Arabic sentiment tweets dataset. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2515-2519.

[40] Al-Harbi, O. (2019). A comparative study of feature selection methods for dialectal Arabic sentiment classification using support vector machine. arXiv preprint arXiv:1902.06242. https://doi.org/10.48550/arXiv.1902.06242

[41] Alharbi, O. (2020). Negation handling in machine learning-based sentiment classification for colloquial Arabic. International Journal of Operations Research and Information Systems (IJORIS), 11(4): 33-45. https://doi.org/10.4018/IJORIS.2020100102

[42] Al-Harbi, O. (2017). Using objective words in the reviews to improve the colloquial Arabic sentiment analysis. arXiv preprint arXiv:1709.08521. https://doi.org/10.5121/ijnlc