Array 14 (2022) 100157

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Transformer-based deep learning models for the sentiment analysis of social media data

Sayyida Tabinda Kokab *, Sohail Asghar, Shehneela Naz

Department of Computer Science, COMSATS University Islamabad, Islamabd 44000, Pakistan

ARTICLE INFO

Keywords:
Sentiment analysis
Social media
Deep learning
BERT
CNN
LSTM

ABSTRACT

Sentiment analysis (SA) is a widely used contextual mining technique for extracting useful and subjective information from text-based data. It applies on Natural Language Processing (NLP), text analysis, biometrics, and computational linguistics to identify, analyse, and extract responses, states, or emotions from the data. The features analysis technique plays a significant role in the development and improvement of a SA model. Recently, GloVe and Word2vec embedding models have been widely used for feature extractions. However, they overlook sentimental and contextual information of the text and need a large corpus of text data for training and generating exact vectors. These techniques generate vectors for just those words that are included in their vocabulary and ignore Out of Vocabulary Words (OOV), which can lead to information loss. Another challenge for the classification of sentiments is that of the lack of readily available annotated data. Sometimes, there is a contradiction between the review and their label that may cause misclassification. The aim of this paper is to propose a generalized SA model that can handle noisy data, OOV words, sentimental and contextual loss of reviews data. In this research, an effective Bi-directional Encoder Representation from Transformers (BERT) based Convolution Bi-directional Recurrent Neural Network (CBRNN) model is proposed with for exploring the syntactic and semantic information along with the sentimental and contextual analysis of the data. Initially, the zero-shot classification is used for labelling the reviews by calculating their polarity scores. After that, a pre-trained BERT model is employed for obtaining sentence-level semantics and contextual features from that data and generate embeddings. The obtained contextual embedded vectors were then passed to the neural network, comprised of dilated convolution and Bi-LSTM. The proposed model uses dilated convolution instead of classical convolution to extract local and global contextual semantic features from the embedded data. Bi-directional Long Short-Term Memory (Bi-LSTM) is used for the entire sequencing of the sentences. The CBRNN model is evaluated across four diverse domain text datasets based on accuracy, precision, recall, f1-score and AUC values. Thus, CBRNN can be efficiently used for performing SA tasks on social media reviews, without any information loss.

1. Introduction

With the advent of digitization and online technology, the development of sharing and expressing emotions, feed-backs or views over the Internet has become incredible [1]. Social media platforms like Facebook, Twitter, Instagram, YouTube, etc, have gained popularity among people. Businesses, consumers, and governments use these platforms for negotiating deals, advertising products and services, discussing important topics, launching campaigns and spreading awareness [2]. The accessibility and advancement in social media technologies have opened new avenues for companies. They used different models for learning people's feedback and attitudes [3]. There have been used several techniques for examining social media material for corporate analytics, intelligence, surveillance of unethical activity, and SA of customer's opinion [4].

SA, which often known as sentiment mining, is a key component of NLP, intending to serve users in analysing and recognizing the emotions included in subjective texts [5]. It is extensively used to analysing social media data in online communities (blogs, Twitter comments, and reviews etc.), and used to identify the sentiment polarity of textual data [6]. Sentiment polarity of a given item stated emotions of a user associated with a piece of text, i.e., whether the text represents the user's positive, negative, or neutral attitude towards the specified item. Consumers may make appropriate buying decisions by detecting the sentiment orientation of a wide range of online product reviews [7].

* Corresponding author.
E-mail addresses: tabi.syed19@gmail.com (S.T. Kokab), sohail.asg@gmail.com (S. Asghar), shahneela.cs@gmail.com (S. Naz).

https://doi.org/10.1016/j.array.2022.100157
Received 29 October 2021; Received in revised form 7 February 2022; Accepted 1 April 2022
Available online 10 April 2022

SA is a particular type of text classification, and according to the requirement, it can be classified as sentence-level [8], document-level, and word or aspect-level SA [9]. For document-level categorization, the entire document is regarded as a single entity, while for the sentence-level classification, a sentence may alternatively be considered as a mini-document [10]. Aspect-based SA directly concentrates on just one aspect or word and its associated polarity [11]. Sentence-level SA is an emerging field in text mining for feature learning, since it incorporates subjectivity and objectivity of the sentence. Similarly, the task of SA comprises essentially four steps main: preprocessing, feature extraction, classification, and interpretation of results, among various domains like movie reviews [12], election opinion prediction [13], airline reviews [14], amazon reviews [15], etc. Among all the steps mentioned above, feature extraction plays an important role for improving the classification efficiency [16]. There are two types of feature extraction methods, such as lexicon-based methods and machine learning-based or deep learning-based methods [17]. In machine learning-based methods, the model tries to find patterns from the data provided, whereas, in lexicon-based, lists of positive and negative words are provided. These words are counted for each sentence [18]. The sentiment is decided by the frequency of the positive-oriented and negative-oriented words. So, in these lexicon-based methods [19], domain dependency makes them less suitable for the domains without specialized lexicons. However, these techniques often have a low precision rate due to the lack of powerful linguistic resources [3].

Machine learning techniques are further expanded over two parts, such as supervised techniques and unsupervised [20]. Supervised techniques require that the model be trained on labelled data before being evaluated on unseen for determining the model's performance. Unsupervised techniques, on the other hand, train the model using data that has not been labelled or categorized, enabling the model to function without the need for supervision [21]. The most commonly used machine learning techniques for feature extraction are Bag-of-Words (BoW), N-grams [22,23]. In these feature extraction techniques, the features are retrieved either using Count Vectorizer (CV) [24] or Term Frequency–Inverse Document Frequency (TF–IDF) [25]. These techniques use a one-hot word representation approach, in which the vocabulary size depends on the total number of words displayed in the document [26]. This makes the feature space high dimensional and also raised scalability challenges [27]. As TF–IDF is based on the BoW model, it cannot capture the words sequences, their syntactical, and semantic details of a sentence. [28]. However, the selection of relevant features and methodology both are critical for the extraction of feature representation.

To address the constraints of the above described feature extraction techniques, word-embedding models were proposed. Word-embedding models solve problems by extracting semantic and syntactic details from word representations [29]. The widespread usage of word-embedding has refocused the attention of numerous research projects on neural networks [30]. A plethora of studies focused on the sentence-level classification challenge as a broader problem of SA [31]. Word2vec [32] and Glove [33] are two most commonly used word-embedding models for text transformation. Word2Vec is built on the basis of two models (Continuous Bag-of-Words (CBoW) and Skip-gram models) [34]. The CBoW predicts a word based on its context, while the Skip-gram predicts a word based on a central word or target word. On the other hand, the GloVe embedding method is a global log-bilinear regression model for word representation that generates vectors based on the co-occurrence of words and the matrix factorization method.

Traditional embedding models focused on semantic and syntactic characteristics, which are insufficient for SA applications [35]. Although, these highly effective approaches have several limitations that must be addressed. Word2vec and GloVe models require a huge corpus for training and generating embedding for each word [36]. These embedding methods generate feature vectors only for words found in their vocabularies and are unable to cope with Out of Vocabulary (OOV) words [37]. Another limitation of these models is that similar words from different sentences may have similar vector representations. In [38], authors identified that similar words from different sentences may have a different context. Moreover, in such types of techniques, the opposite sentimental words such as "bad" and "good" may have the closest vectors [39]. It may cause sentimental and contextual loss. Twitter is one of the most frequently used social media sites, and each day it processes almost 200 million tweets. Since tweets are so short, people frequently make mistakes while they are tweeting. Sometimes, it can be tough to cope with misspellings and other inconsistencies identified in the language used in social media. Besides, the lack of readily available annotated data, makes the classification task challenging. Thus, there is a need to propose an efficient and scalable model without any domain dependency constraint that can focus on syntactic and semantic features and extract sentimental and contextual features.

To tackle the above-descried problems, an enhanced BERT-based CBRNN model is suggested in order to enhance the performance of sentence-level SA. The significant contributions are:

- Initially, the zero-shot algorithm is used to annotate data, and the BERT model is used to generate semantic and contextual embeddings.
- A dilated CNN model is used to extract local and global sentimental features from embedded features using different dilation rates.
- Bi-LSTM model is used to take advantage of learning long-term dependencies in both directions between word sequences in a long text.
- To modify the parameters of the proposed model, a grid search CV algorithm was utilized.
- A comparative analysis is conducted to check the performance of the proposed BERT-based CBRNN model.

The reminder of this paper is arranged as follows: Section 2 describes related work, Section 3 contains details of Background, Section 4 discusses proposed methodology for SA, Section 5 includes implementation detail, Section 6 contains conclusion and future directions of the paper.

2. Literature review

This section reviews the literature that has previously been done in text classification. The literature review may be subdivided into two sections. The first one will discuss the word embeddings and state-of-the-art transformers. The second part will discuss the classification models of SA.

2.1. Word embeddings and transformers

Feature extraction is an important stage in text mining or SA, and the methods used for extracting the features significantly, impact the results. Deep learning models have recently been adopted in the field of SA for learning word embeddings. Word embeddings aim to capture similarities between words and their linguistic connections [4]. Widely used unsupervised-based word embeddings are word2vec [40] and glove embeddings [41,42]. These approaches are founded on the assumption that words containing similar context have the same meaning, they should create similar feature or vectors accordingly [43]. However, the fundamental problem of this assumption is that the obtained vector of certain semantically dissimilar words, which commonly co-occur in a limited region, is similar. Usually, these approaches projected opposite words into the nearest vectors, but actually they have a very opposite meaning, e.g vectors of two dissimilar words "like" and "dislike".

A novel neural word embedding-based approach was suggested by [44] for SA across several domains. They addressed the major limitations of existing methods, which did not perform well on using in other domains than the domain it was trained on. Their new technique outperformed the old one by achieving higher performance. However, these approaches required a large training corpus for generating accurate vectors. In 2019, another word embedding vector was proposed by [30]. They called it model Improved Word Vector (IWV), which was a combination of Parts of Speech (POS) tagging methods, word2vec/glove models, lexicon-based method, and word localization algorithm. This model had minor improvements in accuracy and needed high computation power because four GPUs were used for training this embedding model. Similarly, a word2sent sentimental embedding model was proposed by [39]. The model combined CBoW and senti-wordnet-lexicon models for discovering the embeddings for every word from its neighbouring context words. It retained the syntactic and semantic features while implicitly capturing sentiments. Thus, four datasets were used to conduct sentence-level classification using CNN classifier. However, the main downside of all these methods is the cost of finding the opinion/emotion orientation of all word individually in the built-in vocabulary. Furthermore, it is possible that the emotion/opinion orientation terms may vary by domain.

Traditional embedding models used for sentiment analysis cannot deal with OOV words and can potentially lose sentimental information. These techniques also have a drawback that they consider similar words from different sentences into the same context. However, it is evident that words from different sentences would have different contexts [45]. In the last two years, transformer-based word embedding models have generated vectors for many text classification tasks. Similarly, a BERT model that was trained on a Chinese Wikipedia corpus, was used for enhancing the performance of Chinese stock reviews with a fully connected layer [45], and with BiGRU [46]. In [47], the authors have developed a personality identification technique based on the BERT embeddings. They had discovered that the personality recognition from the text using the BERT model might enhance accuracy significantly. Authors in [38], had compared various deep models using different embeddings for SA of drug reviews. They had applied embeddings of pre-trained clinical BERT with LSTM and got compromising results. A comparative study had been conducted between word2vec and BERT on Tunisian SA and thus concluded that BERT with CNN achieved the highest results in terms of accuracy [48].

In sum, previous research indicates that conventional word embedding models have certain flaws that could be addressed using a transformer-based approach. Table 1 presents the gap between traditional embedding models and transformer-based models.

Table 1
Critical analysis of word embedding models.

Models	Syntactical	Semantics	Contextual	Out of vocabulary
1-Hot encoding	[×]	[×]	[×]	[×]
BOW	[×]	[×]	[×]	[×]
TF–IDF	[×]	[×]	[×]	[×]
Word2vec	[✓]	[✓]	[×]	[×]
GloVe	[✓]	[✓]	[×]	[×]
FastText	[✓]	[×]	[×]	[✓]
BERT	[✓]	[✓]	[✓]	[✓]

2.2. Sentiment classification

Deep learning models have been used in sentence-level SA in a various domain over the last several years for overcoming the constraints of conventional machine learning models. CNN and LSTM models have been used with distributed word representations word2vec [34], GloVe [49] and FasText [33] for the SA of social media data. However, CNN is useful for short textual data and may not be suitable for long-length reviews [2,50]. LSTM, on the other hand, is capable of dealing with long-length textual data. However, it might become challenging for data that have very long-term dependencies [51]. To deal with the very long dependencies of sentences, Bi-LSTM has been used [52,53].

Several studies had recommended adopting hybrid models for SA, since the deep learning models had performed best in combination instead of alone [54]. CNN and LSTM have been combined for taking benefits from both, for two-class (positive and negative) polarity detection of drug reviews [55]. Each review had contained a rating from 1–5, which showed the level of satisfaction of the drug user. In [56], a combination of CNN and LSTM models with Word2vec embedding scheme had been used for detecting the finer-grained polarity of IMDB and Amazon reviews. The hybrid model CNN–LSTM had achieved 91% accuracy on both datasets. In another research, [57] a hybrid Convolution bidirectional RNN model was proposed. In which, two-layers of CNN are connected with a Bidirectional Gated Recurrent Unit (Bi-GRU) for the SA of IMDB dataset. CNN had extracted the big collection of sentence-level characteristics, whereas Bi-GRU obtained the chronological features using long-term dependence. These models were domain-specific and performed well exclusively in the specified domain. Therefore, [58] had proposed a Convolution-LSTM (CO-LSTM) based hybrid model, which could act in a scalable manner for diverse domains. A deep convolution network is adopted for extracting important features using a pooling layer, while LSTM was used for the sequential analysis of the long text. Although these models could handle sequences of any length, be employing them in the feature extraction layer of a deep neural network increases the dimensionality of the feature space. Another limitation of such models is that they take different features equally important. Table 2 is a tabular representation of related work.

To overcome the gap of previous studies, [59] had proposed a combined model of Bi-LSTM and a self-attention approach. The model had applied multiple channels for obtaining important features for the document-level text classification. Bi-LSTM had received input from multi-feature channels and then learnt a representation of all sentences, after that a self-attention approach was applied for focusing on the sentiment polarity information of the representation. The authors in [19] proposed an attention-based model with the combination of convolution and RNN, for dealing with long and short sentences. Two independent layers of Bi-LSTM and Bi-GRU were utilized for extracting the past and future contextual information of the features. After that, the attention was used for putting the worth of different words. Although, previous work had demonstrated that CNN-based models had been widely used for targeted sentiment classification, but still had some limitations. The authors in [60] had pointed out that the widely used classical CNNs were limited in terms of the size of convolutional kernels, that led to two major problems in social media SA. The first was that, in regard to semantics, CNN was only sufficient to capture short-term dependency patterns. As a result of the expansion of convolutional kernels, there had been a substantial rise in the number of parameters. Another variant of CNN called dilated CNN (D-CNN) has been proposed [61] for handling the problems of classical CNN.

Therefore, it is concluded that traditional techniques are simple to comprehend, have minimal hardware needs, and perform well on small-size datasets, but they struggle with complicated classification problems and need specialized knowledge for constructing sentiment lexicons. Deep learning-based techniques may decrease the reliance on manual features, thereby solving the limitations of conventional methods. However, SA is similar to sequential modelling, CNN-based methods required multiple CNN layers to handle long-term contextual dependencies. Similarly, RNN-based approaches are very complex, it is difficult for them to accurately extract the dependencies between them in long-range context. Therefore, CNN-based approaches had performed well for short-text reviews, and RNN-based approaches performed well for long-text reviews. The combination of CNN and RNN architectures may overcome some limitations of each, but there are

Table 2
Summary of literature review.

References	Approach	Accuracy	Limitations
[62]	Random sampling, CNN	77.6%	Large corpus needed.
[30]	IWV, CNN	86.5%	Sequential and contextual loss.
[63]	TF–IDF, Voting classifier	79.1%	Better feature engineering required.
[56]	Word2vec, CNN, LSTM	91.2%	Normalization of multi-polarity words.
[64]	Glove, GRU	84.8%	Strong recurrent model required.
[46]	Chinese BERT, Bi-GRU	NA	Domain specific.
[59]	CBOW, CNN	87.2%	Extraction of important features.
[57]	Word2vec, CNN, Bi-GRU	86.2%	Enhanced similarity measures required.
[65]	Glove, Bi-GRU	71.1%	Lack of tricky implicit knowledge.
[38]	Clinical BERT, LSTM	90.4%	Annotation of corpora.
[39]	SentiWordNet, CNN	86.5%	Cost of finding opinion words.
[58]	Word2vec,CNN,LSTM	94.9%	Loss of important features.
[66]	Word2vec, LSTM	85.0%	Specific weights for highly impactful words.
[67]	BOW, TF–IDF, ETC	93.1%	Loss of semantic and synthetic information.
[68]	Chinese BERT-FC	92.6%	Verification needed on large datasets.

chances for losing of contextual and sentimental information. Furthermore, if the model wants to learn the high-level contextual features, it needs to utilize multiple convolution kernels, which may increase the model's complexity. Therefore, in this study, a dilated CNN and Bi-LSTM based classification model is introduced for the textual SA of long and short-text reviews.

3. Background details

3.1. Word embeddings

Word embedding is a technique that converts text into numeric form [36], it is also called word representation technique. Word embedding plays an essential role in text mining because machine learning techniques cannot operate on text data. Technically, the word embedding technique transforms an individual word into a numerical representation, using a vocabulary. It may be trained on a big corpus of text by applying a neural network. There are various types of embedding techniques. These techniques are divided into two classes, frequency and prediction-based embeddings. Frequency-based embedding methods generate text vectors by counting the frequency of frequently occurred words [69] e.g TF–IDF, co-occurrence matrix and CV [70]. Whereas, the prediction-based embedding methods vectorize a word using previous knowledge and neural network [71]. Skip-gram and CBOW are widely used models of this method [72].

3.2. WordPiece tokenizer

Tokenization is a technique that splits a sentence, phrase, paragraph or any other textual material into the smallest pieces called tokens [73]. There are various types of tokenizers, WordPiece tokenizer being one of them, outlined in [74], initially creates the vocabulary with all the characters, subwords, and words found in the training data. The vocabulary list consists of four things:

- Complete words.
- Subwords that appear at the start of a word or in isolation (for example, "se" in "searchability" is given the same vector as the independent sequence of letters "se" in "go get se").
- Subwords that are not at the start of a word and are preceded by '##' indicate this case.
- Individual characters.

The tokenizer in this approach first verifies whether the word is in the vocabulary or not. If not, then it attempts for breaking down the word into the possible maximum number of subwords available in the vocabulary, and as a last alternative, it decomposes the word into individual characters. So, once the vocabulary is established, we apply it to tokenization.

The BERT tokenizer is constructed using the WordPiece algorithm mentioned above. Therefore, after performing these steps, the BERT tokenizer returns token id's and attention masks. The output of the tokenizer will then use as an input of the BERT model for generating contextual embeddings.

3.3. BERT

Transfer learning is a novel paradigm in machine learning that focuses on utilizing information gained for one task to tackle other similar tasks. In 2018, Google had proposed a new type of transformer [75] which is the pertained BERT model. As BERT is a sequential language model. BERT takes input in a sequence of language format $X = (I_0, \ldots, I_n)$ and outputs contextualized vector representation $H = (h_0, \ldots, h_n)$ for the elements of the input sequence. Being a highly generalizable language representation framework, it accomplishes task through encoder. Encoders, are a neural network architecture taken from the transformer and used to create encoded representations of text. The pertained BERT-Mini has four encoder layers. Each encoder block has two sub-layers which are multi-head attention and feed-forward.

Each encoder layer combined of two sub-processes that can be seen in Fig. 1. First one is a multi-head self-attention layer, which adopted series of metric manipulation operations. The input to the encoder first pass through multi-head self attention layer for extracting most important language features. After extraction, features will be normalized using residual connection and input to the feed forward layer. Now the output of the feed forward layer will be input to the 2nd layer of encoder and then the same process will be repeated for the next encoder layers.

The multi-head attention is made up of several heads that run in parallel, and each head is represented by self-attention. The self-attention identifies the relationship among all the words in a given phrase. For better understanding, it is important to explain the self-attention process [76]. Fig. 2 shows the pictorial representation of the self-attention mechanism.

$$z = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \tag{1}$$

So, Eq. (1) shows: Q = query vector, K = key vector, V = value vector, and d_k denotes the dimension of k.

- For the calculation of similarity scores, there is a need to calculate the dot product of the query and key matrix (QK^T).
- Then, the key matrix (QK^T) is divided by $\sqrt{d_k}$.
- After that, the softmax function is used for normalizing and obtaining the score matrix.
- Finally, the attention matrix, Z is obtained by multiplying the score matrix with V.

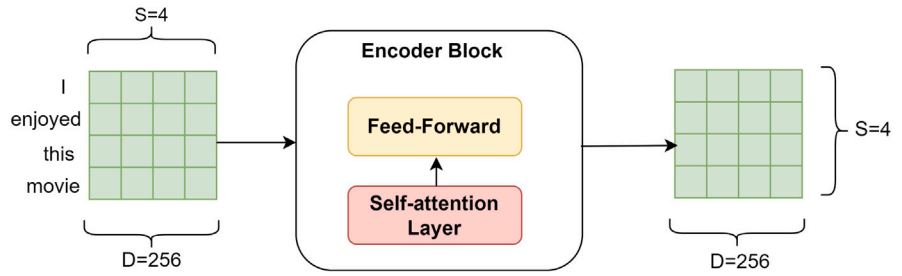


Fig. 1. Schematic diagram of encoder block.

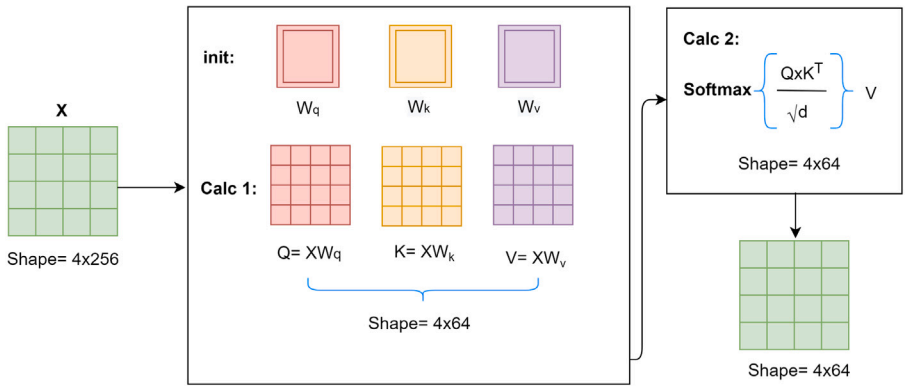


Fig. 2. Schematic diagram of self-attention mechanism.

Similarly, multiple-headed attention enables the model to attend to various representations and subspaces at different locations whereas single attention-head averaging prevents this. Eq. (2) describes multi-head attention, which is denoted by m_a.

$$M_a(Q, K, V) = concat(z_0, z_1, z_2, \ldots ....z_h) \tag{2}$$

The feed-forward is made up of two linear transformations separated by a ReLU activation. It is employed in each and every spot individually and identically. Mathematically, it can be described as:

$$F_n = max(0, xW_1 + b_1)W_2 + b_2, \tag{3}$$

3.4. Convolution neural network

CNN is typically built on convolution and sub-sampling techniques that can operate on different layers [71]. CNNs are employed in NLP for the extraction of local features. The constituent layers of CNN are the convolution, pooling, and fully connected layer. The convolution layer is generally applied on features of the input data. If it is applied to text data, it contributes to the extraction of features from a sentence or phrase-level representations. The function of the pooling layer is to shrink the size of the feature-map obtained from the convolution layer. It is an efficient technique for reducing the size of trainable parameters from high-dimensional input data. Pooling operations can be performed in different ways: max pooling, average or sum pooling, etc. In feature map pooling, the highest value from the feature map area covered by the filter is selected through a pooling procedure known as max pooling. This means that after the maximum pooling layer, the result will be a feature map having the most important features of the preceding feature map. Average pooling takes into account the average of the features in the filter's feature map area. Max pooling provides the most significant aspect in a patch, whereas average pooling provides the mean of all aspects in a patch. After the pooling layer has completed its job, the results are combined to create a pooled feature vector. The obtained vector can then be forwarded to a fully-connected layer. Pooling may provide the convolution kernel a wider receptive field in classical CNN, although it is not a required component of CNN. Excessive use of pooling procedures, on the other hand, tends to result in a significant degree of information loss.

Another improved version of CNN, known as dilated CNN, has been proposed [61] for handling the limitation of conventional CNN. It has various dilation rates, been utilized in several areas such as audio processing, computer vision, and NLP. Dilated convolution has the advantage of expanding the receptive field without requiring pooling, enabling every convolution result to include a broad range of information. It has been used to issues requiring longer sequential information dependencies, such as images and text. Fig. 3 shows the dilated convolution against different dilation rates 1, 2 and 3 using 3×3 kernel size. Essentially, this is a more conventional convolution, but it can be used for collecting increasingly global context from input features without having for increasing number of the parameters. This may also assist to expand the output's spatial size by increasing the number of outputs. However, the most essential thing here is noted that the size of the receptive field increases when there is increase in number of layers.

3.5. Long short-term memory

The LSTM is an advanced form of RNN that is specifically intended for sequential modelling. It is most often employed on text data. When the distance among two dependent words is increased, the effectiveness of the RNN frequently degrades, and the value of gradient is reduced substantially. LSTM solves this issue and works effectively in long-term dependence cases. In LSTM, just an important part of the data is sent to the next layer, rather than the whole data. There are two variants of LSTM are unidirectional LSTM and bidirectional LSTM, both of which are used in machine learning. Its information preservation is restricted to what it has learned from previous inputs, which is due to the fact that the unidirectional LSTM has only seen prior inputs. On the other hand, bidirectional LSTM processes input into two ways, such as forward and backward. The researchers found that Bi-LSTM models beat LSTM models in terms of prediction accuracy [77]. LSTM units are formed of

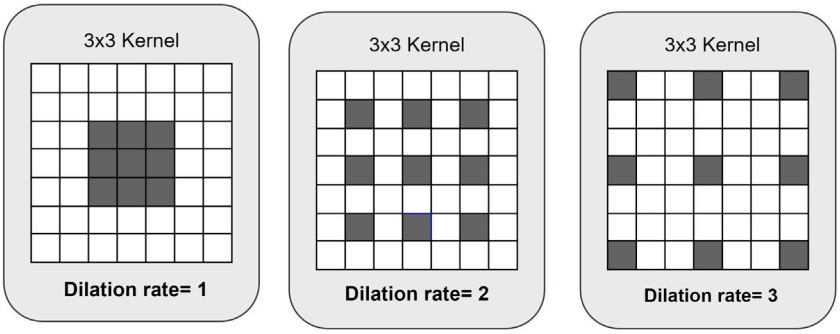


Fig. 3. Systematic 1-D dilated convolution.

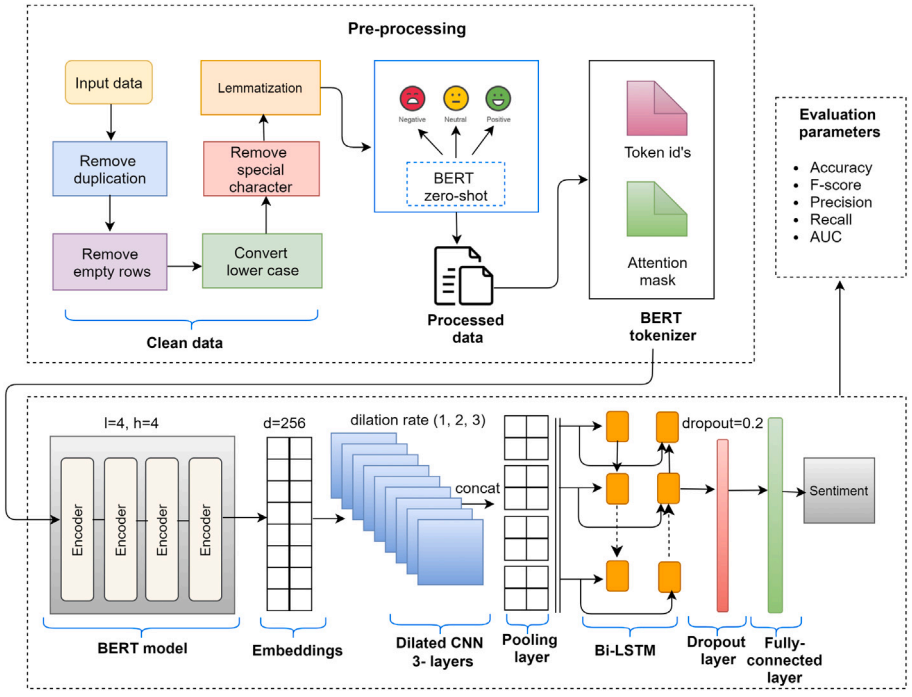


Fig. 4. Proposed model for SA of social media data.

a memory block t_n, which retains its state across unlimited time periods. It consists of three gates such as input e_n, forget p_n, and an output u_n gate. These three gates are intended to control information flow.

$$p_n = \sigma(W_n[h_{n-1}, x_n] + b_n), \tag{4}$$

where σ is used for sigmoid function, W and b denotes weights and their biases, the input of the n^{th} cell is denoted by x_n and output of the preceding LSTM cell is denoted by h_{n-1}. In Eq. (4), p_n may decide which information has to be ignored via the use of x_n and h_{n-1} parameters.

$$e_n = \sigma(W_e[h_{n-1}, x_n] + b_e), \tag{5}$$

$$\tilde{t_n} = tanh(W_t[h_{n-1}, x_n] + b_t), \tag{6}$$

e_n determines that which information will be updated and $\tilde{t_n}$ denotes the information of candidate cell. According to Eq. (7), the input gate determines which information will be stored by calculating e_n and merging it with $\tilde{t_n}$.

$$t_n = p_n * t_{n-1} + e_n * \tilde{t_n}, \tag{7}$$

u_n can decide the state of the output via x_n and h_{n-1}, and afterwards the outcome of the LSTM cell x_n can be determined by multiplying the u_n and the t_n.

$$u_n = tanh(W_u[h_{n-1}, x_n] + b_u), \tag{8}$$

$$h_n = u_n * tanh(t_n), \tag{9}$$

Finally, Bi-LSTM combines forward-going (right-to-left) h_n^f and backward-going (left-to-right) h_n^b hidden layers. This leads to the two-way flow of time in the network, as well as improved learning among the network.

4. Proposed methodology for SA

The proposed work includes the steps of data preprocessing, tokenization and padding, the transformation of data, and the extraction of contextual embeddings using the BERT model. In this study, the BERT model has been used to accomplish two major tasks: annotation of data and extraction of embeddings from the input data. Fig. 4 shows the schematic diagram of the proposed model for SA of social big data. After completing the necessary steps for cleaning the data, the zero-shot classification method has been used for identifying the intensities for annotation. Once the embeddings have been extracted, the BERT model is used for generating feature vectors, which is then passed to three dilated convolution layers to generate a feature map. After that, all the layers are concatenated, and a global max-pooling operation is conducted on them to extract the most relevant features from the

newly created feature map. Bi-LSTM is used for capturing the sequential dependencies between the pooled features in both directions. In this layer, a dropout of 0.2 is included to prevent over-fitting situations from occurring. The outcomes of the Bi-LSTM is given to the hidden layers, and then sent towards the fully-connected or sigmoid layer for prediction. The most commonly used binary cross-entropy is used in sigmoid layer as the loss function for evaluating results. Algorithm 1 shows all the necessary step that the proposed technique have followed.

Step 1: preprocessing

In most of the cases, the reviews of social platforms contain unstructured text. In this step, unstructured data is converted into a structured form using the feature representation techniques. It is necessary to clean the data before performing any text classification task because the original data contains many irregularities that may lead the embedding model to get confused throughout the tokenization process. Due to the informal language used in social media, social media data is prone to noise. This may contain misspellings, special characters, hyperlinks, symbols, etc. The regex and text hero libraries have been used for cleaning data and keep just the relevant information. After that, all the reviews are transformed to lowercase because the text mining methods are case-sensitive methods. Exploratory Data Analysis (EDA) is performed for finding a relationship between words and to visualize data. Text normalization steps such as lemmatization have been performed using the NLTK library. Lemmatization is the process of simplifying an alternative form of a word using its basic form, or lemma, to reduce the number of words that are used regularly by nature. For example, although the words "go," "going," and "gone" are all distinct, in this case, go will serve as the lemma form for all of them. Once we have completed all the necessary steps to clean the data, then the given reviews are utilized to create labels using zero-shot-BERT. Zero-shot BERT is a rating system that is often used to characterize reviews as positive or negative. Zero-shot is not just limited to calculating positive and negative scores, it also provides information on the intensity of sentiment, i.e., how much positive score or negative score it contained. It is fairly fast and may be used for streaming data over the internet without suffering from a significant speed-to-performance trade-off, which is rare. Fig. 4 depicts all the preprocessing processes that we used in the proposed work.

In this research, after extracting the intensities (scores of positive, negative and neutral) of each review, the data is labelled on the basis of predefined conditions. If the negative score of the review is more than neutral and positive scores, then the review is labelled as negative. If positive class contains the highest score than other two classes, then it is labelled as positive. Thus, at the end, this method produced fully pre-processed data that is used for generating numerical form of these text reviews.

Step 2: tokenization and padding

The words have to be vectorized and submitted for the classification, after the data has been cleaned and labelled. Tokenization is a term used to describe this particular procedure. So, first and foremost, we tokenize our text using the WordPiece tokenizer to create input id's and attention masks for further processing. It is necessary to pad and truncate the text to ensure that all reviews have similar lengths. According to the cumulative distribution function, the maximum length of processed texts for tweets is 20 characters and for the reviews dataset the maximum length is 200 characters. Thus, we fixed the length of tokens according to their maximum length of text. The word id's for the words in our sentences are included inside the input id's and attention mask instructs the model on which word it should concentrate its efforts. The reason for this is that once the input is padded to a certain length, shorter sentences will be added with an additional special token. Because this particular token does not include any unique information, the attention mask will ensure that our BERT model does not consider this when producing contextual embeddings for the token.

Step 3: BERT model

Algorithm 1 Pseudocode for the BERT-based CBRNN model.

Iput: Reviews dataset R_D
Output: Sentiment class (positive, negative)

for each review R in R_D **do**
  Calculate polarity scores (Pos_s, Neu_s, Neg_s), using zero-shot BERT classification.
  if $Pos_s > Neg_s \&\& Pos_s > Neu_s$ **then**
    assign label = positive
  else
    if $Neg_s > Pos_s \&\& Neg_s > Neu_s$ **then**
      assign label = negative
    else
      assign label = neutral
    end if
  end if
end for
for each $R \in$ preprocessed R_D **do**
  Performed operations of WordPiece tokenizer, generate token id's and attention mask.
  Extract word embedding vectors V_{ec} using BERT model.
end for
Split data into $T_{train} and T_{test}$
for each $R \in T_{train}$ **do**
  Applied dilated convolution with $D_r = 1, D_r = 2$ and $D_r = 3$.
  Concatenated the output of the previous step.
  Obtained important features from max-pooling layer.
  Performed sequencing operations using Bi-LSTM.
  Dropout layer with dropout=0.2.
  Flatten layer.
  Dense layer.
  Sigmoid is used for calculating probabilities of labels.
end for
for each $R \in T_{test}$ **do**
  Classify R into sentiments using trained model.
  Show output (positive or negative).
end for

BERT model gets tokens id's and attention masks from the tokenizer as shown in Fig. 4. The ability to produce contextualized word embeddings is the primary benefit of BERT over Word2Vec models. Bert creates word representations that are dynamically influenced by the words around them, whereas word2vec has a fixed representation for each word independent of the context in which it occurs. Another viewpoint is that, from a parameter standpoint, training BERT for a particular task is inefficient. However, due to the problem of computation cost, a pre-trained BERT model has been employed. For this experiment, we used the Bert-Mini model, which contains four encoder layers, four attention heads, and has a dimension of 256. The outputs of each encoder are sent to the encoder above it and the final encoder generates the contextual embeddings/vectors for a particular source sentence.

Step 4: dilated convolution layers

In this step, three dilated convolution layers have been applied on input vectors, to extract sentimental and semantic features while the receptive field grows exponentially in size. The high-level characteristics of each sentence vector are extracted using a convolution operation with dilation rate, and an activation function [78] based on a Rectified Linear Unit (ReLU) is used. ReLU is used for preventing the gradient vanishing issue [79], and as it has been discovered to be six-time faster than $tanh$ and σ [80] activation function. All three convolution layers have dilation rates of 1, 2, and 3 correspondingly, as well as 64 filters with a 3×3 kernel size on each of their respective convolution layers. In order to identify long-term semantic characteristics, a low dilation method has been adopted, that focuses on individual words

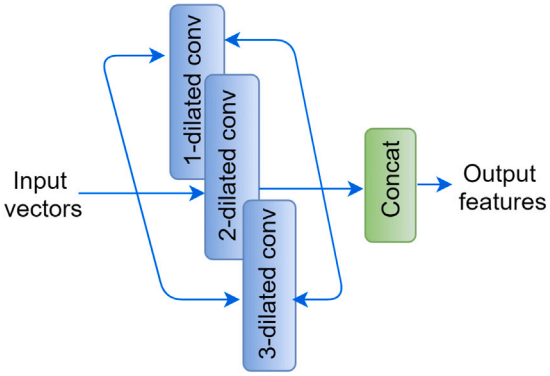


Fig. 5. Dilated convolution block.

and sentences. After obtaining the feature maps from these three layers, the concatenation of all these feature maps are held for getting a single feature map. A single feature map is created when the feature maps from these three layers have been concatenated together in order to create a single feature map, as shown in Fig. 5. The output will be: $X = x_1, x_2, \ldots ..., x_n$ where x_n is the nth word vector.

Step 5: pooling layer

The concatenation layer generates a feature matrix of $n \times m$ dimensions, after which the max-pooling operation is performed using a 2×2 dimension filter. When the filter is traversed in max-pooling mode, the highest or biggest value at each patch of the filter is selected as a result of the selection. As a result, the output of the max-pooling layer would be a pooled feature map that contained the most prominent/important features of the preceding feature map. The resultant feature matrix has dimensions $\frac{n}{2} \times \frac{m}{2}$.

Step 6: Bi-LSTM

The max-pooling layer's output is fed into the Bi-LSTM layer, which analyses the generated feature vectors sequentially in both directions, such as forward direction and backward direction. This layer used 128 units of LSTM and 0.2% dropout. The Bi-LSTM can be expressed by the following equation:

$$h_{nBi-LSTM} = [h_n^f, h_n^b] \tag{10}$$

where h_n^f denotes forward going LSTM and h_n^b denotes backward going LSTM. So, the resultant feature vector which is obtained from the Bi-LSTM is then flattened to feed into the dense layer. The outputs of this dense layer are passed to the fully-connected layer for making predictions of the sentiments.

Step 7: fully-connected layer

After the feature extraction from the output of the preceding dense layer, the probability of the distribution of each category is determined by applying the sigmoid activation function to the feature vector obtained. Mathematically, it is defined as:

$$p_\sigma(c^j) = \frac{e^{oj}}{1 + e^{oj}}, \tag{11}$$

where $P\sigma$ represents the distribution of probabilities for jth value and oj represents the obtained output according to that jth value. The extracted probabilities of the sigmoid layer is then passed for calculating the diversity of sentiments between their actual and predicted values. it is calculated using binary cross-entropy function. The sentiment label of the review text has a discrete value L = [0,1], Where L denotes the label sentiment (negative, positive).

$$loss = -\sum_{j=1}^{r} A(c_i) \times log P\sigma(c_i), \tag{12}$$

where, r represents the total amount of values/categories. Eq. (12) calculates loss by comparing actual values denoted by $A(c_i)$ and predicted values. The main purpose of using the loss function is to reduce the gap between actual and predicted values (see Table 3).

Table 3
Evaluation parameters for airline dataset.

Embeddings	Models	Precision	Recall	F-score	Accuracy	AUC
Word2vec	CNN	0.96	0.93	0.94	0.93	0.950
	LSTM	0.97	0.96	0.96	0.94	0.953
	Co-LSTM	0.94	**0.98**	0.96	0.94	0.968
Glove	CNN	0.90	0.91	0.90	0.91	0.815
	LSTM	0.89	0.92	0.90	0.91	0.801
	Co-LSTM	0.92	0.91	0.91	0.92	0.829
BERT	CNN	0.96	0.96	0.96	0.91	0.815
	LSTM	0.96	0.97	0.97	0.95	0.967
	CBRNN	**0.98**	**0.98**	**0.98**	**0.97**	**0.989**

Table 4
Parameters setting.

CNN filters	Bi-LSTM units	Learning rates	Batch size	Dense size
16	32	0.0001	32	16
32	64	0.00001	64	32
64	128	0.00002	128	64
128	256	0.00005	512	128

5. Implementation

5.1. Datasets

The proposed scheme is evaluated on different datasets for its scalability and efficiency to accurately evaluate techniques on diverse corpora with varying domains and sizes. Four state-of-the-art datasets [58] are taken from diverse domains. We have conducted our experiments on positive and negative reviews and avoid neutral reviews. The description of datasets is given below:

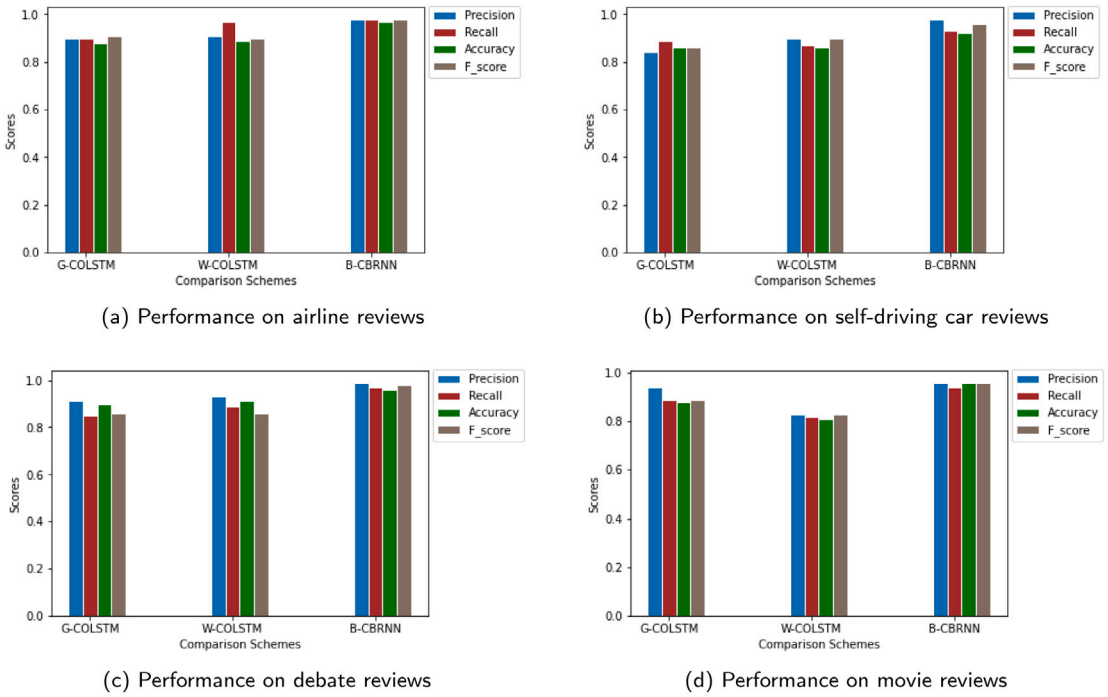
1. Airline reviews: Us-airline datasets are originally collected from Kaggle. It was scrapped in February 2015 [81]. This dataset has 11,517 tweets for six different United States (US) airlines and contains their positive, negative, and neutral sentiments.
2. Self-driving car reviews: Self-driving car dataset [82] has 7156 tweets with three attributes twitter id, reviews, and sentiments associated with each review.
3. US presidential election reviews: US presidential election dataset [83], was the first GOP for the first 2016 GOP presidential debate, which contains 10,729 reviews and 21 attributes.
4. IMDB: IMDB is a large movie review dataset that was collected from Kaggle. This is a balanced dataset and contains 50,000 reviews. It contains 25,000 positive and 25,000 negative reviews.

5.2. Parameters setting

For finding the optimal hyperparameters for the proposed deep learning models, a grid search CV with the 5-fold cross validations to avoid overfitting conditions, is applied. Table 3 lists the many parameters that were tested to determine the overall performance. Grid search CV is a technique in which appropriate values of hyperparameters are given, and then the proposed model is run using those hyperparameter values. The best parameters that provide us with the best results are then identified and selected. The hyperparameters that have been considered for the proposed model are: learning rate, batch size, and dense size. The optimal parameters for the CBRNN model that were discovered via grid search are as follows: CNN filters 64, Bi-LSTM units 128, learning rate 0.00005, batch size 64, and dense size 32 (see Fig. 6).

5.3. Evaluation parameters

In this section, the detail overview of the experimental findings has been conducted. Therefore, the performance of the proposed CBRNN



(a) Performance on airline reviews

(b) Performance on self-driving car reviews

(c) Performance on debate reviews

(d) Performance on movie reviews

Fig. 6. Performance comparison for top models.

technique has been assessed by obtaining the overall scores for recall, precision, f-score, and accuracy. These scores are calculated using the confusion matrix. The Receiver Operating Characteristic (ROC) and the Area Under the Curve (AUC), are also used for evaluating the effectiveness of the model. Many text classification tasks, including SA, make significant use of these performance matrices. The BERT-based CBRNN model is compared with other deep learning models using different embedding schemes.

A confusion matrix can be described as, it is a visual representation of the results obtained from the classification prediction of any problem. In each class, the number of accurate and wrong predictions are summarized with statistical values. It keeps track of the correct classifications and misclassifications, considering four components false positive (F_P), false negative (F_n), true positive (T_P) and true negative (T_n). In this paper, the reviews are classified into two labels (positive and negative). The formulas to calculate the performance of parameters, that has been used in this paper, are stated below:

1. Precision: Precision calculates that, out of the total positive predicted results, how many results were positive. It is used to calculate the exactness of the classifier. It is also called positive predication value, denoted by P_{re}, and can be expressed as:

$$P_{re} = \frac{T_p}{T_p + F_p} \tag{13}$$

2. Recall: Recall identifies, the number of correctly predicted positive samples from the total number of actual positive samples. It is denoted by R_{ec} and can be expressed as:

$$R_{ec} = \frac{T_p}{T_p + F_n} \tag{14}$$

3. F-measure: It is the combination of P_{re}) and R_{ec}, to obtain the harmonic mean, denoted by F_{me}. Mathematically, it can be expressed as:

$$F_{me} = \frac{2 \times P_{re} \times R_{ec}}{P_{re} + R_{ec}} \tag{15}$$

Table 5
Evaluation parameters for self-driving car dataset.

Embeddings	Models	Precision	Recall	F-score	Accuracy	AUC
Word2vec	CNN	0.93	0.85	0.89	0.83	0.867
	LSTM	0.95	0.84	0.89	0.83	0.868
	Co-LSTM	0.94	0.87	0.90	0.86	0.909
Glove	CNN	0.89	0.87	0.88	0.87	0.835
	LSTM	0.88	0.83	0.85	0.83	0.841
	Co-LSTM	0.84	0.89	0.86	0.86	0.809
BERT	CNN	0.95	0.89	0.86	0.86	0.809
	LSTM	0.92	0.90	0.90	0.88	0.925
	CBRNN	**0.96**	**0.91**	**0.94**	**0.90**	**0.958**

4. Accuracy: Accuracy is the proportion of samples for which the predictions are accurate. It is denoted by A_{cc}.

$$A_{cc} = \frac{T_p + T_n}{T_p + F_p + T_n + F_n} \tag{16}$$

5.4. Results and discussion

The proposed BERT-based CBRNN model's performance is evaluated in comparison with different deep learning approaches. For the sake of evaluation, CNN, CO-LSTM [58] and LSTM based fair comparisons are performed with proposed approach. The comparison results are shown in Tables 3–7 have shown the obtained results and graphically represented in Fig. 6.

Table 3 has shown the results of commonly used deep learning approaches for the SA of the US-airline reviews dataset. The evaluation process is conducted using different evaluation parameters such as accuracy, precision, recall, f-score and AUC. The BERT-based CBRNN models achieved the highest scores of 0.97% and 0.989, in terms of accuracy and AUC values, respectively. Word2vec-based LSTM got 0.98% recall which is same as to the BERT-based CBRNN model while the proposed CBRNN model has the highest precision rate 0.98% than the other models. So, it can be observed that the proposed model got better results as compared to other baseline models.

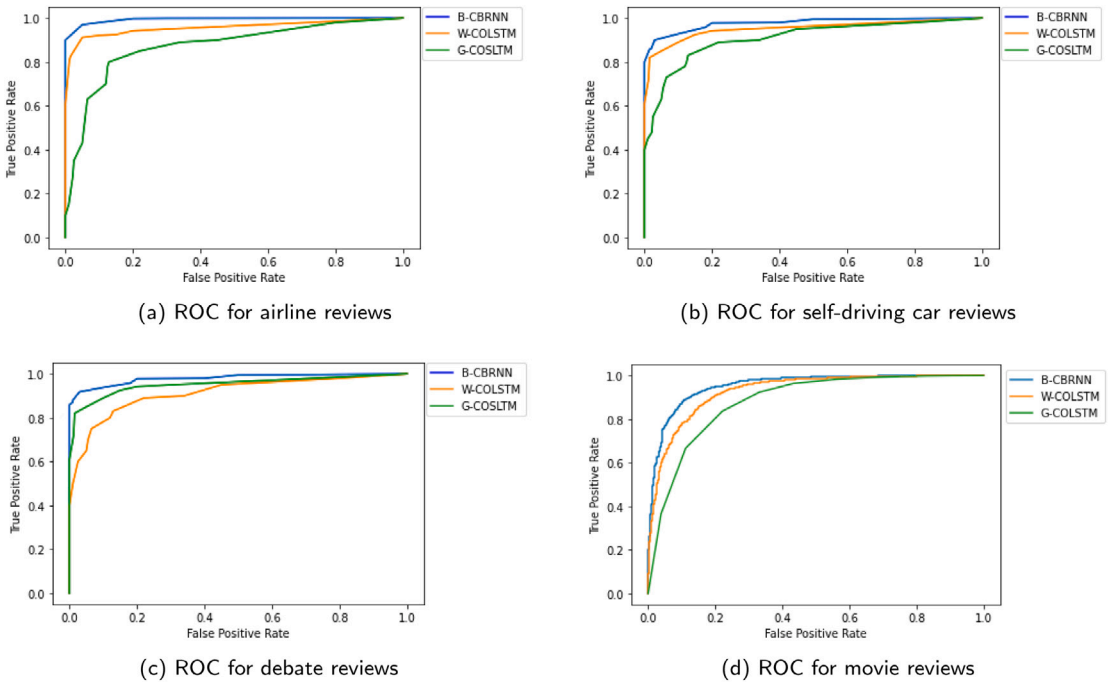


Fig. 7. ROC comparison for top hybrid classifiers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

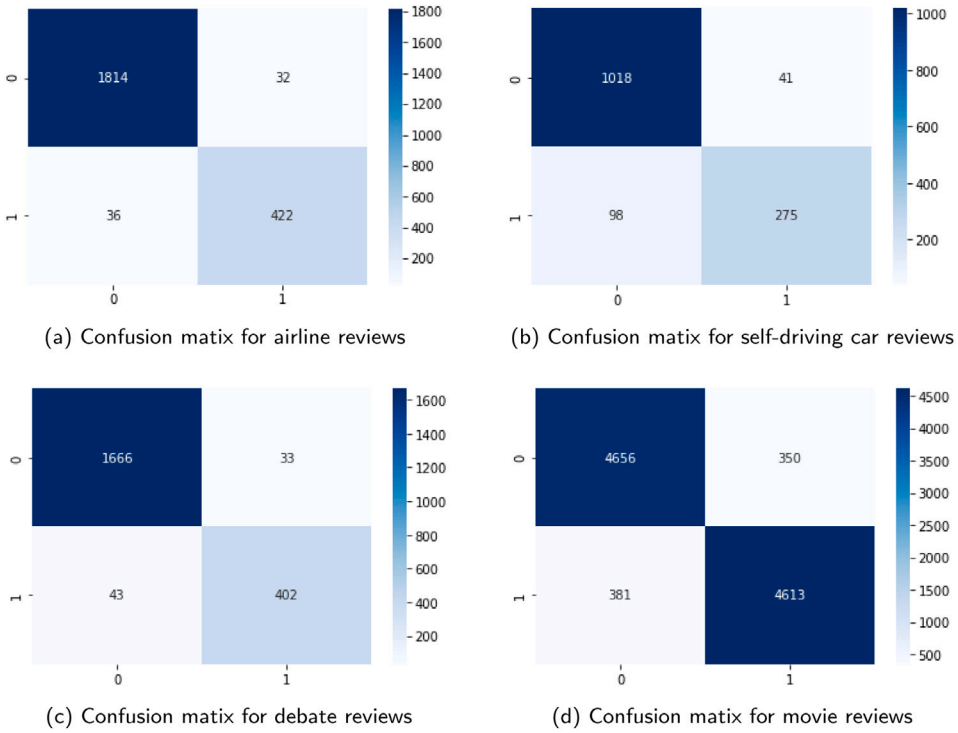


Fig. 8. Confusion matrix of Proposed model for datasets.

Similarly, the experimental results for the self-driving car have been shown in Table 5. The proposed CBRNN model has beat all other comparative models in terms of recall, f-score, accuracy, and AUC, and obtained 0.91%, 0.94%, 0.90% and 0.958% respectively. Just like the US-airline dataset, the obtained precision rate of 0.96% for the BERT-base CBRNN is closer to the precision rate of the Word2vec-based LSTM. The proposed model has the highest accuracy rate than other baseline models. Table 6 has shown the results of all performance measures on US-presidential election reviews. The proposed model obtained the highest scores of recall, f-score, accuracy, and AUC. For the election reviews dataset, the Word2vec-based LSTM and BERT-based CNN model have achieved 0.98% score, but it is observed that they are biased towards positive predictions. Similarly, the BERT-based CBRNN model has got the precision rate of 0.98% and recall rate 0.98%, which shows that the model is not biased towards any class. A comparative analysis of IMDB dataset has been shown in Table 7. The results of different classifier on this movie reviews dataset is clearly stated in table. The proposed model also obtained the highest scores for

Table 6
Evaluation parameters for US-presidential election dataset.

Embeddings	Models	Precision	Recall	F-score	Accuracy	AUC
Word2vec	CNN	0.96	0.90	0.93	0.89	0.922
	LSTM	**0.98**	0.86	0.92	0.86	0.920
	Co-LSTM	0.96	0.92	0.94	0.90	0.934
Glove	CNN	0.90	0.88	0.90	0.90	0.895
	LSTM	0.91	0.89	0.85	0.83	0.901
	Co-LSTM	0.96	0.89	0.94	0.86	0.950
BERT	CNN	**0.98**	0.93	0.95	0.92	0.941
	LSTM	0.97	0.94	0.95	0.93	0.948
	CBRNN	**0.98**	**0.97**	**0.97**	**0.96**	**0.973**

Table 7
Evalution parameters for IMDB dataset.

Embeddings	Models	Precision	Recall	F-score	Accuracy	AUC
Word2vec	CNN	0.80	0.82	0.81	0.82	0.894
	LSTM	0.74	0.78	0.76	0.77	0.863
	Co-LSTM	0.83	0.83	0.83	0.83	0.920
Glove	CNN	0.76	0.78	0.75	0.76	0.835
	LSTM	0.80	0.88	0.82	0.83	0.841
	Co-LSTM	**0.96**	0.89	0.89	0.89	0.901
BERT	CNN	0.89	0.90	0.89	0.89	0.901
	LSTM	0.90	0.91	0.90	0.90	0.923
	CBRNN	0.93	**0.92**	**0.93**	**0.93**	**0.969**

all matrices, which shows the effectiveness BERT-based CBRNN model. Eq. (6) visually illustrates the performance of the top three models for all datasets.

The ROC is a graphical representation of the classifier's performance, constructed by finding the difference between False Positive Rate (FPR) and True Positive Rate (TPR). Fig. 7a–d, depicts the comparison of the top three CNN–LSTM hybrid models using three state-of-art embedding techniques. The plot of ROC curve is split into two parts, x-axis and y-axis, having TPR and FPR respectively. It is the best method to identify the optimal model for SA and contains a range from 0 to 1. When the skewness of the curve is high to a genuine positive score, then the classifier is considered to have higher efficiency. Fig. 7 has shown a comparative analysis of ROC curves between Word2vec-base CO-LSTM (orange line), Glove-based Co-LSTM (green line), and BERT-based CBRNN (blue line) models. It is noticed that the blue curve of the proposed approach is closer to FPR, indicating the highest rate of TPR and lowest rate of FPR.

Fig. 8a–d shows a pictorial representation of the confusion matrix for airline reviews, car reviews, debate reviews, and movie reviews. The results of confusion matrices indicate that the proposed model minimized the difference between actual and predicted labels. It represents the obtained values for TP, TN, FP, and FN. So, the experimental results of the BERT-based CBRNN model shows the superiority of the model.

6. Conclusion and future work

In this paper, an enhanced feature extraction and classification model using BERT model and dilated convolutional Bi-LSTM model. A BERT-based CBRNN SA model has been proposed for sentence-level classification. The data were annotated using zero-shot BERT, then a pre-trained BERT model was employed to obtain sentence-level semantics and contextual features from the data. Then, obtained contextual embeddings had been passed to the neural network, comprised of dilated convolution and Bi-LSTM. Dilated CNN is used for extracting local and global information. Bi-LSTM is used for capturing the long-term sequencing of the sentences. The proposed hybrid CBRNN model was applied on four diverse domain datasets namely US-airline reviews, self-driving car reviews, US-presidential election reviews, and movie reviews. The performance of the CBRNN model is evaluated using five statistical measures, such as accuracy, precision, f1-score, recall, and AUC. The obtained results are then compared with the most commonly used embedding models, such as glove and word2vec. The proposed model obtained significant improvement in f1-score 0.2%, accuracy 0.3% and AUC 0.4%. The experimental results concluded that the proposed CBRNN model is more efficient as compared to the other models. Finally, the BERT-based CBRNN model can applied in industries for performing SA of their products.

As a future direction, the proposed technique can be applied to other resource-poor languages. Furthermore, another future direction is to implement our model on multi-class classification problems.

CRediT authorship contribution statement

Sayyida Tabinda Kokab: Conceptualization of this study, Methodology, Software. **Sohail Asghar:** Concept, Design, Analysis, Writing – review & editing. **Shehneela Naz:** Concept, Design, Analysis, Writing – review & editing.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work.

References

[1] Iqbal F, Hashmi JM, Fung BC, Batool R, Khattak AM, Aleem S, et al. A hybrid framework for sentiment analysis using genetic algorithm based feature reduction. IEEE Access 2019;7:14637–52.
[2] Rani S, Kumar P. Deep learning based sentiment analysis using convolution neural network. Arab J Sci Eng 2019;44(4):3305–14.
[3] Jindal K, Aron R. A systematic study of sentiment analysis for social media data. Mater Today: Proc 2021.
[4] Dang NC, Moreno-García MN, De la Prieta F. Sentiment analysis based on deep learning: A comparative study. Electronics 2020;9(3):483.
[5] Alaei AR, Becken S, Stantic B. Sentiment analysis in tourism: capitalizing on big data. J. Travel Res. 2019;58(2):175–91.
[6] Xu G, Meng Y, Qiu X, Yu Z, Wu X. Sentiment analysis of comment texts based on bilstm. Ieee Access 2019;7:51522–32.
[7] Çalı S, Balaman ŞY. Improved decisions for marketing, supply and purchasing: Mining big data through an integration of sentiment analysis and intuitionistic fuzzy multi criteria assessment. Comput Ind Eng 2019;129:315–32.
[8] Zhang Y, Zhang Z, Miao D, Wang J. Three-way enhanced convolutional neural networks for sentence-level sentiment classification. Inform Sci 2019;477:55–64.
[9] Berka P. Sentiment analysis using rule-based and case-based reasoning. J Intell Inf Syst 2020;1–16.
[10] Choi G, Oh S, Kim H. Improving document-level sentiment classification using importance of sentences. Entropy 2020;22(12):1336.
[11] Nazir A, Rao Y, Wu L, Sun L. Issues and challenges of aspect-based sentiment analysis: a comprehensive survey. IEEE Trans Affect Comput 2020.
[12] Machová K, Mikula M, Gao X, Mach M. Lexicon-based sentiment analysis using the particle swarm optimization. Electronics 2020;9(8):1317.
[13] Chauhan P, Sharma N, Sikka G. The emergence of social media data and sentiment analysis in election prediction. J Ambient Intell Humaniz Comput 2021;12(2):2601–27.
[14] Saad AI. Opinion mining on US airline Twitter data using machine learning techniques. In: 2020 16th International computer engineering conference. IEEE; 2020, p. 59–63.
[15] Meenakshi AB, Intwala N, Sawant V. Sentiment analysis of amazon mobile reviews. ICT Syst Sustain: Proc ICT4SD 2019, Volume 1 2020;1077:43.
[16] Ahuja R, Chug A, Kohli S, Gupta S, Ahuja P. The impact of features extraction on the sentiment analysis. Procedia Comput Sci 2019;152:341–8.
[17] Drus Z, Khalid H. Sentiment analysis in social media and its application: Systematic literature review. Procedia Comput Sci 2019;161:707–14.
[18] Keyvanpour M, Zandian ZK, Heidarypanah M. OMLML: A helpful opinion mining method based on lexicon and machine learning in social networks. Soc Netw Anal Min 2020;10(1):1–17.
[19] Basiri ME, Kabiri A. HOMPer: A new hybrid system for opinion mining in the Persian language. J Inf Sci 2020;46(1):101–17.
[20] Hernández-Rubio M, Cantador I, Bellogín A. A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. User Model User-Adapt Interact 2019;29(2):381–441.
[21] Chaovalit P, Zhou L. Movie review mining: A comparison between supervised and unsupervised classification approaches. In: Proceedings of the 38th annual hawaii international conference on system sciences. IEEE; 2005, p. 112c.

[22] Jianqiang Z, Xiaolin G, Xuejun Z. Deep convolution neural networks for twitter sentiment analysis. IEEE Access 2018;6:23253–60.

[23] Chauhan UA, Afzal MT, Shahid A, Moloud A, Basiri ME, Xujuan Z. A comprehensive analysis of adverb types for mining user sentiments on amazon product reviews. World Wide Web 2020;23(3):1811–29.

[24] Vijayaraghavan S, Basu D. Sentiment analysis in drug reviews using supervised machine learning algorithms. 2020, arXiv preprint arXiv:2003.11643.

[25] Ullah MA, Marium SM, Begum SA, Dipa NS. An algorithm and method for sentiment analysis using the text and emoticon. ICT Express 2020;6(4):357–60.

[26] Kalaivani K, Uma S, Kanimozhiselvi C. A review on feature extraction techniques for sentiment classification. In: 2020 Fourth international conference on computing methodologies and communication. IEEE; 2020, p. 679–83.

[27] Kowsari K, Jafari Meimandi K, Heidarysafa M, Mendu S, Barnes L, Brown D. Text classification algorithms: A survey. Information 2019;10(4):150.

[28] Kulkarni N, et al. A comparative study of word embedding techniques to extract features from text. Turk J Comput Math Educ (TURCOMAT) 2021;12(12):3550–7.

[29] Pham DH, Le AC. Exploiting multiple word embeddings and one-hot character vectors for aspect-based sentiment analysis. Internat J Approx Reason 2018;103:1–10.

[30] Rezaeinia SM, Rahmani R, Ghodsi A, Veisi H. Sentiment analysis based on improved pre-trained word embeddings. Expert Syst Appl 2019;117:139–47.

[31] Ay Karakuş B, Talo M, Hallaç IR, Aydin G. Evaluating deep learning models for sentiment classification. Concurr Comput: Pract Exper 2018;30(21):e4783.

[32] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. 2013, p. 3111–9.

[33] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing. 2014, p. 1532–43.

[34] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. 2013, arXiv preprint arXiv:1301.3781.

[35] Naseem U, Razzak I, Khan SK, Prasad M. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. Trans Asian Low-Resource Lang Inf Process 2021;20(5):1–35.

[36] Jiao Q, Zhang S. A brief survey of word embedding and its recent development. In: 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Vol. 5. IEEE; 2021, p. 1697–701.

[37] Wang S, Zhou W, Jiang C. A survey of word embeddings based on deep learning. Computing 2020;102(3):717–40.

[38] Colón-Ruiz C, Segura-Bedmar I. Comparing deep learning architectures for sentiment analysis on drug reviews. J Biomed Inform 2020;110:103539.

[39] Kasri M, Birjali M, Beni-Hssane A. Word2sent: A new learning sentiment-embedding model with low dimension for sentence level sentiment classification. Concurr Comput: Pract Exper 2021;33(9):e6149.

[40] Hayashi T, Fujita H. Word embeddings-based sentence-level sentiment analysis considering word importance. Acta Polytechnica Hungarica 2019;16(7):152.

[41] Fauzi MA. Word2vec model for sentiment analysis of product reviews in Indonesian language. Int J Electr Comput Eng 2019;9(1):525.

[42] Youbi F, Settouti N. Convolutional neural networks for opinion mining on drug reviews. In: Proceedings of the 1st international conference on intelligent systems and pattern recognition. 2020, p. 33–8.

[43] Yadav A, Vishwakarma DK. Sentiment analysis using deep learning architectures: A review. Artif Intell Rev 2020;53(6):4335–85.

[44] Dragoni M, Petrucci G. A neural word embeddings approach for multi-domain sentiment analysis. IEEE Trans Affect Comput 2017;8(4):457–70.

[45] Kumar H, Harish B, Darshan H. Sentiment analysis on IMDb movie reviews using hybrid feature extraction method. Int J Interact Multimedia Artif Intell 2019;5(5).

[46] Shen J, Liao X, Tao Z. Sentence-level sentiment analysis via BERT and BiGRU. In: 2019 International conference on image and video processing, and artificial intelligence. 11321, International Society for Optics and Photonics; 2019, p. 113212S.

[47] Keh SS, Cheng I, et al. Myers-briggs personality classification and personality-specific language generation using pre-trained language models. 2019, arXiv preprint arXiv:1907.06333.

[48] Fourati C, Messaoudi A, Haddad H. TUNIZI: A Tunisian Arabizi sentiment analysis dataset. 2020, arXiv preprint arXiv:2004.14303.

[49] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Trans Assoc Comput Linguist 2017;5:135–46.

[50] Kim H, Jeong Y-S. Sentiment classification using convolutional neural networks. Appl Sci 2019;9(11):2347.

[51] Ishaq A, Umer M, Mushtaq MF, Medaglia C, Siddiqui HUR, Mehmood A, et al. Extensive hotel reviews classification using long short term memory. J Ambient Intell Humaniz Comput 2020;1–11.

[52] Wei J, Liao J, Yang Z, Wang S, Zhao Q. Bilstm with multi-polarity orthogonal attention for implicit sentiment analysis. Neurocomputing 2020;383:165–73.

[53] Miao YL, Cheng WF, Ji YC, Zhang S, Kong YL. Aspect-based sentiment analysis in Chinese based on mobile reviews for BiLSTM-CRF. J Intell Fuzzy Systems 2021;(Preprint):1–11.

[54] Shen Q, Wang Z, Sun Y. Sentiment analysis of movie reviews based on cnn-blstm. In: International conference on intelligence science. Springer; 2017, p. 164–71.

[55] Min Z. Drugs reviews sentiment analysis using weakly supervised model. In: 2019 IEEE international conference on artificial intelligence and computer applications. IEEE; 2019, p. 332–6.

[56] Rehman AU, Malik AK, Raza B, Ali W. A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis. Multimedia Tools Appl 2019;78(18):26597–613.

[57] Soubraylu S, Rajalakshmi R. Hybrid convolutional bidirectional recurrent neural network based sentiment analysis on movie reviews. Comput Intell 2021;37(2):735–57.

[58] Behera RK, Jena M, Rath SK, Misra S. Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data. Inf Process Manage 2021;58(1):102435.

[59] Li W, Qi F, Tang M, Yu Z. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. Neurocomputing 2020;387:63–77.

[60] Alam M, Abid F, Guangpei C, Yunrong L. Social media sentiment analysis through parallel dilated convolutional neural network for smart city applications. Comput Commun 2020;154:129–37.

[61] Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. 2015, arXiv preprint arXiv:1511.07122.

[62] Yang X, Macdonald C, Ounis I. Using word embeddings in twitter election classification. Inf Retr J 2018;21(2):183–207.

[63] Rustam F, Ashraf I, Mehmood A, Ullah S, Choi GS. Tweets classification on the base of sentiments for US airline companies. Entropy 2019;21(11):1078.

[64] Zulqarnain M, Ghazali R, Ghouse MG, Mushtaq MF. Efficient processing of GRU based on word embedding for text classification. JOIV: Int J Inform Vis 2019;3(4):377–83.

[65] Sachin S, Tripathi A, Mahajan N, Aggarwal S, Nagrath P. Sentiment analysis using gated recurrent neural networks. SN Comput Sci 2020;1(2):1–13.

[66] Dutta A, Das S. Tweets about self-driving cars: Deep sentiment analysis using long short-term memory network (LSTM). In: International conference on innovative computing and communications. Springer; 2021, p. 515–23.

[67] Rustam F, Khalid M, Aslam W, Rupapara V, Mehmood A, Choi GS. A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis. Plos One 2021;16(2):e0245909.

[68] Li M, Chen L, Zhao J, Li Q. Sentiment analysis of Chinese stock reviews based on BERT model. Appl Intell 2021;51(7):5016–24.

[69] Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization. Adv Neural Inf Process Syst 2014;27:2177–85.

[70] Tang D, Wei F, Yang N, Zhou M, Liu T, Qin B. Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (Vol. 1: Long papers). 2014, p. 1555–65.

[71] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. J Mach Learn Res 2011;12(ARTICLE):2493–537.

[72] Cambria E, Havasi C, Hussain A. Senticnet 2: A semantic and affective resource for opinion mining and sentiment analysis. In: Twenty-fifth international FLAIRS conference. 2012.

[73] Vijayarani S, Janani R, et al. Text mining: open source tokenization tools-an analysis. Adv Comput Intell: Int J (ACII) 2016;3(1):37–47.

[74] Schuster M, Nakajima K. Japanese and korean voice search. In: 2012 IEEE international conference on acoustics, speech and signal processing. IEEE; 2012, p. 5149–52.

[75] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018, arXiv preprint arXiv:1810.04805.

[76] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems. 2017, p. 5998–6008.

[77] Siami-Namini S, Tavakoli N, Namin AS. The performance of LSTM and BiLSTM in forecasting time series. In: 2019 IEEE international conference on big data. IEEE; 2019, p. 3285–92.

[78] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: Icml. 2010.

[79] Ide H, Kurita T. Improvement of learning for CNN with ReLU activation by sparse regularization. In: 2017 International joint conference on neural networks. IEEE; 2017, p. 2684–91.

[80] Chou CN, Shie CK, Chang FC, Chang J, Chang EY. Representation learning on large and small data. Big Data Anal Large-Scale Multimed Search Wiley, Hoboken, NJ 2019;3–30.

[81] Wan Y, Gao Q. An ensemble sentiment classification system of twitter data for airline services analysis. In: 2015 IEEE international conference on data mining workshop. IEEE; 2015, p. 1318–25.

[82] Chen LC, Barron JT, Papandreou G, Murphy K, Yuille AL. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 4545–54.

[83] Bifet A, Frank E. Sentiment knowledge discovery in twitter streaming data. In: International conference on discovery science. Springer; 2010, p. 1–15.