Complex & Intelligent Systems (2021) 7:2485–2501
https://doi.org/10.1007/s40747-021-00436-4

ORIGINAL ARTICLE

An efficient sentiment analysis methodology based on long short-term memory networks

J. Shobana[1] · M. Murali[1]

Received: 4 January 2021 / Accepted: 9 June 2021 / Published online: 18 June 2021

Abstract

Sentiment analysis is the process of determining the sentiment polarity (positivity, neutrality or negativity) of the text. As online markets have become more popular over the past decades, online retailers and merchants are asking their buyers to share their opinions about the products they have purchased. As a result, millions of reviews are generated daily, making it difficult to make a good decision about whether a consumer should buy a product. Analyzing these enormous concepts is difficult and time-consuming for product manufacturers. Deep learning is the current research interest in Natural language processing. In the proposed model, Skip-gram architecture is used for better feature extraction of semantic and contextual information of words. LSTM (long short-term memory) is used in the proposed model for understanding complex patterns in textual data. To improve the performance of the LSTM, weight parameters are optimized by the adaptive particle Swarm Optimization algorithm. Extensive experiments were conducted on four datasets proved that our proposed APSO-LSTM model secured higher accuracy over the classical methods such as traditional LSTM, ANN, and SVM. According to simulation results, the proposed model is outperforming other existing models in different metrics.

Keywords Sentimental analysis · Adaptive Particle Swarm Optimization · LSTM · Skip gram · Feature extraction

Introduction

With the proliferation of Web2.0, people are increasingly expressing and sharing their opinion through social media. For example, micro-blogging websites like Twitter are becoming a very popular communication tool. The analysis of this site reveals a large number of social messages expressing their views and feelings on various topics and aspects of life. Because of this expansion, a lot of information is created. Along these lines, opinion investigation was introduced as a device for automatically extracting insightful and valuable data from client-produced information [1]. Sensory analysis is the major natural language processing (NLP) tasks. Due to its usefulness and challenges, this field attracted numerous researchers and professional communities [2, 3]. With the support of social media, people can share information with each other as well as messages opinion, and ideas. A huge number of people utilize websites like Facebook, Twitter, WhatsApp, Instagram, Google Plus, and LinkedIn to express their opinions. Presently about 2.46 billion people utilize social media, with an anticipated increment to 3.02 billion until 2021 [4].

They also used for sharing their daily life events that lead to collecting large and different types of data. People want to share about a specific product's experience using posts, likes, and reviews, and it gives companies a chance to gather this information and analyze the fame of their item and services [5]. How media is created and distributed through sharing and the realization of messages without any control is a revolution. Moreover, social media has an important impact on the business, advertising and e-commerce industry, as it describes consumers' behavior and perceptions about specific business plans, services, and products. Monitoring social media activity is an incredible method to quantify loyalty of the customer, track their impressions of items or brands, influence crusades and the achievement of marketing messages, and identify the best influencers that are generally applicable to the campaign, item or brand. Thus the Social

✉ J. Shobana
shobanaj@srmist.edu.in

M. Murali
muralim@srmist.edu.in

[1] SRM Institute of Science and Technology, Chennai, India

media is recognized as the next logical marketing platform [6, 7].

Specifically, comments on products in tweets are merit mining. Vendors can get purchasers' feedback progressively and afterward update their own items to be increasingly serious in the commercial center; Buyers can pick up the experience of others through these comments to decide if to purchase an item [8]. The tweets in real-time have a large influence on network transmission [9]. And also, the opinions are significant for organizations to know what customers are saying, whether it is positive or negative. If it is negative means, they can improve the product quality and satisfy the customers. Even though to find the customer's opinion is a difficult task, Most of the algorithms belong to machine learning, feature extraction methods are developed for sentimental analysis. The main contribution of the proposed work are summarized as:

- We proposed optimization LSTM algorithm model for effective sentiment analysis. As the improved particle swarm optimization algorithm selects the weight parameters of LSTM, computational complexity reduced and the accuracy of the work improved.
- Skip-gram word embedding method is utilized to obtain the overall higher accuracy. This word embedding model achieves superior results over other word representation.
- The evaluation results executed on four datasets show that our Optimized LSTM model is effective. Furthermore, we analyze different metrics such as accuracy, recall, precision, and F-measure. The results are proved through conducted experiments.

Literature review

Numerous machine learning algorithms are used for sentimental analysis. In [10] sentimental analysis is developed using Naive Bayes classifier. Here, they identify whether the particular sentence is positive, negative or neutral. In [2], Online Movies' Reviews based Sentiment Analysis of utilizing enhanced k-Nearest Neighbor Classifier was explained. In [11], Sentiment Analysis of Customer Product Reviews Using support vector machine is analyzed. Moreover, Deep neural networks also known as DNNs have currently attained huge gains in a different NLP undertakings, for example, language modeling [12], sentiment analysis [13], syntactic parsing [14], and machine interpretation [15]. A recurrent neural network also known as RNN is an uncommon kind of neural system, where connections are made between units which structure a directed cycle, which permits it to show a unique temporal behavior for the model. One of the special variations of RNN is LSTM. Nowadays, many researchers have developed LSTM for sentimental analysis, but the work in this area is not enough in terms of accuracy. So, an efficient sentimental analysis methodology is presented in this work.

The purpose of the proposed methodology is the automatic prediction of customer opinion of different products. The proposed model has three modules namely, pre-processing of text, implicit and explicit feature engineering, and polarity prediction based on sentiments. Gathering reviews from social media is the first task and then pre-processing is done to enhance the data quality. Text pre-processing is the first step. Feature engineering is done using skip-gram-based word embedding. Then, the extracted features are given to the LSTM network.

The remaining portions in the paper are sorted as pursues; in "Literature review" some of the existing literatures are discussed.

Many researchers had developed customer review-based sentimental analysis. some of them are discussed in this section. Sheng et al. [16] explained the finding of a rumor based on consumer opinion. To achieve the predictive process, they used a convolution neural network with LSTM (CNN-LSTM). In this paper, LSTM was incorporated into the pooling layer on CNN. Meanwhile, perception has been added as an important element in the rumor detection model. The effectiveness of this method has been verified by experiment. Moreover, Zhigang et al. [17] explained a stock closing forecast depends on LSTM and sentiment analysis. Here, they were first introduced to include investor's perceptions of stock forecasting, which improved the forecasting accuracy of the model. Second, the stock price sequence is a complex time series with varying time fluctuations, making accurate forecasting more challenging. Then, they introduced a gradual decomposition of the stock price complex by adding the empirical model decomposition (EMD), which attained better predictive accuracy. Third, they adopted LSTM because of the merits of evaluating the relationships between time-series data via its memory function. The experimental results depicted that their proposed model not only improves the accuracy but also reduces the delay.

Fu et al. [18] described Lexicon- enhanced LSTM using an attention mechanism. The research focused on word embedding quality improvement and thereby improve the sentiment classifier's accuracy using the sentiment lexicon. He carried out experiments on three English datasets namely IMDB, Yelp2013, MR and two Chinese datasets namely NB4000 and Book 4000. when compared with ALE-LSTM and WALE-LSTM, the proposed method obtained higher accuracy of 89%, 60.6%, 79.9%, 93% and 96%. Guixian et al. [19] proposed a BiLSTM model sentiment analysis tasks of comment text. Sentiment information concatenates with traditional TD-IDF word representation is utilized in this work. Relu activation function is used to overcome overfitting and gradient vanishing problem with the help of the gating mechanism.

Similarly, Latif et al. [20] had explained an ensemble approach for sentimental analysis. Here, they introduced a combination of three kinds of features namely, unigram, lexicon, and phrases. Then, two-level ensembles were introduced for the selection of features by using Logistic Regression (LR), Gini Index (GI), SVM, and Information Gain (IG). Finally, the classification was done by SVM. Here, they reviewed different products namely, Books, DVDs, electronics products, kitchen items, and movies review. The performance of the introduced method attains the maximum accuracy of 81.85%, 91.45%, 89.70%, 93.05% and 97.60% for electronics, DVD, Books products, kitchen items, and movies review respectively.

In [21], Graber et al., had explained an aspect-based opinion mining of drug review. Here, initially, SA was performed to foresee the sentiments concerning by and large fulfillment level, effectiveness, and side effects of client reviews on particular drugs. Then, the features are given to the logistic regression model to predict the recommended drugs based on positive reviews. This method was attained a maximum accuracy of 75.8%. Da'U and Salim [22] presented an efficient neural attention-based recommendor system. This model composed of LSTM encoder, semi-supervised topic model, mechanism for co-attention, and an analysis layer for predicting the rating of the users. The specialty of this model is the better learning capacity of the aspects of the products and sentiment lexicons and thereby improve the efficiency of the recommender system. Sailunaz and Alhajj [23] explained sentiment analysis task on Twitter comments. The main contribution of this work is the detection of emotions and sentiments from the posts and tweets people in the social networks using Naïve Bayes classifier. This model provides topic based general recommendations and user-based customized recommendation.

In [24], Shoieb and Ajit had explained an emoticon-based SA using web data. Initially, reviews are collected from the web. Then, pre-processing was done. In data pre-processing, POS tagging, stop-word removal, lemmatization, and emotion processing were applied. Then using sentiwordnet emotion-based features using derived. After the feature extraction process, classification was done by using different classifiers namely, SVM, IBK, MLP, and Naive bias classifier. Among the classifier, naive bias classifier attained the maximum precision of 84.7% for the college dataset and 83.3$ for using hospital dataset. In this paper, reviews are labeled as neutral tweets, positive, strong positive, strong-negative, weak-positive, negative, and weak-negative. Additionally, Zeeshan et al. [25] had presented a lexicon and ANN-based SA. Here, they utilized the movie review dataset and the dataset consists of two labels namely positive and negative. The trained network managed to achieve a final accuracy of 91%. Ankit et al. [26] suggested an ensemble sentiment classifier for Twitter sentiment analysis. The computation of the work is to predict the category of the tweet as positive, neutral or negative. Four datasets like Stanford-sentiment 140 corpus, Health Care Reform (HCR), First GOP debate, and sentiment analysis dataset were collected from Twitter and used for testing.The proposed ensemble technique acquires 75.79%, 70.28%, 76.85%, and 73.33%.

Weijiang et al. [27] had explained a Bidirectional LSTM with multi-channel features and self-attention mechanism (SAMF-BiLSTM) for classification of sentiment. SAMF-BiLSTM model was completely misuse the connection between sentiment polarity words and target words in a sentence, and does not depend on physically sorted-out opinion dictionary. Moreover, they presented the SAMF-BiLSTM-D model dependent on SAMF-BiLSTM model for record-level content order assignments. The strategy gets the portrayal of all sentences in the archive through SAMF-BiLSTM preparing, at that point coordinates BiLSTM to become familiar with the portrayal all things considered, and further gets the assessment include data of the whole record. At long last, they assess test results under various datasets. The outcomes show that SAMF-BiLSTM and SAMF-BiLSTM-D were better than other propelled strategies in order precision as a rule.

Moreover, Alqaryouti et al. [28] had presented aspect depend sentimental analysis utilizing data of government review. This approach has been adopted to address the challenges of language analysis techniques, rules, and dictionaries in many sensory analysis and to provide concise results. Identification of Indirect features in this approach which used to enhance the accuracy of the feature extraction process. Furthermore, the combined model for classification surpasses the dictionary-depend criteria and other rule combinations by an average accuracy of 5%. While utilizing the same dataset, the introduced method surpasses machine learning schemes using the support vector machine abbreviated as SVM. Ray and Chakrabati [29] proposed a combined approach of the Rule-based and Deep Learning method for aspect level sentiment analysis. This research used dependency parsing, machine learning techniques and seven-layered deep convolution neural network (CNN)for tagging each aspect in the comments.

Akyol et al. [30] describes a Social Impact Theory depend Optimization Algorithm and whale optimization model based on opinion mining. The prevalent exchanging methodology dependent on the sentiment feedback quality between the tweets and news utilizing conventional programming optimization strategy was discussed by Yang et al. [31]. Keshavarz et al. [32], proposed a genetic algorithm-based sentimental analysis model. Six different datasets are used for conducting experiments and the result achieved higher accuracy. In Paper [33], the author proposed a sentimental analysis method based on the cross-domain aspect. They introduced a heterogeneous organization depending on the depiction that merges different qualities into a single network.

Meskele and Frasincar [34] given a neural network-based ontology model for giving an answer for sentence-level opinion mining. Based on a viewpoint's opinion esteem in a given sentence. For estimating the importance of the words in the given sentence based on the aspect's sentiment value, bidirectional mechanism is used. Liang et al. [35] described a topic embedding model for short texts. Gibbs sampling process is utilized in this research to enhance the topic coherence. Global and Local word embedding contributes more in increasing the performance of this model.

Hu et al. [36] proposed a LSTM network that aggregates the PSO algorithm for safety forecast model. Enhanced PSO–GD aggregated LSTM is best suited for the analysis for Time-series data. GD methods are applied iteratively to LSTM parameters to reduce the cost and to improve accuracy. Wang et al. [37] presented a solution for toll station-based lane work-schedule using toll data. LSTM and PSO algorithm predicts the average length queue of the lane with the three input parameters such as the traffic volume, queue length average, and time taken for service. The proposed PSO-LSTM method increased 2% and 3% of the accuracy respectively when compared to the SVR models and traditional LSTM. At last, the toll station operating cost is used for the estimation of work-schedule of toll lane.

Proposed sentiment analysis using Optimized LSTM

The proposed model's workflow diagram is depicted in Fig. 1. As shown in the figure, reviews or tweets from the datasets such as the Amazon dataset, Trip Advisor dataset, Demonetization dataset and Book review dataset are pre-processed by pursuit the phased tokenization, stop words removal, stemming and segregation. Then the pre-processed Twitter words are represented as a vector using one-hot encoding representation method. Skip-gram based word2vec architecture model is used to mention the words

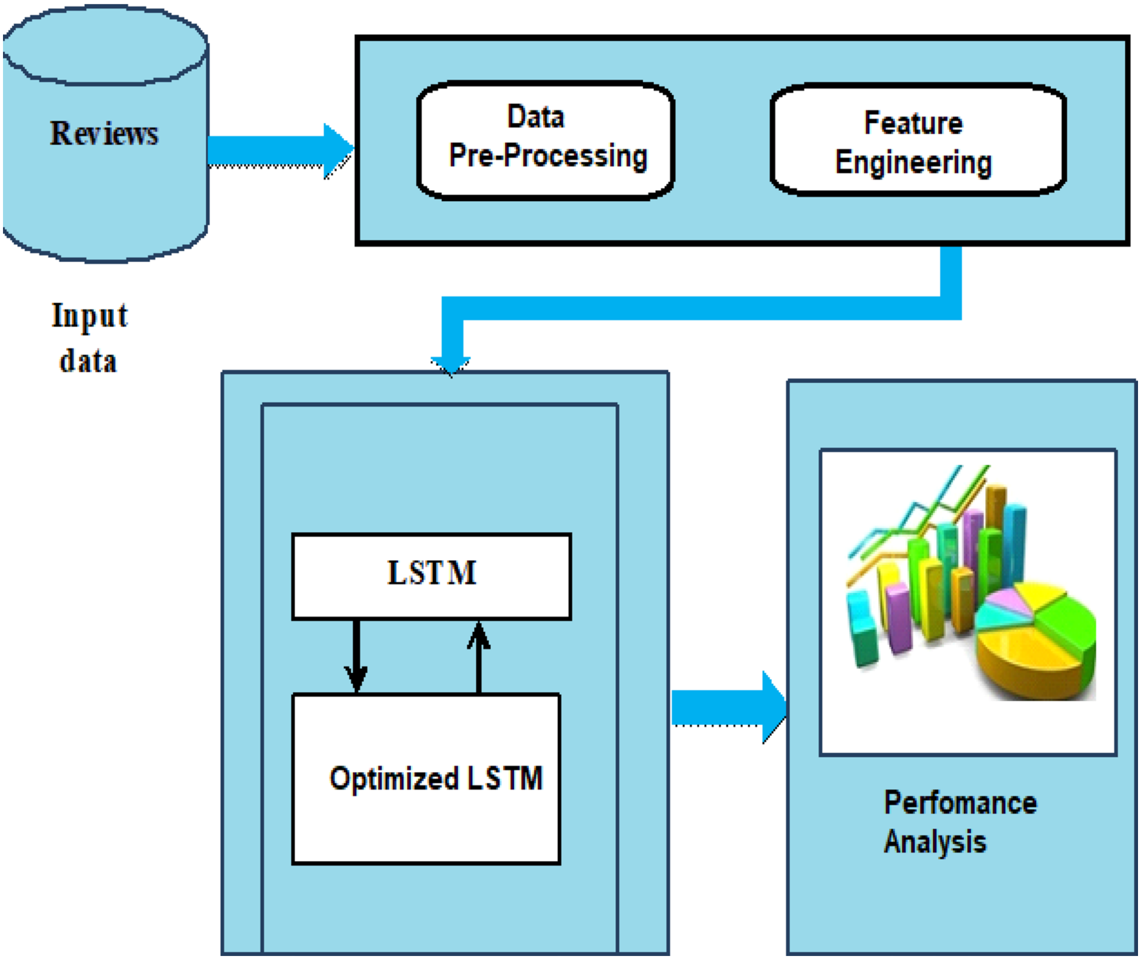


Fig. 1 The Flow diagram of proposed model

in lower-dimensional space and to make the representation more accurate. Finally, the input tweet or review is classified as positive or negative polarity using the LSTM network. To improve the performance of the LSTM, optimal weight parameters using the Adaptive Particle Swarm Optimization (APSO) algorithm are chosen.

Pre-processing

Before classifying the sentiment of tweets or reviews, the following steps are applied on datasets.

Tokenization: in this process, tweets are splitted into phrases or tokens, symbols and words.

Stop words removal: stop words such as I, and, the, for, should and etc., are removed from the tweets using NLTK's(Natural Language Toolkit) stop word list.

Stemming: the main work of Stemming process is to reduce the words to its base forms with the help of morphology stemming. For instance, the words 'advising' and 'advised' are reduced to their root word as 'advice'.

Segregations: in this process, the special characters such as "' ? !;: # $ % & () * + − / < > = [] n ˆ _ { } |~ are removed from the tweets.

Padding the excess data: one of the significant task of opinion analysis is Padding**.** Padding deals with reviews of different lengths by padding or truncating the reviews to a fixed length. The fixed length is determined by sequence length. Padding helps in the process of reducing wastage of memory. Like the number of time steps in LSTM layer, sequence length is defined. In reviews, if the seq_length is lesser, 0 s wil be dumped and if the seq_length is higher then, reduction should be done based on the first seq_length words.

Words to vector representation

To convert words as vectors, one-hot encoding technique is used. Figure 2 delineates the one-hot encoding strategy. It refers to words as vectors that are similar in size as vocabulary dictionary. For illustration, as in Fig. 1, if there are 100 words in a dictionary, the words in a sentence deals with a dimensional vector size of 100. The Skip-Gram algorithm is used to marks words at a lower-dimensional space and to show their significance in a vector.

Skip-gram word embedding model

Numerous words list are extricate from contents statement into vectors with lower dimensions, usually from 10 to 1000 dimensions. Break down the frequency of words in text analysis and statements by large neglects of the sequence of sentences, paragraphs, and words. In any case, this sort of evaluation may confine the comprehension of words' importance in the sentence on the grounds that the logical significance of the words and the presence of the words are barred from the analysis. In this way, we used the word embedding method to more precisely understand users' reviews and to understand the basic qualities and meaning of words.

Skip-gram algorithm is used in this proposed work as it shows better accuracy in extracting relationships between semantic words. The structure of the skip-gram word2Vec representation is shown in Fig. 3. The skip-gram word2Vec model speculates words that can be shown with regards to the current word. The information layer utilized words introduced as one-hot encoding vector. This vector implies as a context word, just one unit from U units, $\left\{x_1, x_2, ...,x_U\right\}$, will be 1, and every single other unit are 0 as appeared in Fig. 1. The size of one hot encoder vector is equivalent to the size of the vocabulary dictionary utilized in the phase of training. The words utilized by the input layer are anticipated to a M-dimensional vector W of U × M size are contributed

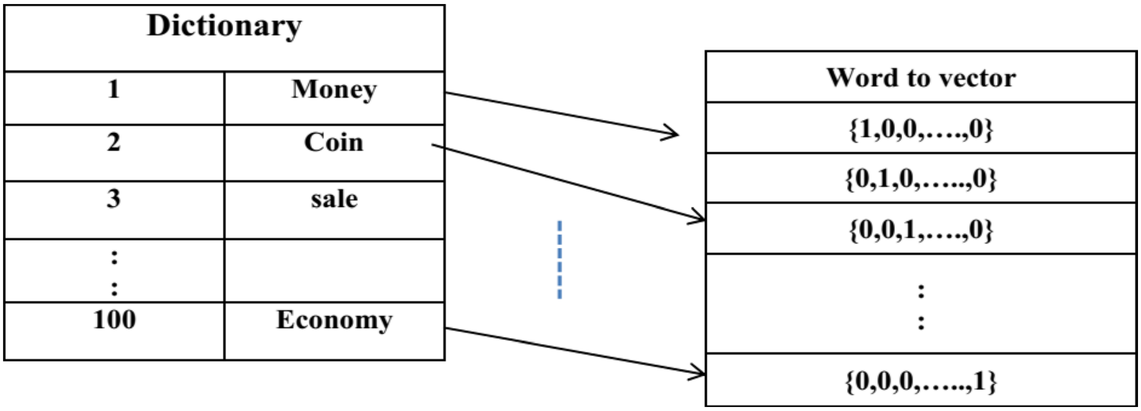


Fig. 2 An example of one-hot encoding technique

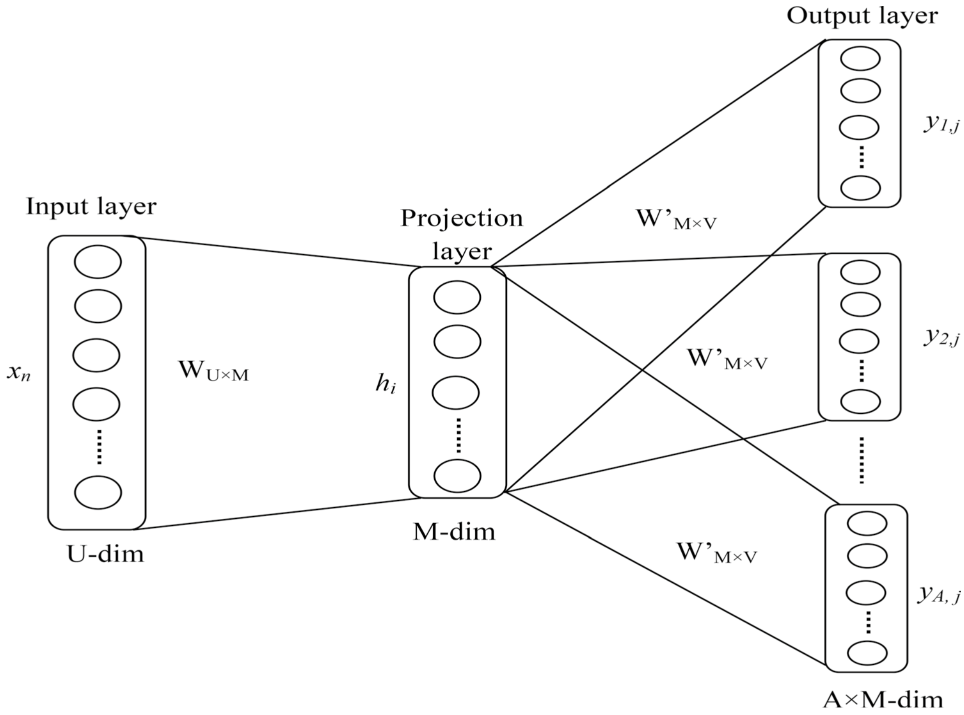


Fig. 3 Architecture of skip-gram model

towards the projection layer. Yield from this is increased by the weight matrix W' of M × U size by a projection matrix and given to the output layer. Finally with the help of softmax function, the output layer predicts neighborhood words. However, softmax function is costly in terms of computation and it reduces the effort for training. Along these lines, hierarchical softmax is used in the output layer rather than softmax function. For estimating the probability in hierarchical softmax, a binary tree structure is used to predict words. Dissimilar to the softmax function, the hierarchy softmax does not need to search for the entire word, so we can decrease the computation time in the output layer and the output probability of a word is calculated as follows:

$$\mathrm{pr}\left(w/w_{\mathrm{input}}\right) = \prod_{j=1}^{L-1} \sigma\left(m(w, j+1) = \mathrm{child}(m(w, j))\, v'_{m(w,j)}\ v_{w_{\mathrm{input}}}\right), \tag{1}$$

where, input word is represents as w_{input} and v denotes the input vector and v' denotes the output vector representations of the word w. $\sigma()$ represents a sigmoid function. Path length is denoted as L. The jth node in the binary tree is represented as *m(w,j)*, child(m) denotes a child node of $m$chosen subjectively. Then, the predicted words should minimize error function and the error is determined as follows:

$$\mathrm{Error} = \frac{1}{N} \sum_{n=1}^{N} \sum_{-c \le j \le c} \log\ \mathrm{pr}\left(w_{n+j} \middle| w_n \right), \tag{2}$$

where the size of the context is represented as c and the number of word sequences is represented s N. The goal of this function is to update the weight matrix *W*' with minimum error rate. In the same context, if two reliable words are used, both words will be assigned with the un-alike vector values. In these ways, various assumptions and analyses are made based on them.

Sentimental analysis using optimized LSTM

The output vectors or features are given as classifier's input which classifies the tweet or review as positive context or negative context. The set of feature is denoted as follows,

$$Y = \left\{ Y_1,\ Y_2, \ldots.., Y_S \right\}, \tag{3}$$

whereY_S denotes the *S*th feature.

A special type of recurrent neural network (RNN) called as LSTM neural network, is utilized in the proposed model for classification. Conventional neural structures do not consider sequential factors and cannot review the substance of past. RNN was structured for understanding this issue. Figure 4 displays that the structure of an RNN. The hidden state Ht time is purchased from the Yt information and ultimately from the Ht − 1 yield. It is used to find the loss model of the current layer and to determine the Ht + 1 of the following layer. Regardless, the hidden structure of the RNN sequence index position t was upgraded to keep the target distance from the gradient disappearing issue, considering the way an RNN gradient would deal with the breakdown. Then, an

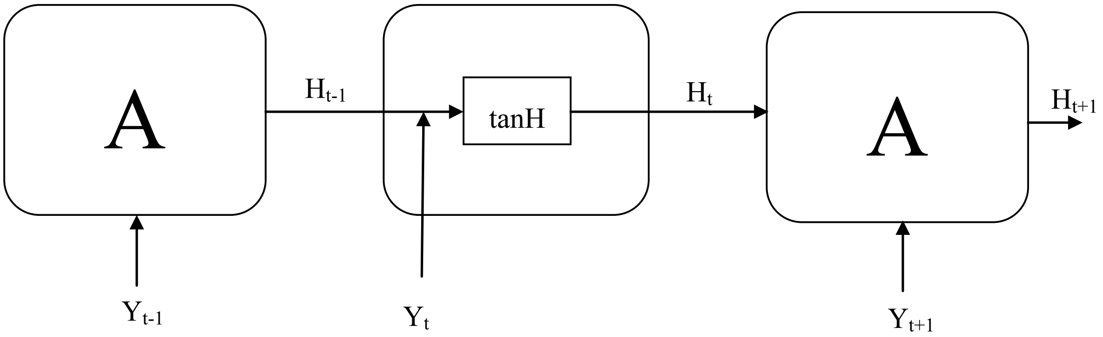


Fig. 4 Structure of RNN

abnormal RNN model called LSDM can adjust the long-distance reliability information. LSDM is somewhat inaccurate to the general neural system module An of RNNs. In RNN, the re-emphasized NN module A has a basic structure, for example, a tanH layer.

$$H_t = \tan H\left[w_H\left(Y_t, H_{t-1}\right) + c_H\right], \tag{4}$$

where c_H denotes the control parameters of hidden state. tan$H$ allows neural network to add or remove information to the previous input. w_H refers to the weight parameter of the hidden state H_t.

On the other hand, Fig. 5 displays that LSTM consists of four neural network layers that connect within an exceptional way. By means of a phenomenally designed structure named as a "gate", LSTM can add or delete information to the memory cell state. This is the area where gate actually selects operational data, viz. features of the input. It has the sigmoid neural system layer and multiplication function. The sigmoid layer switches over the input values of the features through the sigmoid function and outputs value some place in the scope of 0 and 1, depicted how much input features can encounter that in Section A of framework. "0" indicates that no data is permit to pass. "1" shows that all data is permit to pass. At each sequence index level t, the gate system in the LSTM, everywhere, connected to the gates. The sigmoid output is to assumed if the limit is of [0,1], Eqs. (5)–(9) describe how a model of LSTM works, as shown in Fig. 5

The Forget gate will select which information to discard or keep from last minute's memory:

$$F_t = \sigma\left[w_F\left(Y_t, H_{t-1}\right) + c_F\right], \tag{5}$$

where F_t represents forget gate. c_F and w_F denote the control and weight parameters at forget gate. Y_t represents input at the current timestamp, $H_t - 1$ denotes the output obtained at the timestamp $t - 1$ from the previous block of LSTM. σ denotes the logistic sigmoid function and output. If the output is '0' then it means blocking of gates. If the output is '1' gates letting everything pass.

The input gate chooses the information that should be stored:

$$I_t = \sigma\left[w_I\left(Y_t, H_{t-1}\right) + c_I\right], \tag{6}$$

where I_t is the input gate; σ is the sigmoid function; $H_t - 1$ is the output of previous timestamp; Y_t si the output of the

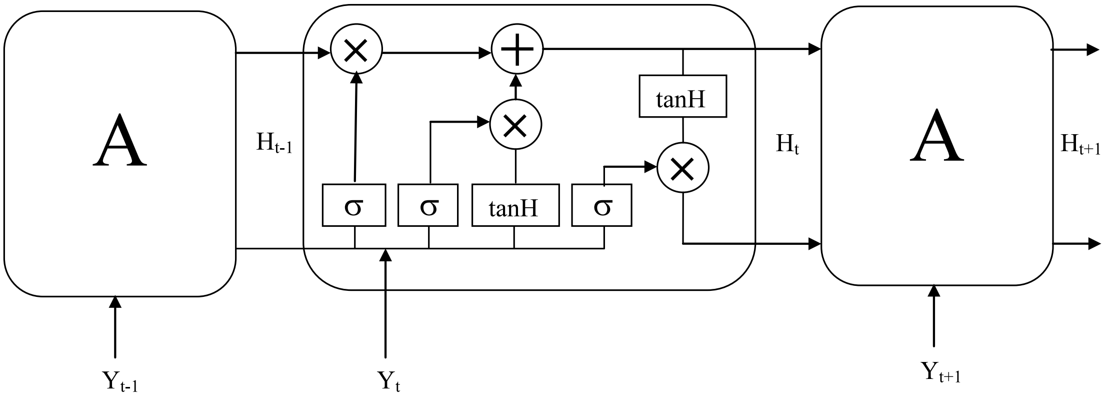


Fig. 5 Structure of LSTM

current LSTM block; W_I is the weight parameter of the input gate neurons; C_I is the points to the bias for the input gate.

Another candidate value vector is made by a tan$H$ layer and is denoted as follows:

$$V_t = \tan H\left[w_V\left(Y_t, H_{t-1}\right) + c_V\right], \tag{7}$$

where, tan$H$ allows LSTM to add or remove information to the previous input. V_t denotes candidate at timestamp (t) for the cell state. c_V and w_V refer to the control and weight parameters at tan$H$ layer.

The input gate selects the generation of the candidate value vector, and the forget gate selects whether to keep or discard the information to create the final memory.

$$V_t = F_t * V_{t-1} + I_t * V_t, \tag{8}$$

where V_t denotes memory cell state at current timestamp(t) and * represent the element wise multiplication of the vector. Finally, the output gate determines which part of the memory is in the long run yielded:

$$O_t = \sigma\left[w_O\left(Y_t, H_{t-1}\right) + c_O\right]. \tag{9}$$

By then, the passed data streams into the tanH layer for getting ready. The output is regard between [−1, 1] and the output gate multiplies the yielded regard. Ot denotes the output gate. W_o carries the weight parameter of the output gate neurons. σ points the sigmoid function.

$H_t - 1$ denotes the obtained output from the previous timestamp, Y_t points to the obtained output of the current LSTM block, c_0 points to the bias for the input gate. Finally, the output is evaluated by

$$H_t = O_t * \tan H\left(V_t\right), \tag{10}$$

where * denotes the vector's element-wise multiplication. Through the softmax output layer, predicted output from the current block is obtained and is pointed by H_t. V_t denotes memory cell state at the (t) current time stamp.

Finally, the loss function of this model is estimated by calculating the MSE (mean square error). The calculation is done as follows

$$\text{Loss} = \sum_{t=1}^{N}\left(H_t - T_t\right)^2, \tag{11}$$

where T_t denotes the desired output. N is the prediction generated from a sample of n data points. Loss calculates the mean squared prediction error.

The mean square error (MSE) is the average squared difference between the values evaluated and the actual value.

If the estimated score is below 0 (negative values), then the tweet or review is considered as a sentiment of negative and if the estimated score is above 0 (positive), then the tweet or review is considered as a positive sentiment and lastly if the estimated score is 0 it is considered as neutral.

Weight optimization using APSO

To enhance the performance of the LSTM, Weight parameters $\left\{w_F, w_I, w_V \text{ and } w_O\right\}$ from Eqs. (5) to (9) are optimized utilizing the proposed APSO algorithm. These weight parameters are used to attain the target output in LSTM. In the training phase, the LSTM network is to be trained. For reducing computational complexity, LSTM network is trained by selecting the weight parameters using the proposed APSO algorithm. Before executing the LSTM network, weight parameters of each layer within the range [0, 1] are given as input to the APSO algorithm. The PSO algorithm was developed in 1995 by Kennedy and Eberhart. This mechanism is motivated by the behavior of the flock of bird. It is a population-based evolutionary algorithm which starts with a population of particles with random solutions. Each particle is the initialization of solutions to the evolutionary optimization problem. The random position and velocity have been initialized for the Particles. The particle swarm optimization (PSO) algorithm is the best optimization algorithms as it has only fewer parameters to adjust. The PSO technique is getting famous because of its straightforwardness of execution and capacity to achieve a good solution.

The PSO has 4 levels:

(1) Initialization of particles,
(2) Estimating the particles with the fitness function,
(3) Renewing particle positions and velocities,
(4) Updating the experiences of the particles with the general knowledge of the swarm.

For increasing population diversity and for avoiding the premature convergence of PSO, the opposition-based learning (OBL) method is used with PSO. Opposition-based learning (OBL) combined with PSO becomes the Adaptive Particle Swarm Optimization (APSO).

APSO optimizer plays key role in increasing the accuracy of the proposed LSTM neural network model by adjusting the attributes such as weights and learning rate in order to reduce losses.

Opposition based learning (OBL): OBL has amazing learning and optimization skills. OBL (opposition-based learning) is well known for its ability in selecting best optimal solution from a set of feasible solutions with fast convergence. OBL is integrated with the PSO variant to have the ability to control the particle's speed and its directions using velocity clamping. It helps to choose the finest particles from the current swarm and its counter swarm to improve the fitness of the whole swarm.

While surveying an answer x to a given issue, figuring its opposite answer x' at the same time,gives another opportunity to estimate a closer solution for the global optimum. The concept of opposition-based learning can be integrated with neural networking;

Opposite weights: for all the weights chosen for ANN, the opposite weights are generated. This process is analogous to mutation mechanisms in genetic algorithms. The number of weights to choose and the way to select them offers a wide variety of feasible schema that can be inspected. Using this algorithm, optimal weight parameters' are choosing and are described as followed:

Initialization**:** with a d dimensional vector, the particles or candidate solutions are initialized. In this algorithm, weight parameters of LSTM are considered as the candidate solutions and are selected within the range [0, 1].

The initialization is done as,

$$X_k(i) = \left\{X_{k1}(i), X_{k2}(i), \ldots.. X_{kd}(i)\right\}, \tag{12}$$

where $X_{kd}(t)$ denotes the kth particle's position in the dth dimension vector at iteration i and defines as follows

$$X_{kd}(i) = \left\{w_F, w_I, w_V \text{ and } w_O\right\}_{kd}(i), \tag{13}$$

where w_F is the weight parameter of forget gate, w_I is the weight parameter of the input gate neurons; w_o si the weight parameter of the output gate neurons; w_V weight parameter of tan$H$ layer.

Besides, the opposite solution is represented as follows;

For a given problem, whenever a answer x is found then the opposite answer $\bar{X}$ is to be evaluated. This evaluation may be based upon experience or randomly guessed solution.

$$\bar{X}_k(i) = \left\{\bar{X}_{k1}(i), \bar{X}_{k2}(i), \ldots.. \bar{X}_{kd}(i)\right\}, \tag{14}$$

where, $X \in [a, b]$ is a real number and the opposite solution $\bar{X}$ is estimated as,

$$\bar{X} = a + b - X, \tag{15}$$

where b and a are the maxima and minimum weight values individually.

Fitness: using the conditions (16), every solution's fitness value is appraised after initializing the candidate solutions and opposite solutions. Fitness function is defined using the Eq. (11),

$$F(i) = \mathrm{Min}(\mathrm{Loss}(i)). \tag{16}$$

The optimal solution is the solution with the least fitness value. APSO finds the optimal weight based on the environment within less time duration. If the number of iterations increases then computational complexity will increase. This also minimizes the loss.

Updating velocity and position: After estimating each solution's fitness, as per the its velocity vector and position, the solution is updated. Using the equations of (17) and (18), the every solution is updated until determining the best solution. In each iteration, the particle's velocity have to be adjusted to its newly formed position Pbest and global position Gbest. According to conditions, the velocity v of each particle is to be updated.

$$V_{kd}\ (t+1) = w * V_{kd}(t) + \left(P_{\mathrm{best}_{kd}}(t) - X_{kd}(t)\right) c_1 r_1 + \left(G_{\mathrm{best}_d}(t) - X_{kd}(t)\right)\ c_2 r_2, \tag{17}$$

where $X_{kd}(t)$ represent the velocity and $V_{kd}(t)$ denotes the position of the kth particle. d denotes the dimensional space and t is the iteration. For any given particle's (P) velocity. P_{best} represents personal or local best values and G_{best} value represent global best value. We require to calculate the differences of these two. First ($\mathrm{G}_{\mathrm{best}\,-}P$) is calculated and then ($P_{\mathrm{best}}-$ P) is calculated. since any given particle's (P) velocity is as per the variation between the global best (G_{best}), and the particle best (P_{best}). c_1 and c_2 denote the coefficients of acceleration which is equivalent to 2. r_1 and r_2 denote the random variables inside the range [0, 1].

$$X_{kd}(t+1) = X_{kd}(t) + V_{kd}(t+1), \tag{18}$$

w denotes inertia weight. The inertia weight is used for the searching process. The inertia weight will be reducing while maximizing the iteration. This is estimated as follows,

$$w = w_{\mathrm{maximum}} - \frac{w_{\mathrm{maximum}} - w_{\mathrm{minimum}}}{t_{\mathrm{maximum}}} \times t, \tag{19}$$

wherew_{minimum} and w_{maximum} denote the minimum and maximum inertia weight respectively. t_{maximum} denotes the maximum number of iterations.$P_{\mathrm{best}_{kd}}(t)$ and $G_{\mathrm{best}_d}(t)$ represent the best position of the particle k and best position of the group at iteration t. If the kth particle's fitness ($X_{kd}(t+1)$) is lesser than that of the previous $P_{\mathrm{best}_{kd}}(t)$, then the particle is assumed as new $P_{\mathrm{best}_{kd}}(t+1)$. Otherwise, the particle $X_{kd}(t)$ is considered as new $P_{\mathrm{best}_{kd}}(t+1)$. Additionally, if the fitness of the kth particle ($X_{kd}(t+1)$) is lesser than that of previous $G_{\mathrm{best}_d}(t)$, then the particle is considered as new $G_{\mathrm{best}_d}(t+1)$. Otherwise, the particle $X_{kd}(t)$ is considered as new $G_{\mathrm{best}_d}(t+1)$.

$$P_{\mathrm{best}_{kd}}\ (t+1) = \begin{cases} X_{kd}(t+1) & \text{if}\, F\left(X_{kd}(t+1)\right)\ \le\ F\left(P_{\mathrm{best}_{kd}}(t)\right) \\ X_{kd}(t) & \text{otherwise} \end{cases}, \tag{20}$$

$$G_{\mathrm{best}_d}\ (t+1) = \begin{cases} X_{kd}(t+1) & \text{if}\, F\left(X_{kd}(t+1)\right)\ \le\ F\left(G_{\mathrm{best}_d}(t)\right) \\ X_{kd}(t) & \text{otherwise} \end{cases}. \tag{21}$$

Termination**:** until determining the best solution or weight parameters of LSTM, the above steps are continued. The iteration continues till the best solution is obtained. After selecting the optimal weight parameters which are used in the LSTM network for testing. Figure 6 depicts the flow diagram of the proposed APSO algorithm. The APSO algorithm is given as follows.

Algorithm: Optimize the weight parameters of LSTM using Adaptive Particle Swarm algorithm

1. Update inertia weight (*w*), and random variables (*r₁* and *r₂*), Acceleration coefficients (*c₁* and c₂) .
Where, *w* denotes inertia weight which is utilized to control the impact of the previous velocity of a particle on its current velocity. c_1 is the cognitive weight and c_2 is the social weight.. The random values r_1 and r_2 are in the range[0,1].

2. Initialize the candidate solutions or weight parameters

Initialize the velocity(v),the Position of particles (p), the Number of particles (k), The dimensional space (d), the iteration (*t*),the Individual particle experience (P_{best}) , and the common knowledge of the swarm (G_{best}) .

3. Initialize the opposite solution using OBL.

Find P_{best} using opposite-based learning . Choose and update G_{best} .

The fundamental principle of OBL is to simultaneously consider one best solution and the corresponding counter-solution to approximate the global optimum

4. Evaluate the fitness for each solution and opposite solution using (16).

Using evaluation function , Compute the fitness value of each particle . Best p is the particle's local best position, best g is the particle's global best position and obtain the opposition-based solutions of personal best positions .

5. Update velocity and position of the solution and opposite solution using (17) and (18).

At every iteration, each particle adjusts its velocity to be simultaneously close to its new position G_{best} and its P_{best} . The velocity of each particle, v, is updated according to the conditions.

6. Compare the fitness value for every iteration .

Each particle's current fitness value should be compared with its previous best fitness (*Pbest*). If the estimated current value is better than P_{best}, at that point set P_{best} equivalent to the current value and the P_{best} area equivalent to the current area in the d-dimensional space.

7. If fitness of $X_{kd}(t+1)$ is greater than the fitness of $P_{best\,kd}(t)$.

8. Then $X_{kd}(t+1)$ is considered as new $P_{best\,kd}(t+1)$

9. Else $X_{kd}(t)$ is considered as $P_{best\,kd}(t+1)$.

10. Evaluate the greatest fitness (G_{best}) by comparing the P_{best} of particles with each other and replace the global best position of the swarm.

11. If fitness of $X_{kd}(t+1)$ is greater than the fitness of $G_{best\ d}(t)$

12. Then $X_{kd}(t+1)$ is considered as new $G_{best\ d}(t+1)$

13. Else $X_{kd}(t)$ is considered as $G_{best\ d}(t+1)$

14 . End

15. The velocity of every particle has been accelerated against its P_{best} and G_{best}. A Random term is used to weigh this acceleration. Every particle's new position in the solution space is determined by summing up the particle's position vector and new velocity value .

Until finding the best weight parameters or optimal solution, the above steps 4-14 are to be continued. The iteration continuous till the best solution is obtained .

Results and discussion

The proposed sentiment analysis on various datasets is implemented using the programming language Python 3.7 in the operating system of windows 2007 with 64 bit and with 4 GB main memory at 2 GHz dual-core PC. In this simulation, various datasets such as Amazon, Trip advisor, demonetization and Books are utilized. From each dataset, 80% of the dataset are taken for training the proposed classifier APSO-LSTM and 20% of the dataset is taken for testing the classifier. Using the training dataset, pre-processing and word embedding are done and the classifiers are APSO-LSTM, LSTM, ANN and SVM are trained. Similarly, pre-processing and word embedding processes are also done on testing dataset. Finally, the embedded word features from the testing dataset are given as input to the trained classifiers. Due to the performance of trained classifiers, the sentiment score of the input tweets is analyzed. After the classification (positive, negative, and neutral) of input tweets, on the basis of precision, recall, accuracy, and F-score, the classifiers' performance is appraised. The following section defines the performance metrics.

Dataset description

For conducting experiments, four datasets are used namely, Amazon reviews, trip advisor, demonetization reviews, and book reviews. Table 1 describes the datasets.

The Amazon dataset includes 60,000 Amazon customer reviews and star ratings. The 60,000 reviews are classified as 25,627 positive reviews, 19,567 negative reviews, and 14,806 are neutral reviews. The Trip advisor dataset includes customer's reviews about 1000 hotels which are delivered by Datafiniti's Business Database. The Trip advisor dataset consists of hotel name, location, review data, title, username, rating, and more. It has 20,000 numbers of reviews out of which 10,000 reviews are positive, 7000 reviews are negative and 3000 reviews are neutral. The demonetization Twitter's dataset with 12,974 tweets is classified as 2974 positive tweets, 4936 negative tweets and 5064 neutral tweets. Besides the books review, dataset contains 213,335 numbers of reviews out of which 177,268, 38,434 and 27,567 numbers of reviews are positive, negative and neutral respectively.

Classifiers

For performance analysis, we evaluate the classifiers using four datasets namely, Amazon reviews, trip advisor, demonetization reviews, and book reviews. The results of these classical sentiment classifiers such as ANN(Artificial neural network), SVM (Support vector machine) and traditional LSTM are compared with our proposed classifier APSO-LSTM.

Artificial neural network: this classifier is very useful in performing various tasks in many applications namely, prediction and recognition. There are three layers mainly composed by ANN. For getting information (data), signs, attributes, or estimations from the external condition, input layer is necessary. These information sources (tests or models) are commonly institutionalized inside the limit esteems conveyed by activation functions (AF). The Hidden layer is called a center layer which is placed between the Input and Output layer. AF applies on a hidden layer in case it is open and the loads in the hidden hub need to test using training information. The hubs presented in the output layer are a dynamic one. This layer is additionally made out of neurons and is responsible for conveying and presenting the last network yields, which result from the preparation performed by the neurons in the past layers.

Support vector machine: the SVM approach Tehranyet et al.[38] aims to divide the hyperplane between classes by taking into account the training instances focused on the class descriptions. support vectors are derived from the training cases. Other training vectors are rejected vectors. The purpose of SVM is to improve the generalization capability by extending classification gaps through a discriminant function. For all linear classification problem, let the training sample set be $\left\{u_i, v_i\right\}$, $(i = 1, 2, ..., m)$. The mathematical expression of the optimal hyperplane is given in Eq. (22).

$$f(u) = \omega . \phi(u) + a, \tag{22}$$

where $a$is the threshold value; ω is the wWeight factor.

The discriminant function of the training sample is given in Eq. (23).

$$f(u) = \mathrm{sgn}\left(\sum_{i=1}^{m} v_i . b_i . r\left(u_i, u\right) + a\right), \tag{23}$$

where b_i is the lagrange multiplier; $a$is the threshold value; u_i, v_i are the two vectors; $r\left(u_i, u\right)$ is the kernel function.

Long short-term memory(LSTM): this neural network model is best suited in various Natural Language Processing such as whole data sequence predictions and prediction over time series data. This classifier is will very suitable for grouping, handling time series predictions. LSTM solves the gradient vanishing problem which arises in RNN. When compared to ANN and SVM, the conventional LSTM classifier attains the best results on the datasets used.

Performance metric

The metrics such as accuracy, recall, precision and F-score are defined with the TP, TN, FP and FN. Where TP denotes

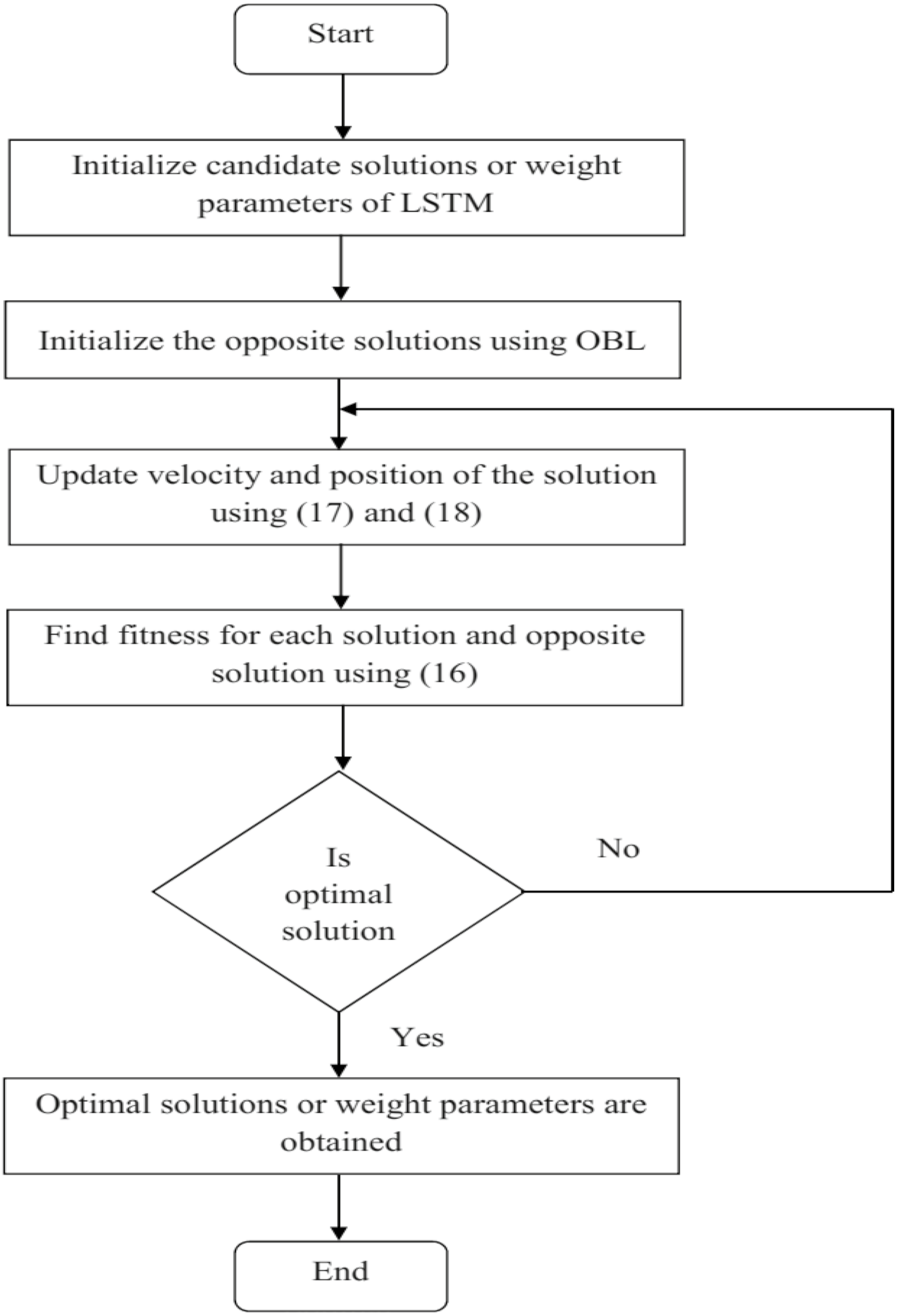

Start
Initialize candidate solutions or weight parameters of LSTM
Initialize the opposite solutions using OBL
Update velocity and position of the solution using (17) and (18)
Find fitness for each solution and opposite solution using (16)
Is optimal solution
No
Yes
Optimal solutions or weight parameters are obtained
End

Fig. 6 Flowchart of the APSO algorithm

True positive, TN denotes True Negative, FP denotes False Positive and FN denotes False Negative. The performance metrics are defined as follows.

$$\text{Accuracy} = \frac{(\text{TN} + \text{TP})}{(\text{TN} + \text{TP} + \text{FN} + \text{FP})}, \quad (24)$$

$$\text{Recall} = \frac{\text{TP}}{(\text{FN} + \text{TP})}, \quad (25)$$

$$\text{Precision} = \frac{\text{TP}}{(\text{FP} + \text{TP})}, \quad (26)$$

Table 1 Description of datasets

Datasets	Total reviews	Total positive reviews	Total negative reviews	Total neutral reviews
Amazon reviews	60,000	25,627	19,567	14,806
Trip Advisor	20,000	10,000	7000	3000
Demonetization Twitter data set	12,974	2974	4936	5064
Books reviews	213,269	147,268	38,434	27,567

$$F\text{ - measure} = \frac{2(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}. \tag{27}$$

Comparative experiments

For demonstrating the proposed technique efficacy, many comparative analyses were undergone. Good selection of feature extraction technique contributes well for better sentimental prediction task.

Selection of parameter for skip-gram word embedding

In Table 2, the Hyper-Parameter choices for word embedding are tabulated below.

Results of different word embedding size

In this paper skip gram algorithm is used for feature extraction process. This model converts the word into vector format. Here, the performance comparison between the N-gram model and proposed skip-gram model is done.

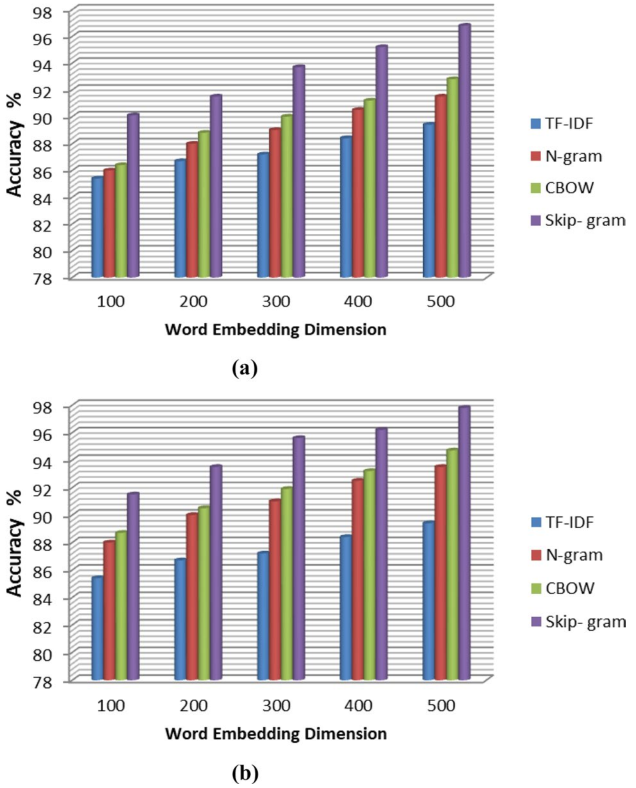
Table 3 shows the comparative analysis of different word embedding methods by different word embedding size for Amazon dataset. As skip-gram model reduces the computational complexity, accuracy percentage of the skip-gram model attained the maximum accuracy of 96.8% which is 92.8% for CBOW based word embedding, 91.5% for

Table 2 Hyperparameters for word embedding

Hyper parameters	Choices
The Architecture	Skip gram
Window size(w)	10
Iteration	30
Vector size	500
Activation function	Hierarchical Softmax
Subsampling rate	1e-3

Table 3 Accuracy for different word embedding methods by varying WE sizes on Amazon Review dataset

WE sizes WE Methods	100	200	300	400	500
TF-IDF	85.4	86.7	87.2	88.4	89.4
N-gram	86	88	89	90.5	91.5
CBOW	86.4	88.8	90	91.2	92.8
Skip- gram	90.1	91.5	93.7	95.2	96.8



(a)

(b)

Fig. 7 **a** Accuracy plot for different feature extraction methods at different Word Embedding dimensions using Amazon dataset. **b** Accuracy plot for different feature extraction methods at different Word Embedding dimensions using Trip advisor dataset

Table 4 Accuracy for different word embedding methods by varying WE sizes on trip advisor dataset

WE Sizes WE Methods	100	200	300	400	500
TF-IDF	85.4	86.7	87.2	88.4	89.4
N-gram	88	90	91	92.5	93.5
CBOW	88.7	90.5	91.9	93.2	94.7
Skip-gram	91.5	93.5	95.6	96.2	97.8

N-Gram based word embedding and 89.4% for TF-IDF based word embedding. The Table 3 corresponding graphical representation is given in Fig. 7a.

In Table 4, results obtained at different word embedding size using Trip advisor dataset is shown, our skip-gram model shows maximum accuracy of 91.5% for size 100 which is 3% better than CBOW based word embedding, 3.9% better then N-Gram model and 6.65 better than TF-IDF based word embedding. Similarly, when the word embedding size is 500, the proposed approach attained the maximum accuracy of 97.8%. Virtual representation corresponding to Table 4 is given in Fig. 7b.

While making predictions, skip-gram model learn better representations for the rare words because there is no averaging of embedding vector. Skip gram based word embedding yields consistent best accuracy in models using large corpora and a high number of dimensions.

Parameter setting

The parameter setting for the sentiment classifier is listed below in Table 5. As shown in the table, the values of hyper-parameters which used in this simulation. The LSTM hidden state is set to 20. Batch size is 32, the learning rate is 0.001, gradient clipping is 5, dropout is 0.5 and max epoch is set t.

Performance analysis of proposed APSO-LSTM at different iterations on Amazon reviews dataset

The goal of the proposed methodology is the prediction of reviewer's opinion using APSO-LSTM algorithm. Evaluation is done on proposed model using Amazon review dataset to assess the performance.

The experimental results obtained by using Amazon dataset is given in Table 6. To enhance the performance of LSTM, APSO algorithm selects the optimal weight parameters. As shown in the Table 6, the maxima accuracy of 96.8%, precision of 85.28%, recall of 76.08% and F-measure of 80.45% attained by the proposed strategy. This is due to optimal weight selection using APSO. The corresponding graphical representation is given in Fig. 8.

Table 5 Parameters used in this work

Hyper parameter	Values
Algorithm	LSTM
Epoch	500
Batch size	32
Optimizer	APSO
Number of layers	3
Loss function	MSE
Learning_ rate	0.001
Dropout_ rate	0.5
Number of hidden unit	20
Training	80%
Testing	20%

Table 6 Performance analysis of Proposed APSO-LSTM at different Iterations on Amazon review dataset

Metrics iterations	Accuracy %	Precision %	Recall %	F-measure %
100th	90.1	80.12	71.25	75.43
200th	91.5	81.16	72.27	76.46
300th	93.7	82.58	73.67	77.87
400th	95.2	83.79	74.92	79.11
500th	**96.8**	**85.28**	**76.08**	**80.45**

Bold values indicate are the results of Proposed system

Comparison of the performance evaluation of different classifier

For demonstrate the proposed technique's efficacy, the presented APSO-LSTM based sentimental analysis compared with different algorithms namely, LSTM, ANN and SVM-based sentimental analysis. Similarly, experimental results analyzed using different four datasets namely, Amazon review dataset, trip advisor dataset, demonstration dataset, and book review dataset. Table 7 shows the comparative analysis.

When analyzing Table 7, for Amazon product review dataset, our proposed strategy acquired the highest accuracy of 96.8%, LSTM model attains 94.1%, 81.9% by using ANN-based sentimental analysis, and 91.8% by using SVM based sentimental analysis. Compared to ANN based sentimental analysis and SVM-based sentimental analysis, APSO-LSTM and LSTM methods attained better results. Similarly, our proposed method attained the higher precision as 85.28%, recall as 76.08% and F-Measure as 80.4%. Figure 9 indicates the comparative results using Amazon product review dataset is given in Fig. 9. For analyzing trip advisor dataset, the highest accuracy of 97.8% acquired by

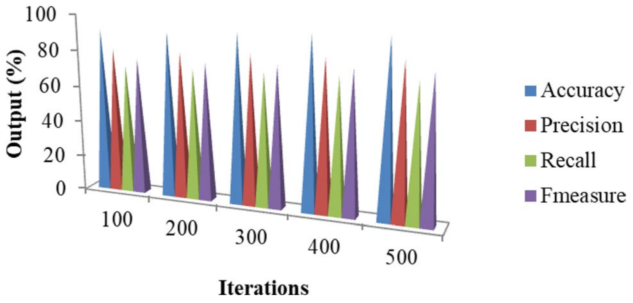


Fig. 8 Experimental results using Amazon dataset at different Iterations

Table 7 Accuracy, recall, precision and F-measure of different classifiers

Dataset	Methods	Accuracy %	Precision %	Recall %	F-measure %
Amazon product review	APSO-LSTM	**96.8**	**85.28**	**76.08**	**80.4**
	LSTM	94.1	83	74.5	78.57
	ANN	81.9	79.5	72	76.8
	SVM	91.8	81.0	73	75.6
Trip advisor	APSO-LSTM	**97.8**	**87.28**	**78.08**	**82.42**
	LSTM	95.1	85	76.5	80.5
	ANN	82.1	81.5	74	77.6
	SVM	92.8	83	75	78.8
Demonetization	APSO-LSTM	**93.2**	**82.3**	**73.5**	**77.4**
	LSTM	91.5	80	71	75
	ANN	79	76.5	69	72
	SVM	88.9	78	70	73.8
Book review dataset	APSO-LSTM	**95.2**	**83.7**	**74.9**	**79.1**
	LSTM	92.8	81.5	72.9	76.9
	ANN	80.2	78	69.5	73.5
	SVM	90.1	80	71	75.2

Bold values indicate are the results of Proposed system

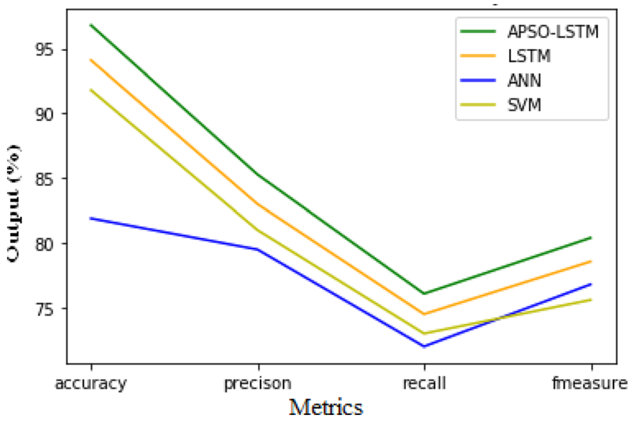


Fig. 9 Comparative analysis using Amazon dataset

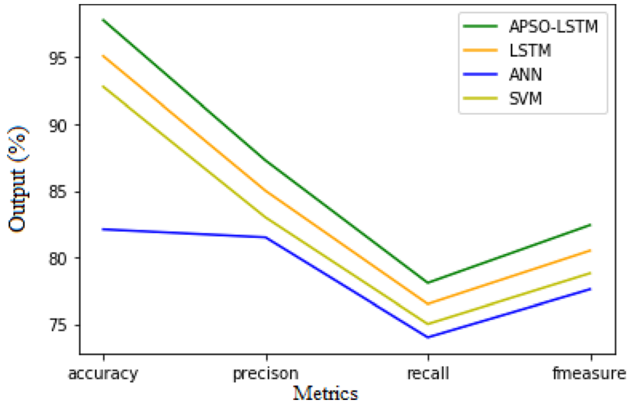


Fig. 10 Comparative analysis using Trip adviser

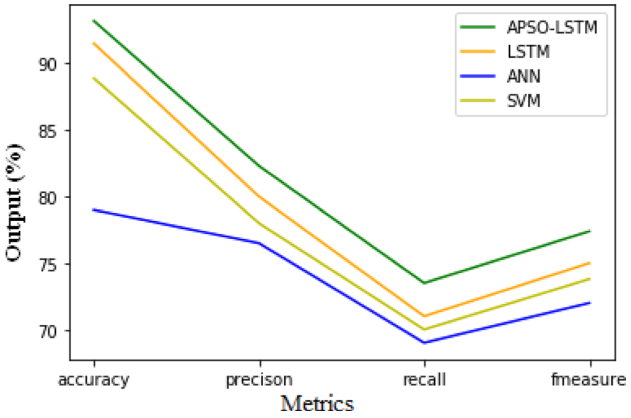


Fig. 11 Comparative analysis using demonetization

proposed method, precision of 87.28%, and recall of 78.08% and F-measure of 82.42%.

Due to skip-gram model and APSO, better results can be achieved. Figure 10 shows the results obtained using trip advisor dataset. For demonization dataset, the proposed strategy got the higher accuracy 93.2%. Similarly, the proposed model attained more precision of 82.3%. Comparative analysis using demonetization review dataset is given in Fig. 11. For analyzing book review dataset, the proposed method attains the maximum accuracy compared to exiting methods. The graphical representation of book review dataset is given in Fig. 12.

LSTM shows superior results over the classic algorithms SVM and ANN due to its long term memory capabilities at predicting text sequence. LSTM solves the gradient vanishing and exploding problems. APSO assists LSTM in

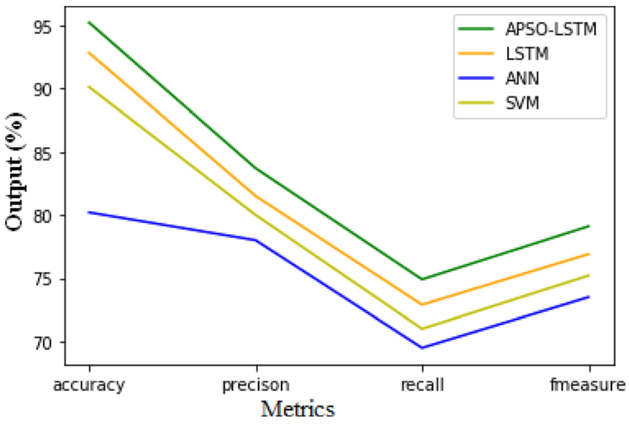


Fig. 12 Comparative analysis using Book review dataset

selecting the best weight for the environment in less number of iterations. The computational complexity reduces when the number of iterations decreases. So APSO-LSTM's ability in selecting optimal weights for neural network combined with good hyper-parameter choices leads to improved accuracy when compared to traditional LSTM. Figure 13 shows the comparison between the computation complexity of the APSO-LSTM and LSTM. As shown in the figure computation time or complexity increases when the number of iterations increases. However, the computation time of the proposed APSO-LSTM is reduced to that of the LSTM as the weight parameters. of the LSTM are optimized using the APSO algorithm.

Figure 14 shows the computational complexity of different classifiers such as APSO-LSTM, PSO-LSTM and LSTM. As the weight parameters of the LSTM are optimized using PSO, computational complexity is reduced than the conventional LSTM as shown in the figure. However, the performance of the PSO-LSTM is further increased by including OBL with the PSO. Thus computation complexity of the APSO-LSTM is decreased than the PSO-LSTM.

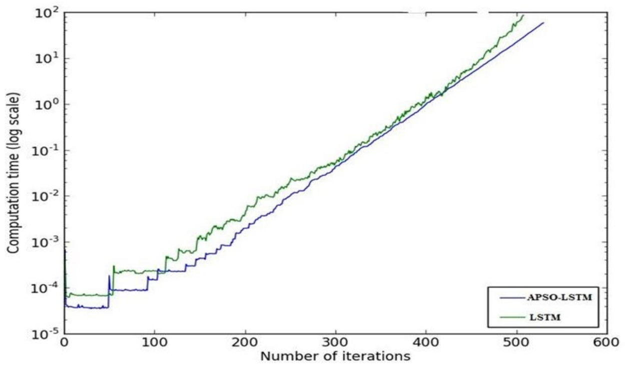


Fig. 13 Computation time of APSO-LSTM and LSTM

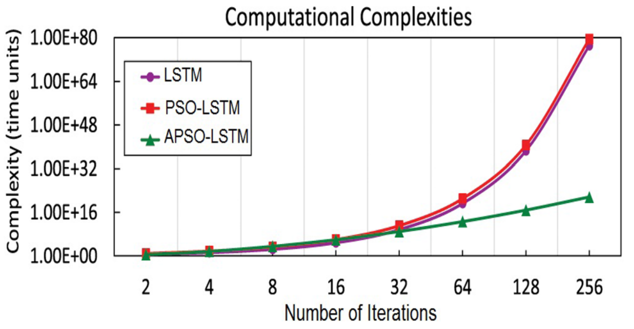


Fig. 14 Computational complexity of APSO-LSTM, PSO-LSTM and LSTM

Conclusion

An efficient sentimental analysis framework is proposed in this work. The optimization problem is solved by adding OBL with the PSO algorithm. The skip gram model based feature extraction has been used for word embedding. Skip-gram Word to Vector representation requires less memory space and yield continuously higher accuracy when compared with other Word to vector representations. LSTM can achieve a good result even without optimization due to its good generalization ability. To further improve the performance of the LSTM, APSO algorithm is used for optimizing the weight parameters of the LSTM. The contribution of APSO in selecting weight parameters for LSTM neural network increases accuracy and decreases computational complexity. For experimental analysis, four types of the dataset have been used. The performance evaluation of the proposed method has been done using different metrics like accuracy, recall, precision, and F-measure. The proposed methodology attained the maximum accuracy of 96.8% for the Amazon dataset, 97.8% for trip advisor dataset, 93.2% for the demonetization dataset, and 95.2% for book review dataset. Empirical results prove the superior performance of the proposed Methodology.

Declaration

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Pozzi FA, Fersini E, Messina E et al (2017) Challenges of sentiment analysis in social networks: an overview. J Sentiment Anal Soc Netw 1:1–11
2. Liu B (2012) Sentiment analysis and opinion mining. Morgan & Claypool, Williston
3. Pang B, Lee L (2008) Opinion mining and sentiment analysis. FNT Inf Retriev 2:1–135
4. Kharde VA, Sonawane SS (2016) Sentiment analysis of twitter data: a survey of techniques. Int J Comput Appl 139(11):5–15
5. Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of the seventh conference on international language resources and evaluation, pp 1320–1326
6. Bifet A, Frank E (2010) Sentiment knowledge discovery in Twitter streaming data. In: Proceedings of the 13th International Conference on Discovery Science. Springer, Berlin, pp 1–15
7. Agarwal, Xie B, Vovsha I, Rambow O, Passonneau R (2011) Sentiment analysis of Twitter data. In: Proceedings of the ACL 2011Workshop on Languages in Social Media. pp 30–38
8. Xia R, Zong C, Li S (2011) Ensemble of feature sets and classification algorithms for sentiment classification. Inf Sci 181(6):1138–1152
9. Apoorv A, Boyi X, Ilia V, Owen R, Rebecca P (2011) Sentiment analysis of Twitter data. In: Proceedings of the Workshop on Languages in Social Media [D], LSM'11 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2011), pp 30–38
10. Suppala K, Rao N (2019) Sentiment analysis using naïve bayes classifier. Int J Innov Technol Explor Eng 8(8):246–269
11. Arora T, Dhawan S, Singh K (2016) Sentiment analysis of online movies' reviews using Improved k-Nearest Neighbor Classifier. Adv Comput Sci Inf Technol (ACSIT) 3(4):241–245
12. Zainuddin N, Selamat A (2014) Sentiment analysis using support vector machine. In 2014 International Conference on Computer, Communications, and Control Technology (I4CT), pp 333–337. IEEE
13. Mikolov T, Karafiat M, Burget L, Cernocký J, Khudanpur S (2010) Recurrent neural network based language model. Interspeech 2:3
14. Legrand J, Collobert R (2016) Deep neural networks for syntactic parsing of morphologically rich languages. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, no.CONF
15. Hemalatha I, Saradhi Varma GP, Govardhan A (2013) Sentiment analysis tool using machine learning algorithms. Int J Emerg Trends Technol Comput Sci (IJETTCS) 2(2):105–109
16. Lv S, Zhang H, He H, Chen B (2020) MicroblogRumor detection based on comment sentiment and CNN-LSTM. J Artif Intell China. https://doi.org/10.1007/978-981-15-0187-6_17
17. Jin Z, Yang Y, Liu Y (2019) Stock closing price prediction based on sentiment analysis and LSTM. Neural Comput Appl
18. Xianghua Fu, Yang J, Li J, Fang M, Wang H (2018) Lexicon-enhanced LSTM with attention for general sentiment analysis. IEEE Access. https://doi.org/10.1109/ACCESS.2018.2878425
19. Guixian Xu, Meng Y, Qiu X, Ziheng Yu, Wu X (2019) Sentiment analysis of comment texts based on BiLSTM. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2909919
20. Latif M, Qamar U (2019) A novel ensemble approach for feature selection to improve and simplify the sentimental analysis. In: Intelligent Computing-Proceedings of the Computing Conference. Springer, Cham, pp 573–592
21. Graber F, Kallumadi S, Malberg H, Zaunseder S (2018) Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In: Proceedings of the 2018 International Conference on Digital Health, pp 121–125
22. Da'u A, Salim N (2019) Sentiment-aware deep recommender system with neural attention networks. In: 2169–3536 2019 IEEE. Translations and content mining, vol. 7, https://doi.org/10.1109/ACCESS.2019.2907729
23. Sailunaz K, Alhajj R (2019) Emotion and sentiment analysis from Twitter text. J Comput Sci. https://doi.org/10.1016/j.jocs.2019.05.009 (**Published by Elsevier Ltd**)
24. Ahamed S, Danti A (2018) Effective emoticon based framework for sentimental analysis of web data. In: International Conference on Recent Trends in Image Processing and Pattern Recognition. Springer, Singapore, pp 622–633
25. Shaukat Z, AhadZulfiqar A, Xiao C, Azeem M, Mahmood T (2020) Sentiment analysis on IMDB using lexicon and neural networks. SN Appl Sci 2(2):1–10
26. Ankita, Saleena N (2018) An ensemble classification system for Twitter sentiment analysis. In: Peer-review under responsibility of the scientific committee of the International Conference on Computational Intelligence and Data Science. Published by Elsevier Ltd., https://creativecommons.org/licenses/by-nc-nd/3.0/
27. Li W, Qi F, Tang M, Yu Z (2020) Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. J Neuro Comput 387:63–77
28. Alqaryouti O, Siyam N, Monem A, Shaalan K (2019) Aspect-based sentiment analysis using smart government review data. Appl Comput Inform. https://doi.org/10.1016/j.aci.2019.11.003
29. Ray P, Chakrabarti A (2019) A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis. In: Production and hosting by Elsevier B.V. on behalf of King Saud University, pp 2210–8327. https://doi.org/10.1016/j.aci.2019.02.002
30. Akyol S, Alatas B (2020) Sentiment classification within online social media using whale optimization algorithm and social impact theory based optimization. Phys A 540:123094
31. Yang SY, Mo SYK, Liu A, Kirilenko AA (2017) Genetic programming optimization for a sentiment feedback strength based trading strategy. Neurocomputing 264:29–41
32. Keshavarz H, SanieeAbadeh M (2017) ALGA: adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs. Knowl-Based Syst 122:1–16
33. Marcacini RM, Rossi RG, Matsuno IP, Rezende SO (2018) Cross-domain aspect extraction for sentiment analysis: a transductive learning approach. Decis Support Syst 114:70–80
34. Meškelė D, Frasincar F (2020) ALDONAr: a hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and a regularized neural attention model. Inf Process Manag. https://doi.org/10.1016/j.ipm.2020.102211 (**Published by Elsevier Ltd**)
35. Liang W, Feng R, Liu X, Li Y, Zhang X (2018) Gltm: a global and local word embedding-based topic model for short texts. In: IEEE. Translations and content mining, VOLUME 6, 2018 , date of current version August 28, 2018. 2169–3536. IEEE, https://doi.org/10.1109/ACCESS.2018.2863260
36. Hu Y, Sun X, Nie X, Li Y, Liu L (2019) An enhanced LSTM for trend following of time series, Xuzhou 221116, China. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2896621
37. Wang P, Zhao J, Gao Y, Sotelo MA, Li Z (2020) Lane work-schedule of toll station based on queuing theory and PSO-LSTM model. IEEE Access. https://doi.org/10.1109/ACCESS.2020.2992070
38. Tehrany MS, Pradhan B, Mansor S, Ahmad N (2015) Flood susceptibility assessment using GIS-based support vector machine model with different kernel types. Catena 125:91–101. https://doi.org/10.1016/j.catena.2014.10.017