Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
June 2023, vol. 28, no. 1, pp. 137–147
https://doi.org/10.2478/acss-2023-0013
https://content.sciendo.com

Multichannel Approach for Sentiment Analysis Using Stack of Neural Network with Lexicon Based Padding and Attention Mechanism

Venkateswara Rao Kota[1*], M. Shyamala Devi[2]
[1,2] *Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, T.N., India*

Abstract **– Sentiment analysis (SA) has been an important focus of study in the fields of computational linguistics and data analysis for a decade. Recently, promising results have been achieved when applying DNN models to sentiment analysis tasks. Long short-term memory (LSTM) models, as well as its derivatives like gated recurrent unit (GRU), are becoming increasingly popular in neural architecture used for sentiment analysis. Using these models in the feature extraction layer of a DNN results in a high dimensional feature space, despite the fact that the models can handle sequences of arbitrary length. Another problem with these models is that they weight each feature equally. Natural language processing (NLP) makes use of word embeddings created with word2vec. For many NLP jobs, deep neural networks have become the method of choice. Traditional deep networks are not dependable in storing contextual information, so dealing with sequential data like text and sound was a nightmare for such networks. This research proposes multichannel word embedding and employing stack of neural networks with lexicon-based padding and attention mechanism (MCSNNLA) method for SA. Using convolution neural network (CNN), Bi-LSTM, and the attention process in mind, this approach to sentiment analysis is described. One embedding layer, two convolution layers with max-pooling, one LSTM layer, and two fully connected (FC) layers make up the proposed technique, which is tailored for sentence-level SA. To address the shortcomings of prior SA models for product reviews, the MCSNNLA model integrates the aforementioned sentiment lexicon with deep learning technologies. The MCSNNLA model combines the strengths of emotion lexicons with those of deep learning. To begin, the reviews are processed with the sentiment lexicon in order to enhance the sentiment features. The experimental findings show that the model has the potential to greatly improve text SA performance.**

Keywords **– Attention mechanism, Bi-LSTM, convolution neural network, lexicon, multichannel approach, padding, sentiment analysis, stack of neural network.**

I. Introduction

Textual sentiment analysis has become more difficult as social media platforms like Twitter and Facebook continue to rise in popularity. As a result, there has been a lot of research done on determining the tone of texts [1]. CNNs and LSTM are two examples of recent deep learning models that have shown promise in sentiment analysis. These models have already shown that they can handle sequences of arbitrary lengths. Yet, when implemented at the feature extraction layer [2], they attribute equal value to different features, have a highly dimensional feature distance, and work with sparse text input. In a rudimentary CNN, a local n-gram is convolved with a set of kernels [3]. To reduce the computational complexity of CNN, max pooling is used to reduce the amount of output from the previous convolution. In the realms of text, strings, and sequence data, recurrent neural network (RNN) is the current gold standard. To better represent texts containing sequence order data [4], the RNN architecture uses a complicated technique to consider the information from prior words. Training a recurrent neural network is challenging because of vanishing and growing gradient problems [5]. When it comes to maintaining long-term trust, they perform better than standard RNNs. Layer upon layer can be added to these already deep NN models to make them even more capable [6]. In order to accommodate both local n-gram features and long dependencies, this model combines a CNN layer, a max-pooling layer, and an LSTM layer. Instead of breaking up the feature space into smaller regions, this layer applies LSTM to the feature map sequence generated by the max-pooling layer [7].

Finding adjectives, which are strong indicators of opinion, is a part of the speaking task. Opinion words and phrases frequently use terms that reflect emotions, such as good or terrible, like or dislike. Nonetheless, there are also utterances that convey opinions without explicitly employing opinion terms [8]. The usage of negative words can have an effect on a person's outlook when good is associated with bad. The study of this area is gaining interests, with researchers from both academia and industry taking an interest. The purpose of SA is to gain insight from people's online writings by analysing the underlying emotions. In recent years, SA has become a hot issue in the realm of NLP [9] due to its varied corporate and academic applications and the proliferation of social networks. As a result, numerous methods have been presented as a means of determining a document's polarity [10]. Most SA applications rely heavily on polarity detection, a binary categorization problem. When it comes to polarity categorization, most prior SA methods relied on training

[*] Corresponding author's e-mail: kvenkateswarar6@gmail.com

shallow methodologies with carefully crafted efficient features [11]. To classify linguistic qualities, such as lexical features, part-of-speech (POS) tags, and n-grams, these models sometimes use Naive Bayes, support vector machines (SVM), and latent Dirichlet allocation (LDA) [12].

A neural language model that takes into account numerous lexical relationships produces a dense real-valued vector called a word embedding [13]. This has led to widespread use of word embedding as the input of DNN in current NLP works [14]. The implementation of tasks involving machine learning or learning embeddings, such as categorization and grouping, has been proposed as the primary goal of using DNNs for evaluating text data. While there is a vast variety of deep networks, those most commonly used in text processing studies are RNNs and CNNs [15]. CNN models are given preference because of their ability to pick up on regional patterns, while RNNs have proven their worth in sequential modelling. While RNNs [16] find value in a variety of text processing tasks, they struggle with vanishing and exploding gradients, especially when dealing with input data that have long dependencies. Most NLP methods, especially those used in the field of SA, rely heavily on these kinds of dependencies [17].

There are two main categories of textual content on the web: facts and opinions. Statements about reality that lack emotional nuance are called facts [18]. Much prior work with online texts has concentrated on factual data for use in a variety of NLP applications as informing retrieval, text categorization, etc. Research on comprehending subjective expressions is limited at present because of the multiple challenges associated with the topic. Research into SA has become increasingly popular over the past decade as its many practical implications have become increasingly apparent in both industry and society at large [19]. Opinion mining, also known as sentiment analysis, is a field of research that delves into how readers feel about products, services, and other topics mentioned in the text. Automatically analysing web content to create an opinion calls for a system with a deep comprehension of natural language [20]. There are three different levels at which SA can be studied: documents, sentences, and features/aspects. The SA of a review document indicates if the polarity is positive or negative [21]. The document is handled as if it were a single unit. Sentence-level SA looks at a single sentence to determine the attitude or viewpoint expressed [22]. Sentiment analysis based on documents or individual sentences does not work. Aspect type SA refers to methods used to locate the textual elements where the sentiment is expressed. Types in sentiment analysis based on text are shown in Fig. 1.

LSTM, which can store long dependencies, was developed to address the aforementioned issue in SA. Many academics in the field of NLP have taken an interest in LSTM because of its potential to solve the issues plaguing RNNs. In order to merge the forward hidden layers and the backward hidden layers, the bidirectional LSTM (Bi-LSTM) model was proposed. This model is able to solve the problem of sequential modelling. Many natural language processing tools make use of bi-LSTM. Layers of recurrent neural networks (RNNs) can learn long-term dependencies between time steps in time series or sequence data in both directions using a bidirectional LSTM (BiLSTM) layer. When the RNN has to learn from the entire time series at each time step, these dependencies can be helpful. Bidirectional training involves running your inputs in two directions, one from the past into the future and the other from the future into the past. This strategy differs from unidirectional training in that the future is preserved in the LSTM that runs backward, and the two hidden states can be used together at any time to preserve both the past and the future. The Bi-LSTM model is shown in Fig. 2.

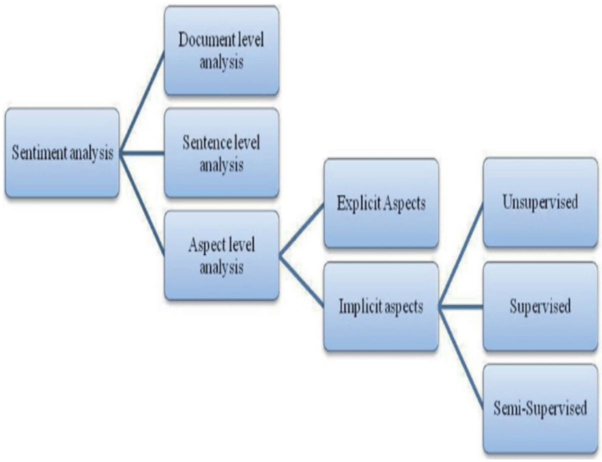

Fig. 1. Types of SA based on text.

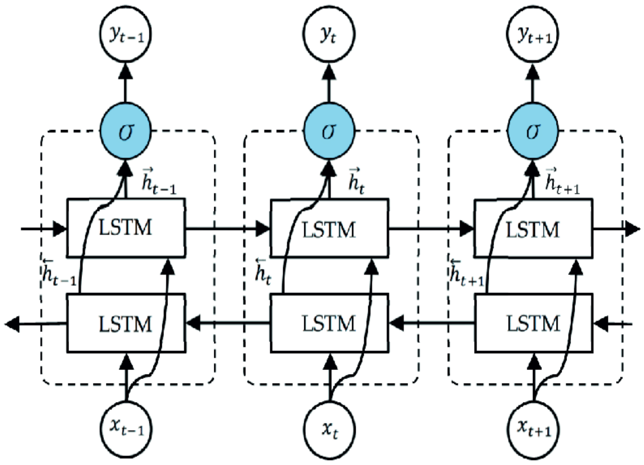

Fig. 2. Bi-LSTM model.

Unfortunately, this approach has two significant flaws: First, the model becomes more complex and difficult to improve due to the high-dimensional input distance common in text processing applications [23]. Second, the model is unable to zero down on the crucial elements of the contextual information. CNNs have been used to reduce the dimensional feature distance and extract meaningful patterns from the text. The attention mechanism provides different emphasis weights for different elements of the context. This study proposes multichannel word embedding and employing stack of neural networks with lexicon-based padding and attention mechanism (MCSNNLA) method for SA.

II. Literature Review

Online shopping has become increasingly common as Internet connectivity and related technologies have improved over the past few years. Large-scale sentiment analysis of shopper feedback on e-commerce sites has been shown to effectively boost customer happiness. Based on the sentiment lexicon and a combination of CNN and attention-based bidirectional gated recurrent unit, a new model for sentiment analysis, dubbed SLCABG, is proposed by Yang et al. [1]. The SLCABG model is an improvement over pre-existing sentiment analysis models of product reviews because it combines the benefits of a sentiment lexicon with deep learning technology. The SLCABG model incorporates the best features of both the sentiment lexicon and deep learning. To begin, the author boosted the review sentiment features using the sentiment lexicon. Then, the attention mechanism was applied to weight the extracted primary sentiment features and context features from the reviews using the CNN and the GRU network. Finally, the sentiment weighted features were categorized. For the sake of training and testing Chinese-based models, this paper scans and cleans the real book assessment of dangdang.com, a popular Chinese e-commerce website.

An essential component of the sentiment analysis process is the sentiment lexicon, which contains sentiment information for words. At the moment, most sentiment lexicons only allow one polarity per word and do not take into account emotional nuance. In this study, Yin et al. [2] suggested constructing the sentiment lexicon using context-dependent POS chunks, or CP-chunks, with the goal of resolving lexical sentiment ambiguity. Since the polarity and intensity of words change depending on their context, the authors used CP-chunks as the basic measuring stick for sentiment analysis. The task of text sentiment categorization was used to assess this approach.

Identifying, extracting, and organising feelings from user-generated documents or phrases have new potential and challenges as sentiment-rich resources such as blogs and online reviews rise in availability and popularity. Many recent studies have used lexicon-based techniques or supervised learning algorithms to execute sentiment analysis tasks in isolation; however, these methods either do not account for the context of sentences or the sentiment information encoded in the words themselves. Huang et al. [3] proposed a new model called Sentiment Convolutional Neural Network (SentiCNN) to analyse the sentiments of sentences by combining contextual and sentiment information of sentiment words, where relevant knowledge was managed to capture from embeddings and sentiment details were identified using existing lexicons. By enhancing the relationship between characteristics of both phrases and their sentiment words, this model is able to adaptively mix sentiment and context information from sentences. Additionally, in order to uncover the most crucial markers of sentiments and create more accurate predictions using SentiCNN model, the authors proposed three lexicon-based attention mechanisms (LBAMs).

Classifying sentiment at the aspect level is a fine-grained task in the field of sentiment analysis. Several deep learning network-based categorization models for aspect term-sentence relationships have been presented in recent years as a result. Existing techniques rely on end-to-end deep neural network models; however, they are rigid and exclude sentiment word information. In light of this, Ren et al. [4] proposed a bidirectional long short-term memory based lexicon-enhanced attention network (LEAN). In addition to identifying sentiment words, LEAN can zero in on the most pertinent details of a text. In addition, the model adaptability and durability will be improved by using lexical data.

Learning a sentiment lexicon is crucial for sentiment analysis. Sentiment lexicon learning has a significant obstacle in the form of their domain-specific behaviour. Knowledge transfer from one domain to another via a sentiment lexicon is an unsolved research challenge. In this research, Sanagar et al. [5] provided a transfer learning technique to overcome this difficulty by generating novel learning insights across diverse domains within the same genre. The authors presented a method for learning sentiment dictionaries without supervision, which could be used to completely new areas within the same genre. The technique and incremental learning approach learn polarity seeded words from various dynamically selected source domains using a corpus-based approach. Then, it applies what it has learned about genres from the corpus to the new areas. Sentiment lexicon generation for several target domains of the same genre uses the seed words learnt from the corpus. Latent semantic analysis is the foundation of the sentiment lexicon learning process, which draws on unlabelled source and target domain training data. Twenty-four domains of the same genre in this case, reviews of consumer products were used in the study. When compared to competing baselines, the suggested model outperforms them on all of the widely-used assessment metrics.

Sentiment analysis is widely employed in a variety of sectors, including politics, journalism, and more, making it a major area of study in the field of natural language processing. Sentiment analysis relies heavily on word embeddings. Current sentiment embedding techniques simply incorporate sentiment lexicons into a more conventional word representation. Inaccurate sentiment information for a word in multiple contexts is not provided by this sentiment representation approach because it can only distinguish the sentiment data between various words. Wang et al. [6] suggested using the concept of sentiment to address the issue at hand. The authors began by using the context of words to determine the best sentiment notion in Microsoft Concept Graph. In order to achieve accurate embedding of sentiment information and supply more accurate meanings and sentiment representation for words, the authors first created a multi-semantics sentiment intensity lexicon, from which they retrieved the sentiment information of words under an ideal sentiment concept. Eventually, they achieved a more complete word representation by combining two improved word embedding techniques.

Conventional web crawling methods involve initially saving copies of all crawled pages in databases. As the only way to identify sentiment words is to scan all of the stored web pages, this method can end up storing unneeded pages and takes more time to execute when building a sentiment dictionary for a certain domain. On et al. [7] offered two hash-based approaches to implement sentiment-aware web crawling in order to solve these issues. Both use a hash join algorithm; however, one is a bucket-sorted variant. For effective sentiment-aware web crawling, the authors suggested a novel bucket-sorted hash join. The experiments demonstrate that the suggested web crawling approach employing bucket-sorted hash join greatly reduces both running time and storage space, making it superior to existing web crawling methods.

There are several text mining uses where sentiment analysis plays a crucial role. There is a plethora of approaches to sentiment classification that have been offered in the literature. Most current approaches presume that a high-quality training set is available. Yet, in practical settings, it can be difficult to build a training set with precise labels. This is due to the fact that annotating text samples is subjective and typically involves dealing with sophisticated representations of sentiment. In this research, Wu et al. [8] took a different approach to labelling and building a sentiment classifier using a two-tier long short-term memory network to overcome this difficulty. Traditional research has made use of lexical cues for their value in sentiment analysis. Sentiment analysis relies heavily on polar and negative words, for instance. To address the limitations of one-hot encoding and successfully incorporate informative lexical signals, a novel encoding approach, called hot encoding, is presented. Furthermore, due to label noise or context, the emotional polarity of a word may shift across sentences. To model the polar reversal of words, a flipping model is presented. This labelling strategy and recommended methods inform the creation of three Chinese language datasets.

IMDB dataset is considered and positive and negative reviews are analysed. There are 50 000 movie reviews in the dataset, with 25 000 reviews each for good and negative reviews. There are 110 870 distinct words in total. The text datasets for SA are shown in Table I.

TABLE I
TEXT DATASETS FOR SA

Dataset	Description	Language	Total Reviews	Positive	Negative	Unique Words
eRezeki	Digital worker perceptions	English and Malay	4316	2356	1960	3475
Yelp	Hotel and restaurant reviews	English	598 000	24695	351050	313 404
AMAZON	Product reviews	English	296 337	158 230	138 107	134 758
IMDB	Movie reviews	English	50 000	25 000	25 000	110 870

Participants in the financial markets constantly watch financial and economic news. The efficient market hypothesis states that current stock prices accurately reflect all available information and that any newly available information is immediately factored into the pricing of stocks in the future. Hence, traders, portfolio managers, and investors must quickly extract positive or negative attitudes from news in order to make sound investment decisions. Extracting useful signals from the news can be made much easier with the help of sentiment analysis models. Yet, it is difficult to do financial sentiment analysis due to the absence of big labelled datasets and the use of domain-specific language. Applying a generic paradigm of sentiment analysis to a highly specialized field like finance yields little to no useful results. In order to address these issues, Mishev et al. [9] developed an evaluation platform to compare and contrast different sentiment analysis algorithms that use different combinations of text representation strategies and machine-learning classifiers. Using publicly released datasets annotated by financial experts, the authors conducted over a hundred experiments. The authors began the evaluation with finance-oriented lexicons and expanded it to encompass word and sentence encoders, all the way up to the most recent state-of-the-art NLP transformers.

Words and phrases are used by social media users to express their thoughts and feelings. Nonetheless, some people employ caustic, humorous, or metaphorical idioms or proverbs to leave a more lasting impression or grab the reader's attention. Idioms and proverbs are examples of metaphorical language that cannot be taken at face value because of the unity of their underlying themes. Previous study suggested expanding IBM's Emotion Lexicon of Idiomatic Phrases to include over 9000 idioms; both lexicons are manually annotated by a crowd sourcing tool. To circumvent the need for human annotation of idioms, Tahayna et al. [10] presented a knowledge-based expansion strategy in this study. The proposed method has the advantage of not requiring any fine-tuning for the BERT model, making it particularly well-suited for sentiment categorization. The results of these experiments demonstrate that sentiment classifiers benefit greatly from automated idiom enrichment and annotation.

Even though GloVe has demonstrated better results on similarity and evaluation tasks than Word2Vec, according to the authors, this has not been empirically shown, and the usage of either can lead to superior outcomes: both are worth attempting. The numerous word embedding methods are illustrated in Table II.

The bidirectional LSTM hidden state is the concatenation of the forward and backward hidden states at each time step, capturing both past and future information. Self-attention LSTM: The self-attention mechanism is typically used to improve LSTM or GRU performance by automatically learning the contribution of each hidden state and assigning more weight to high contributors [11]. Tree-LSTM: Tree-LSTM employs a recursive LSTM unit to construct structured representations from predefined parsing trees. This model should be able to capture the structured data that influence the subsequent prediction. CNN-LSTM: To create a CNN-LSTM model, this method stacks a CNN layer, a max-pooling layer, and an LSTM layer in that order [12]. As this model does not separate regions, LSTM is applied to the sequence of feature maps produced by the max-pooling layer [13]. The state of art on stack of neural network for SA models is presented in Table III.

TABLE II

WORD EMBEDDING METHODS

Word Embedding	Description	Limitations
One hot encoding	Each word in the given text data is written as a vector with only 1 and 0 values. A hot vector is a vector with just 1 and 0 members.	Does not capture relation between the words.
FastText	FastText is a very fast NLP library created by Facebook. FastText operates at a more granular level with character n-grams.	Computationally inefficient.
Word2vec	Word2Vec learns vectors only for complete words found in the training corpus. It is based on word information from the native language. Training parameters for new languages cannot be shared.	Word2Vec struggles with words that are not in the dictionary. For the network to converge, it needs a comparatively larger corpus.
GloVe	The goal of Glove is to derive the relationship between the meaning of a word and the general structure of the corpus.	The glove provides less weight too frequently occurring word pairs so that meaningless stop words like "the" and "an" do not dominate the training process.

TABLE III

STATE OF ART ON STACK OF NEURAL NETWORK FOR SA

Ref.	Neural Network stack	Datasets	Optimizer	Accuracy in %
[10]	Bi-LSTM+ FFNN	Twitter	Minimum batch gradient descent	91.78
[11]	CNN+ LSTM	SST, FB, EB, CVAT	Adam	90.5
[12]	CNN + Bi-LSTM + Attention (SLCABG)	book reviews	Adam	91.9
[13]	CNN+ Bidirectional GRU + Attention	SST	Minimum batch gradient descent	91.5
[14]	CNN+RNN	IMDB	Adam	91.4
[15]	RNN+ Attention	SST	Adam	89.56

III. PROPOSED MODEL

When it comes to NLP, sentiment analysis is one of the most prominent research areas due to its importance in text classification. Aspect-level analysis [24], sentence-level analysis, and document-level analysis are the three tiers at which sentiment can be analysed. User reviews and complaints have become increasingly commonplace since the advent of social media and other user-centric platforms. Automatically sorting text documents into those with the desired polarity is another consideration in the field of sentiment classification. Any of positive, negative, or neutral polarity is possible. Opinion, assessment, attitude, and emotion towards entities and their attributes are all subject to analysis in the discipline of text classification, of which sentiment analysis is a subfield. The ability to extract and analyse public mood and perspectives, as well as obtain business information and make better decisions has made sentiment analysis a significant tool for enterprises, governments, and scholars.

As a subsidiary task of SA, emotion classification at the sentence level has attracted attention from academics and businesses alike. In order to evaluate the polarity or intensity of a sentence-level sentiment, sentiment classification attempts to extract document properties and capture sentence semantic relationships. This kind of sentiment analysis has several applications, including online stores and social media monitoring. The input sequence is better described and noise from various word representations is removed with the multichannel embedding layer conversion to a semantic and sentiment embedding tensor. In order to investigate additional contextual semantics information and integrate multi-scale contextual characteristics, the detachable dilated convolution module, which uses variable dilation rates and fewer parameters, provides ideal feature extraction capacity.

In order to create lexical embeddings, scores from several lexicon datasets are combined. Keys in each lexical dataset are words, and values are collections of sentiment ratings for those words. These ratings go from −1 to 1, with −1 and 1 representing the extremes on both ends of the scale. Other dictionaries, however, include non-probabilistic values, which are standardized to [−1, 1]. The use of a lexicon is one method of performing semantic analysis. Using the semantic orientation of lexicons, this method determines the overall sentiment orientations of a document or set of sentences. It is possible to have a positive, negative, or neutral semantic orientation. Texts are analysed for their emotional tone by using a so-called valence dictionary to assign labels to individual words. Figure 3 depicts the lexicon incorporation into the CNN model.

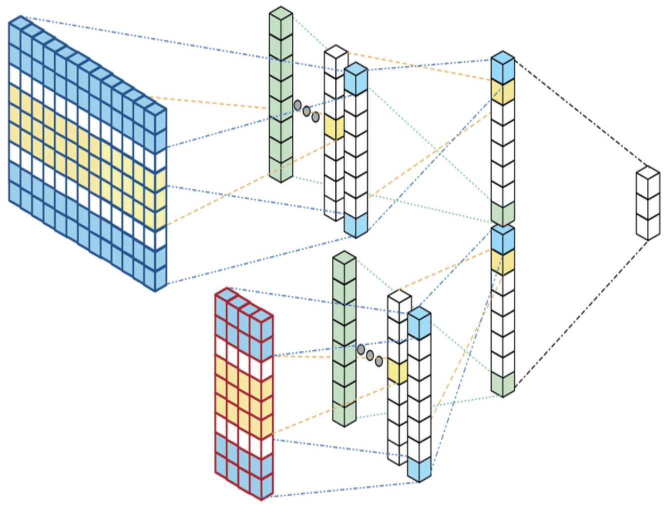

Fig. 3. Lexicon incorporation into CNN.

A lexical embedding is built by summing all the scores across the lexicon datasets for the word $w \in W$, where W is the union of all the words in the lexicon datasets. When w is missing from a data set, a value of 0 is substituted. The final embedding is a vector, v ∈ R e, where e is the sum of all scores in the lexical datasets. In this method, lexical embeddings are encoded separately from the word embedding channel, and the two channels are convolved [25]. To make up for the fact that lexicon embeddings are much lower-dimensional than word embeddings, zeros are appended to the lexicon embeddings to make their dimensions equal to those of the word embeddings. As the two channels have the same shape, multichannel convolution can be applied to the source file [26].

Several convolutions and max pooling are employed to incorporate all of the document matrix embeddings into the final EAV. Initially, it is necessary to convolve the document matrix s ∈ R n x d m times with a filter of length 1 [27]. The document matrix for lexical embeddings has dimension R n e. Then an attention matrix sa ∈ R $n \times m$ is created, where n is the total number of words in the document and m is the total number of filters with length 1 [28]. The attention vector va ∈ R n is created by performing max pooling on each row of the attention matrix sa. To obtain the embedding attention vector ve∈R×d, the document matrix s is transposed so that sT∈Rd×n, and then this result is multiplied by the attention vector ve∈Rd.

We use a bidirectional version of BiLSTM to obtain information from both directions for words. The bidirectional LSTM contains the forward LSTM(f), which reads the sentence S_i from w_i to w_{it}, and the backward LSTM(b), which reads from wit to w_i. The BiLSTM model maps each word embedding x_{it} to a pair of hidden vectors H(v). Different parameters are used for the forward LSTM and backward LSTM. These hidden vectors are the composition of sentence embedding s_i, and used as features for calculating attention weights. Deep learning models can be enhanced with an additional layer of neural networks known as an attention mechanism. By giving distinct pieces of the input different weights, they let the model zero in on specific information. The importance of various aspects of the input to the goal at hand is usually used to determine how heavily they should be weighted. The goal of the attention mechanism was to provide the decoder some leeway in picking out and using the most important pieces of the input sequence through a weighted combination of all the encrypted input vectors, with the most pertinent vectors being given the highest weights. The Bi-LSTM model is shown in Fig. 4.

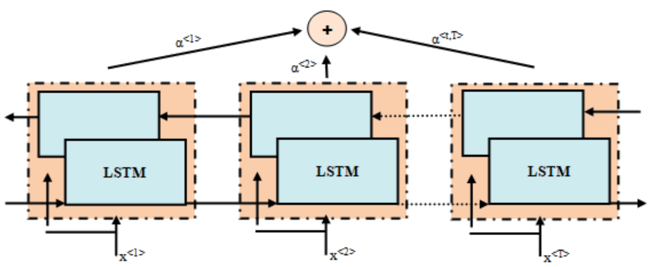

Fig. 4. Bi-LSTM network with the attention mechanism.

Epochs stand for the total number of iterations used in the training dataset. The ability of the model to generalize improves as the number of epochs grows. However, an overfitting problem can quickly arise if the number of iterations is too big, reducing the model generalization potential. The proposed MCSNNLA model phases are shown in Fig. 5.

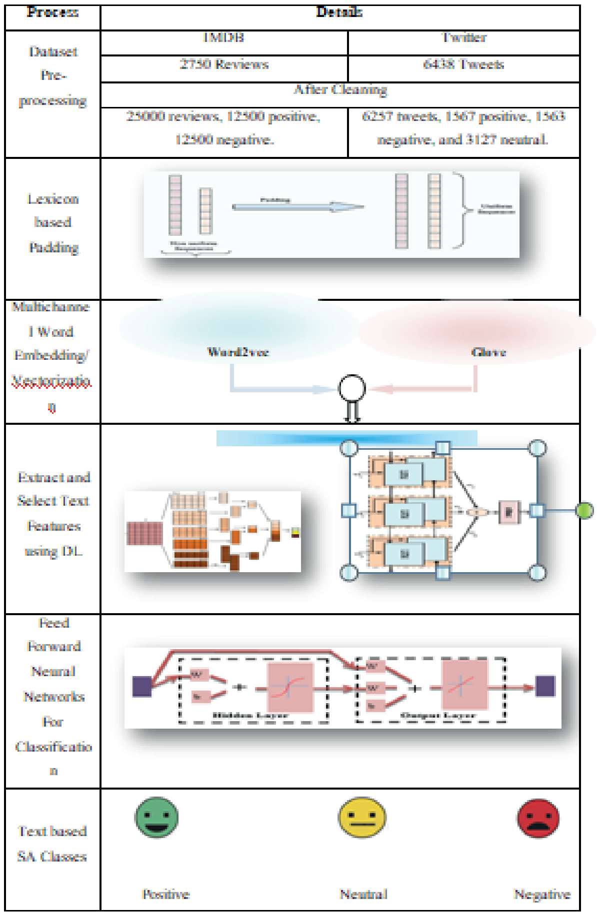

Fig. 5. The proposed MCSNNLA approach for SA.

Algorithm MCSNNLA

{

Input: IMDB, Twitter Datasets {Dset}

Output: Text based Sentiment Classes Prediction Set {Pset}

In order to make mining of text data, deep learning, and other data science processes more efficient, preprocessing is performed. The process of text preprocessing is cleaning and standardizing text before feeding it into a model for analysis and learning. Some text preprocessing methods are generic and can be used in a wide variety of programs, while others are task-specific. The IMDB, Twitter Datasets are considered for text analysis and text classification. The preprocessing is made to clean the dataset to perform padding and feature selection. The preprocessing is performed as

$$ProSet[M] = \sum_{i=1} \frac{\mathrm{Dset}(i) + \mathrm{mean}\big(\mathrm{Dset}(i), \mathrm{Dset}(i+1)\big) + \mathrm{std}(i, i+1)}{\mathrm{size}(\mathrm{Dset})} - \mathrm{drop}(\delta)$$

mean() is used to find the mean values of two consecutive text words and std() is used to find the standard deviation of text attributes. δ is the null attributes in the set that is removed from the dataset using drop() model. Size() is used to detect the dataset size. Here i is the current index and i+1 is the next index to consider to perform mean and standard deviation operations.

The difficulty of text classification is greatly increased by the presence of sequences of varying lengths. Sentences and messages in text datasets like Stanford Sentiment Treebank can range in length from a single word to well over a hundred. When the CNN model is employed for text classification, this becomes a major problem. In this light, text classification difficulty is handled via the padding process, which is commonly used to cushion the border in image classification. The sentiment padding method adds lexical information about an author's intended tone to preexisting features of a text. It is required to pad or reduce some sequences for the purpose to make every combination in a batch fit a specified standard length, which arises from the requirement to encode sequence information into continuous batches. The lexicon based padding is performed as

$$LexPad[M] = \sum_{i=1}^{M} \frac{len(R)}{ProSet(i)} + \sum_{i \in ProSet} \frac{\mathrm{maxrange}(i, i+1) - \mathrm{minrange}(i, i+1)}{\mu} + \prod_{i=1}^{M} \frac{(\mu + \mathrm{mean}(R))}{\mathrm{size}(i)}$$

Here R is the message considered; μ is the padding bits, size() is the model for length calculation of the records in a message. Maxrange() considers the maximum values and minrange() considers the minimum attributes.

Mathematical representations of words can be created using word embedding techniques. The Glove model makes use of the full corpus worth of word-to-word co-occurrence counts. In contrast, word2vec makes use of co-occurrence in the immediate context. Word Embeddings, also known as word vectorization, is a technique used in NLP to convert words and phrases into a numerical vector that may then be used to make predictions about the words and their meanings. Vectorization is the process of translating text into numerical form. The vectorization process is performed as

$$Vector(LexPad[M]) = \sum_{i=1}^{M} \frac{\max\,(LexPad(i)) + \sum_{i=1}^{M} \mu + \mathrm{mean}(i, i+1)}{\mathrm{size}(ProSet)}$$

μ is the attributes with normalized values and mean() is used to find the mean values among the current and next attribute.

Feature selection is a technique used in sentiment analysis to narrow down the number of features to analyse in order to improve the model accuracy in three key areas: first, computational cost; second, overfitting; and third, classification accuracy. Assisting classifiers in performing better and reducing the processing load by restricting the feature set are two key benefits of feature selection for developing successful and effective sentiment analysis applications. The feature selection process is performed as

$$FeatSet[M] = \sum_{i \in Vector[M]} \frac{\max\,(\mathrm{corr}\big(\mathrm{Vector}(i, i+1)\big) + \min\,(\mathrm{getattr}(\mu, i)}{\mathrm{size}(\mathrm{Vector})}$$

Corr() mode is used to identify the correlation among the features. The correlation factor is used to detect the dependency of features.

The complexity and effect of the model are influenced by the number of hidden layer nodes. The ability of the network to learn will be hindered if the number of nodes is too small. The intricacy of the network topology increases as the number of nodes increases. Simultaneously, it is easier to fall into local minimum points during the training process, resulting in a slower network learning rate. It consists of a memory cell (c_t) that maintains its state through time intervals of any length. The input gate (i_t), forget gate (f_t), and output gate (o) are the three non-linear gates that make up the LSTM unit (o_t). The data flow in memory cell (c_t) regularization is the primary goal of these three gates. σ(.) is sigmoid, tanh(·) is tangent, and ⊙ is product function. At a particular time t, input vector is x_t and hidden state vector is h_t. The f_t decides what information needs to be forgotten by outputting a number in [0, 1], i.e.,

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f).$$

The i_t gate will decide what new information should be stored by computing i_t and $\widetilde{C}_t$ and combining, i.e.,

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i),$$
$$\widetilde{C}_t = \tanh(W_c h_{t-1} + U_c x_t + b_c),$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{C}_t .$$

The o_t gate will decide which parts of the cell state should be outputted, i.e.,

$$o_t = \sigma(W_o h_{t-1} \lim_{n\to\infty} \left(1 + \tfrac{1}{n}\right)^n + U_o x_t + b_o)$$
$$h_t = o_t \odot \tanh(c_t).$$

The information resulting from previous and future elements is used by Bi-LSTMs while providing output for the current element. To capture all available information, two independent networks (one for each direction) are used, and the impacts from both networks are combined to forecast the final output. Simply put, bidirectional LSTMs are two LSTMs constructed side by side.

The output of Bi-LSTM, $H \in R^{d\times N}$, isthe hidden vector matrix, where d is the size of hidden layers and N is the length of the given sentence. H consists of output vectors [h_1, h_2,..., h_N]. To perform attention mechanism, we compute α as follows:

$$z = \tanh(W_1 * H),$$
$$\alpha = \mathrm{softmax}(wTz),$$

$$r = H\alpha^T,$$

where $W_1 \in R^{d\times d}$, $w \in R^d$, $z \in R^{d\times N}$, $\alpha \in R^N$. tanh activation is performed on the weighted representation of the sentence $r \in R^d$. Final hidden representation, h_{final}, is computed as follows:

$$h_{final} = \tanh(W_2 * r + b_2),$$

where $W_2 \in R^{d\times d}$, $b_2 \in R^d$. The outputs of the final sentence representation are passed to a softmax classifier to predict label $\hat{y}$ from a discrete set of 27 classes for a sentence S.

$$\hat{y} = \mathrm{softmax}(W3 * h_{final} + b_3),$$

where $W_3 \in R^{1\times d}$, $b_3 \in R$ classes. The weights and bias parameters W_1, W_2, W_3, b_1, b_2 are projection parameters learned during training.

A popular technique of giving varying weights to distinct phrases in a sentence is to use a weighted mixture of all hidden states, SAW as follows:

$$\alpha_t = \frac{\exp(V^T.\ \tilde{h})}{\sum_t \exp(V.\ \tilde{h})}$$

$$S_{A_w} = \sum_t \alpha_t\ h_t\ .$$

h_t is a hidden layer gate and s is the softmax model.

Sentiment classification refers to the algorithmic process of determining whether a piece of text is positive, negative, or neutral in tone. The three-way decision theory classifies final verdicts as either accept, reject, or delay. These three sections stand in for the positive, negative, and border categories in a binary sentiment categorization system. The SA classification is performed as

$$SAclass[M] = \prod_{i=1}^{M} \frac{\alpha_t * \max(h_{\text{final}})}{len(o_t)} + \max\left(FeatSet(i_t, c_t)\right) + \lim_{i \to \text{Vector}} \left(\alpha + \frac{S_A}{h_t}\right)^n.$$

IV. Results

There are two main categories of textual content on the web: facts and opinions. Statements about reality that lack emotional nuance are called facts. Much prior work with online texts has concentrated on factual data for use in a variety of NLP applications as informing retrieval, text categorization, etc. Research on comprehending subjective expressions is limited at present because of the multiple challenges associated with the topic. Research into SA has become increasingly popular over the past decade as its many practical implications have become increasingly apparent in both industry and society at large. Opinion mining, also known as sentiment analysis, is a field of research that delves into how readers feel about products, services, and other topics mentioned in the text. Automatically analysing web content to create an opinion calls for a system with a deep comprehension of natural language.

There are three different levels at which SA can be studied: documents, sentences, and features/aspects. The SA of a review document indicates if the polarity is positive or negative. The document is handled as if it were a single unit. Sentence-level SA looks at a single sentence to determine the attitude or viewpoint expressed. Sentiment analysis based on documents or individual sentences does not work. Aspect type SA refers to methods for extracting the features and context from a text in which a particular attitude is expressed. This research proposes multichannel word embedding and employing stack of neural networks with lexicon-based padding and attention mechanism (MCSNNLA) method for SA. The proposed model is compared with the traditional sentiment convolutional neural network (SentiCNN) [3] model and the results represent that the proposed model is better in the analysis.

For the purpose of evaluating the stack of neural networks described, two English language datasets are considered. Of the 25 000 reviews included in IMDB's dataset, the same number is positive and negative. There are a total of 6257 tweets in the datasets, 1567 of which are good, 1563 of which are negative, and 3127 of which are neutral. The dataset considered and its properties are represented in Table IV.

TABLE IV

Properties of the Considered Datasets

Parameter	IMDB	Twitter
Positive sentences count	12 500	1567
Negative sentences count	12 500	1563
Neutral sentences count	---	3127
Minimum length of reviews	1 word	1 word
Maximum length of reviews	162 words	147 words
Maximum length of reviews	11 words	8 words
Total no of reviews	25 000	6257

The model is trained on challenging data by beginning with an embedding layer of 300 inputs in length that uses ReLU and sigmoid as activation functions. To lessen the suggested model complexity and overfitting, neurons are randomly removed at a dropout rate of 0.3. Selecting an appropriate learning rate is essential when attempting to optimize weights and offsets. Overshooting the extreme point is easy if the learning rate is too great, making the system unstable. The list of parameters is represented in Table V.

TABLE V

List of Parameters

Parameter	Value
Datasets	2
Optimizer	adam
Loss function	categorical_crossentropy
Epochs	70
Learning rate	0.2
Maximum length	100
Dense layer	3
Dropout	0.24
Batch size	120
Node number	128
Vector size	300

$$\text{Sensitivity} = \text{Recall} = \frac{TP}{TP+FN},$$

$$\text{Positive Prediction Value} = \frac{TP}{TP+FP},$$

$$\text{Specificity} = \frac{TN}{TN+FP},$$

$$\text{Precision} = \frac{TP}{TP+TN},$$

$$F1-\text{Score} = \frac{\text{Presision}\cdot\text{Recall}}{\text{Presision}+\text{Recall}}.$$

The accuracy can be expresses as the ratio of precise guesses to the amount of entire classified cases:

$$\text{Accuracy} = \frac{\text{True prediction count}}{\text{Total prediction count}} = \frac{TP+TN}{TP+TN+FP+FN}$$

A set of traits with an associated emotional value is called a lexicon. The dictionary is essentially a prepared list of words used for reference, with numerous synonyms connected with each individual word. Padding is a subset of masking in which the steps to be skipped are placed at the beginning or end of the sequence. Encoding sequence data into continuous batches necessitates padding because some sequences must be padded or truncated to conform to a predetermined length standard. The padding accuracy levels of the proposed and existing models are shown in Fig. 6.

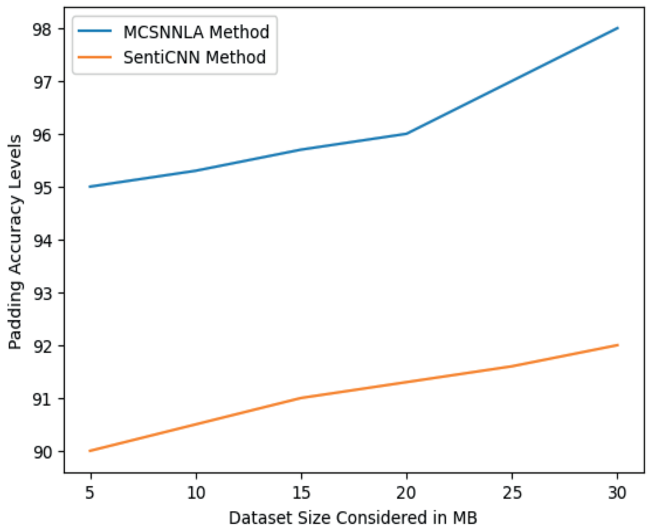

Fig. 6. Padding accuracy levels.

Pre-trained word embeddings are used in the matrix, and the resulting channel is referred to as Word2Vec or glove. Thus, the ultimate representation of an expression or phrase employing many ways of representation is multichannel word embedding. Text analysis is one application of the embedding. Words that are near together in vector space are anticipated to have similar meanings, and this is encoded in the representation, which is often a real-valued vector. The multichannel word embedding time levels of the proposed and existing models are shown in Fig. 7.

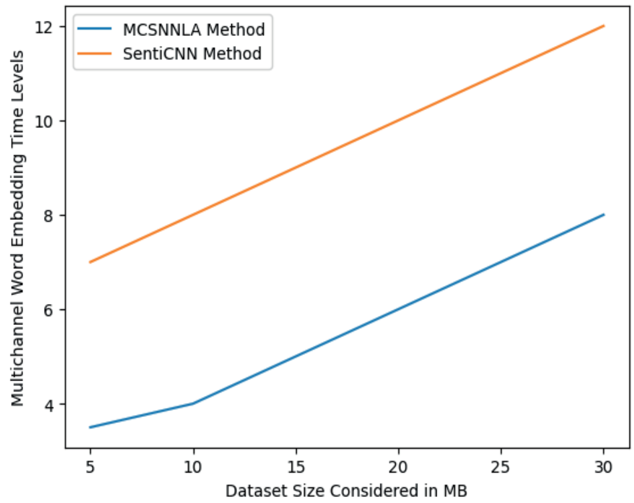

Fig. 7. Multichannel word embedding time levels.

Feature selection is a technique for streamlining the model inputs by identifying and excluding irrelevant information. It is when the learning model makes feature selections on its own based on the sort of problem users trying to address. The DL based feature selection accuracy levels of the proposed models are shown in Fig. 8.

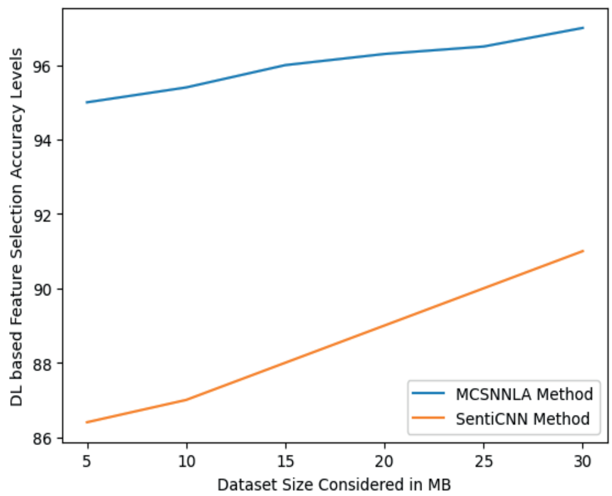

Fig. 8. DL based feature selection accuracy levels.

Sentiment classification refers to the algorithmic process of determining whether a piece of text is positive, negative, or neutral in tone. Classifying whether the conveyed opinion in a document, sentence is positive, negative, or neutral is one of the most fundamental tasks in sentiment analysis. The sentiment classification accuracy levels of the proposed and traditional models are shown in Fig. 9.

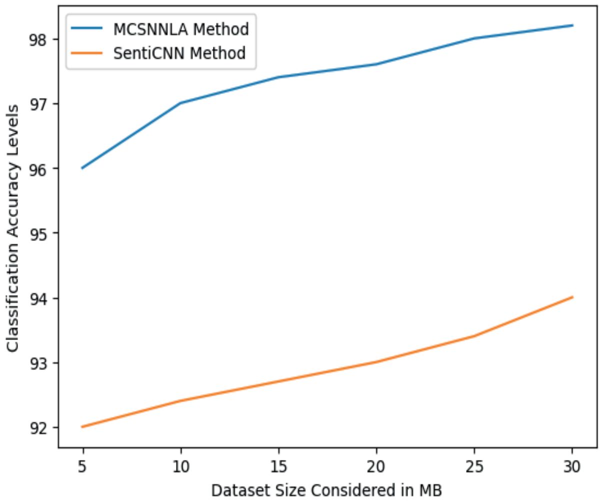

Fig. 9. Classification accuracy levels.

V. CONCLUSION

User satisfaction can be improved by analysing the tone of a large number of reviews posted on social media. In this research, a novel SA model called MCSNNLA, which is based on a multichannel approach for sentence level SA employing a stack of neural networks with lexicon-based padding and attention mechanism, is proposed. Lexicon-based padding solves the problem of input sequences that are not uniform. Convolutional neural networks are a well-explored deep learning framework, and this study provides multiple methods that successfully incorporate lexical embeddings and an attention mechanism to this framework. These results demonstrate that incorporating a vocabulary into a conventional CNN model can increase its precision, robustness, and efficacy. The generalizability of the benefits of lexicon embeddings and attention vector embeddings is demonstrated by the outcomes of training with multiple random seeds. Another benefit of this integration is more robust learning, as demonstrated by the comparison of training curves. Together, word2vector and weights gave a multichannel strategy for word embedding. The effectiveness of the provided MCSNNLA method was assessed using two text datasets. The presented method achieves an accuracy of 93.98 % on the Twitter dataset and 93.71 % on the IMDB dataset. In future, hybrid optimization techniques can be applied with the deep neural networks for enhancing the analysis accuracy levels.

REFERENCES

[1] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment analysis for E-commerce product reviews in Chinese based on sentiment lexicon and deep learning," *IEEE Access*, vol. 8, pp. 23522–23530, Jan. 2020. https://doi.org/10.1109/ACCESS.2020.2969854

[2] F. Yin, Y. Wang, J. Liu, and L. Lin, "The construction of sentiment lexicon based on context-dependent part-of-speech chunks for semantic disambiguation," *IEEE Access*, vol. 8, pp. 63359–63367, Mar. 2020. https://doi.org/10.1109/ACCESS.2020.2984284

[3] M. Huang, H. Xie, Y. Rao, Y. Liu, L. K. M. Poon, and F. L. Wang, "Lexicon-based sentiment convolutional neural networks for online review analysis," *IEEE Transactions on Affective Computing*, vol. 13, no. 3, pp. 1337–1348, July-Sept. 2022. https://doi.org/10.1109/TAFFC.2020.2997769

[4] Z. Ren, G. Zeng, L. Chen, Q. Zhang, C. Zhang and D. Pan, "A lexicon-enhanced attention network for aspect-level sentiment analysis," *IEEE Access*, vol. 8, pp. 93464–93471, May. 2020. https://doi.org/10.1109/ACCESS.2020.2995211

[5] S. Sanagar and D. Gupta, "Unsupervised genre-based multidomain sentiment lexicon learning using corpus-generated polarity seed words," IEEE Access, vol. 8, pp. 118050–118071, Jun. 2020. https://doi.org/10.1109/ACCESS.2020.3005242

[6] Y. Wang, G. Huang, J. Li, H. Li, Y. Zhou, and H. Jiang, "Refined global word embeddings based on sentiment concept for sentiment analysis," *IEEE Access*, vol. 9, pp. 37075–37085, Mar. 2021. https://doi.org/10.1109/ACCESS.2021.3062654

[7] B.-W. On, J.-Y. Jo, H. Shin, J. Gim, G. S. Choi, and S.-M. Jung, "Efficient sentiment-aware Web crawling methods for constructing sentiment dictionary," *IEEE Access*, vol. 9, pp. 161208–161223, Nov. 2021. https://doi.org/10.1109/ACCESS.2021.3129187

[8] O. Wu, T. Yang, M. Li, and M. Li, "Two-level LSTM for sentiment analysis with lexicon embedding and polar flipping," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3867–3879, May 2022. https://doi.org/10.1109/TCYB.2020.3017378

[9] K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev, and D. Trajanov, "Evaluation of sentiment analysis in finance: From lexicons to transformers," *IEEE Access*, vol. 8, pp. 131662–131682, Jul. 2020. https://doi.org/10.1109/ACCESS.2020.3009626

[10] B. M. A. Tahayna, R. K. Ayyasamy, and R. Akbar, "Automatic sentiment annotation of idiomatic expressions for sentiment analysis task," *IEEE Access*, vol. 10, pp. 122234–122242, Nov. 2022. https://doi.org/10.1109/ACCESS.2022.3222233

[11] A. A. Raza, A. Habib, J. Ashraf, B. Shah, and F. Moreira, "Semantic orientation of cross lingual sentiments: Employment of lexicon and dictionaries," *IEEE Access*, vol. 11, pp. 7617–7629, Jan. 2023. https://doi.org/10.1109/ACCESS.2023.3238207

[12] M. R. Wrobel, "The impact of lexicon adaptation on the emotion mining from software engineering artifacts," *IEEE Access*, vol. 8, pp. 48742–48751, Mar. 2020. https://doi.org/10.1109/ACCESS.2020.2979148

[13] L. G. Singh, A. Anil, and S. R. Singh, "SHE: Sentiment hashtag embedding through multitask learning," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 417–424, Apr.2020. https://doi.org/10.1109/TCSS.2019.2962718

[14] I.Awajan, M. Mohamad, and A. Al-Quran, "Sentiment analysis technique and neutrosophic set theory for mining and ranking big data from online reviews," *IEEE Access*, vol. 9, pp. 47338–47353, Mar. 2021. https://doi.org/10.1109/ACCESS.2021.3067844

[15] A. Rasool, R. Tao, M. Kamyab, and S. Hayat, "GAWA–A feature selection method for hybrid sentiment classification," *IEEE Access*, vol. 8, pp. 191850–191861, Oct. 2020. https://doi.org/10.1109/ACCESS.2020.3030642

[16] F. Iqbal *et al.*, "A hybrid framework for sentiment analysis using genetic algorithm based feature reduction," *IEEE Access*, vol. 7, pp. 14637–14652, Jan. 2019. https://doi.org/10.1109/ACCESS.2019.2892852

[17] S. Poria, N. Majumder, D. Hazarika, E. Cambria, A. Gelbukh, and A. Hussain, "Multimodal sentiment analysis: Addressing key issues and setting up the baselines," *IEEE Intell. Syst.*, vol. 33, no. 6, pp. 17–25, Nov.–Dec. 2018. https://doi.org/10.1109/MIS.2018.2882362

[18] S. Wu, F. Wu, Y. Chang, C. Wu, and Y. Huang, "Automatic construction of target-specific sentiment lexicon," *Expert Syst. Appl.*, vol. 116, pp. 285–298, Feb. 2019. https://doi.org/10.1016/j.eswa.2018.09.024

[19] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, "SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," in *Proc. 32th Int. Conf. Assoc. Adv. Artif. Intell.*, vol. 32, no. 1, pp. 1795–1802, Apr. 2018. https://doi.org/10.1609/aaai.v32i1.11559

[20] R. Othman, Y. Abdelsadek, K. Chelghoum, I. Kacem, and R. Faiz, "Improving sentiment analysis in Twitter using sentiment specific word embeddings," in *Proc. 10th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Systems: Technol. Appl. (IDAACS)*, Metz, France, Sep. 2019, pp. 854–858. https://doi.org/10.1109/IDAACS.2019.8924403

[21] X. Wang, J. Chen, A. Hawbani, F. Miao, and C. Shao, "Building sentiment lexicon with representation learning based on contrast and label of sentiment," in *Proc. 4th Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Chicago, IL, USA, Aug. 2018, pp. 151–156. https://doi.org/10.1109/BIGCOM.2018.00032

[22] Y. Wang, Y. Zhang, and B. Liu, "Sentiment lexicon expansion based on neural PU learning double dictionary lookup and polarity association," in *Proc. Conf. Empirical Methods Natural Lang. Process*., Copenhagen, Denmark, Sep. 2017, pp. 553–563. https://doi.org/10.18653/v1/D17-1059

[23] Q. Yang, Y. Rao, H. Xie, J. Wang, F. L. Wang, and W. H. Chan, "Segment-level joint topic-sentiment model for online review analysis," *IEEE Intell. Syst*., vol. 34, no. 1, pp. 43–50, Jan.-Feb. 2019. https://doi.org/10.1109/MIS.2019.2899142

[24] Z. Lei, Y. Yang, and M. Yang, "Sentiment lexicon enhanced attention-based LSTM for sentiment classification," in *Proc. Nat. Conf. Artif. Intell*., vol. 32, no. 1, Apr. 2018, pp. 8105–8106. https://doi.org/10.1609/aaai.v32i1.12142

[25] C. Wu, F. Wu, J. Liu, Y. Huang, and X. Xie, "Sentiment lexicon enhanced neural sentiment classification," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage*., Nov. 2019, pp. 1091–1100. https://doi.org/10.1145/3357384.3357973

[26] Y. Cao, Y. Cao, S. Wen, T. Huang, and Z. Zeng, "Passivity analysis of delayed reaction-diffusion memristor-based neural networks," *Neural Netw*., vol. 109, pp. 159–167, Jan. 2019. https://doi.org/10.1016/j.neunet.2018.10.004

[27] Y. Cao, S. Wang, Z. Guo, T. Huang, and S. Wen, "Synchronization of memristive neural networks with leakage delay and parameters mismatch via event-triggered control," *Neural Netw*., vol. 119, pp. 178–189, Nov. 2019. https://doi.org/10.1016/j.neunet.2019.08.011

[28] M. Dong, S. Wen, Z. Zeng, Z. Yan, and T. Huang, "Sparse fully convolutional network for face labeling," *Neurocomputing*, vol. 331, pp. 465–472, Feb. 2019. https://doi.org/10.1016/j.neucom.2018.11.079

Venkateswara Rao Kota completed BTech in Computer Science and Engineering at SVH College of Engineering and Technology, Machilipatnam, Andhra Pradesh, India in 2005. He completed MTech in Computer Science and Technology at GITAM College of Engineering and Technology, Vizag, Andhra Pradesh, India in 2009. He has published 10 papers in reputed international Scopus indexed journals.

E-mail: kvraliet@gmail.com
ORCID iD: https://orcid.org/0000-0003-4841-5531

M. Shyamala Devi completed B.E in Computer Science and Engineering at P.S.N.A College of Engineering and Technology, Dindigul, Tamil Nadu, India in 2005. She has completed M.E in Computer Science and Engineering at P.S.N.A College of Engineering and Technology, Dindigul, Tamil Nadu, India in 2009. She completed M.B.A in Systems Area at Madurai Kamaraj University, Madurai. She received PhD in the area of Cloud Computing from Anna University, Chennai in 2017. She has more than 18 years of teaching experience to her credit and her research interests include theoretical computer science, distributed system and cloud computing. She has authored 13 engineering books on Theory of Computation, Principles of Compiler Design, Data Structures and Algorithm Analysis, Graphics and Multimedia, Fundamentals of Computer Programming, Digital Computer Fundamentals and Visual Programming published by Shri Krishna Hi Tech Publishing Pvt Ltd, Chennai, India. She has also authored a textbook on Theory of Computation from Sahara Publications, National Book Traders, Chennai. She has published 150 papers in reputed international SCI/ Scopus indexed journals and presented 100 papers in reputed IEEE/Springer/ACM international conferences and won several best paper awards.

E-mail: shyamaladevim@veltech.edu.in
ORCID iD: https://orcid.org/0000-0001-6118-389X