Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data

Article *in* Information Processing and Management · January 2021

DOI: 10.1016/j.ipm.2020.102435

4 authors, including:

Ranjan Kumar Behera
National Institute of Technology Rourkela

Santanu Kumar Rath
National Institute of Technology Rourkela

Sanjay Misra
Østfold University College

Co-LSTM: Convolutional LSTM Model for Sentiment Analysis in Social Big Data

Ranjan Kumar Behera[1], Monalisa Jena[2], Santanu Kumar Rath[3], Sanjay Misra[4]

[1,3] *Department of Computer Science & Engineering, National Institute of Technology, Rourkela, India, 769008*
[2] *Department of Information and Communication Technology, F. M. University Balasore, Odisha, India*
[4] *Department of Electrical and Information Engineering, Covenant University, Ota 1023, Nigeria*
[4] *Department of Computer Engineering, Atilim University, Ankara Turkey*
jranjanb.19@gmail.com[1], *bmonalisa.26@gmail.com*[2], *skrath@nitrkl.ac.in*[3], *sanjay.misra@covenantuniversity.edu.ng*[4]

Abstract

Analysis of consumer reviews posted on social media is found to be essential for several business applications. Consumer reviews posted in social media are increasing at an exponential rate both in terms of number and relevance, which leads to big data. In this paper, a hybrid approach of two deep learning architectures namely Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) (RNN with memory) is suggested for sentiment classification of reviews posted at diverse domains. Deep convolutional networks have been highly effective in local feature selection, while recurrent networks (LSTM) often yield good results in the sequential analysis of a long text. The proposed Co-LSTM model is mainly aimed at two objectives in sentiment analysis. First, it is highly adaptable in examining big social data, keeping scalability in mind, and secondly, unlike the conventional machine learning approaches, it is free from any particular domain. The experiment has been carried out on four review datasets from diverse domains to train the model which can handle all kinds of dependencies that usually arises in a post. The experimental results show that the proposed ensemble model outperforms other machine learning approaches in terms of accuracy and other parameters.

Keywords: Deep Learning; Big Data; Sentiment Analysis; Word Embedding; RNN; CNN; LSTM

1. Introduction

Social media provides an extraordinary platform for big data analytics in various real-world applications. A massive amount of data is continuously generated when users are posting their views or opinions while communicating with each other through various social platforms like Twitter, Facebook, Myspace, etc. Social data is one of the big data generated from various social channel, poses all the three big data characteristics like velocity, heterogeneity, large-volume. Apart from these, it possesses a unique characteristic known as semantic, which refers to the fact that it is generated manually and contains symbolic information having inherent subjective meaning. This unique characteristic of social big data leads to several challenges and opportunities for sentiment analysis. Sentiment analysis (SA) is found to be an emerging research direction since early 2000. Various terminologies like opinion mining, sentiment classification, review mining, sentiment mining, opinion extraction are also used for sentiment analysis. It is the way of predicting attitude towards numerous products or social entities from sentiments. The source of sentiment analysis often varies from textual to visual representations. The sentiments involved in social media are certainly a source for modeling business strategies to achieve the business goal. It is often used for managing the online reputation of a specific product or brand. However, as the amount of data in social media repository is increasing at an exponential rate, the traditional algorithms often fail to extract the sentiments from such big data. Affective computing is one of the emerging research applications of sentiment analysis which able to capture the public sentiments automatically from the social media posts [1]. Sentiment analysis can be treated as a classification task as it classifies the orientation of a text into either positive, negative or neutral. Some of the widely adopted approaches towards big data sentiment analysis of unstructured data can be categorized into lexicon-based, linguistic-based, or machine-learning-based approaches.

The classification task involved in SA can be categorized into four different domains such as subjectivity classification, word sentiment classification, document sentiment classification, and opinion extraction. Subjectivity classification intends to classify sentences as subjective or objective. The subjective level of a sentence indicates that the particular sentence is an opinion about a topic or subject whereas objective classification infers the factual information associated with the sentences. It is one of the sub-categories of sentence-level sentiment analysis (SA). In document-level classification, the whole document is treated as a unit for sentiment analysis. The techniques involved in sentence-level sentiment analysis are not different from document level SA, as they can also be treated as mini-documents. Word sentiment classification determines the polarity of a sentiment involved with the particular word.

Sentiment analysis can be categorized based on the dataset used for processing. The major sources of data are from the public reviews associated with a product, organization, movie, or any other social entity. These reviews are important to business analytics as it plays a vital role in taking decisions about their products. Sentiment analysis is not only applied to product reviews but can also be applied on stock prediction, movie review, news articles, or political debates. For example, in political debates, it may be desired to figure out the opinions of voters on certain electoral candidates or political party. The election results may be heavily influenced by predicting the sentiment of users from their political posts. Various micro-blogging and social network websites are found to be rich sources of information, as people post their opinions and thoughts for discussion about a certain topic freely, which can be used as valuable resources in sentiment analysis. In this paper, social media reviews of various domains like the airline, movies, self-driving car, and election data are considered for modeling the architecture for sentiment analysis. Severyn and Moschitti [2] have shown in their paper that traditional machine learning algorithms perform well for classification and regression tasks for small size dataset. However, deep networks are more suitable for processing big social media, especially in the area of text classification. Wang et al. [3] and Yin et al. [4] observed that both feature detection and dependency capturing for long sentences are necessary to accurately classify a sentence.

The major contribution of this paper can be stated as below:

- In this paper, an effort has been made to develop an effective deep neural network architecture for sentiment analysis which can process big social data in a scalable manner without compromising with the performance. To address the issue, the functionality of both CNN and RNN is leveraged in the proposed Co-LSTM model for sentiment classification. The CNN model is mainly used for deep learning which automatically extract the features from the big social data instead of manual intervention as in case of traditional machine learning models. They are able to tune the hyper-parameters of the classifier model automatically which makes the model scalable to handle big data. In the proposed approach, the CNN model is used for better feature extraction through a pooling process and the LSTM is adopted for capturing the long term dependency among words in a sentence.
- The second contribution is to develop the sentiment analysis model, which should be domain-independent. To address the issue, we have trained the deep learning architecture using reviews from four different domains where almost all kinds of word dependencies exist and evaluated the performance separately for each dataset. The reviews from various domains like the movie, airline, self-driving car, and presidential election data have been considered to develop a generalized classifier which does not need any domain-specific knowledge.

The following sections of the paper are organized as follows: In section 2, the motivation towards the hybridization of CNN and LSTM has been discussed. Section 3 brings out the literature survey on techniques involves in sentiment analysis. The methodology adopted in the study is presented in section 4. Step by step process of the proposed algorithm is discussed in section 5. Evaluation parameters for the algorithm have been discussed in section 6. The implementation and results have been discussed in section 7. In section 8, the conclusion and future work for the paper are presented. Section 9 pointed out a few statements on the threat to the validation of the work.

2. Motivation

The motivation towards the research work may be described as follow:

- In the present world of digitization, social media available on the web is a big source of customer interactions and reviews. Sentiment analysis of such a huge amount of data helps to identify and track customer behavior about products, services, or brands [5]. Customer feedback is essentially required in the decision-making process. For example, customer reviews about an e-commerce product can help a new user to decide on the product before buying it. The same approach is also applicable for movie reviews, as they help users in deciding the movie for watching. Also for business, one can study the sentiments of a specific product or brand in specific demographic areas to identify the potential customers or the business potential of the new product or service in that area. Thus the sentiment analysis helps to enhance the business of an enterprise. Likewise, there are several applications of SA, which are helpful in our day to day activities [6].

- The opinion mining or sentiment analysis in social media has some major hurdles associated with it. One of the biggest challenges is the authentication of the end-user, where there is the possibility of incorporation of noise in the data acquired. Another major hurdle is inconsistency in social media data. The expression of sentiments and wording styles vary from person to person. People sometimes use shorthand notations, which make it difficult for the classifier to properly distinguish between word features. For example words like 'us' can be used for 'United States of America' as well as a pronoun, thus the classifier might get confused between 'us' as a pronoun or 'us' as a name or noun. Generally, no proper grammar and spelling protocols are often followed while writing the reviews in social media. Sometimes people use an acronym that makes the analysis more complicated. Social media sentiment analysis poses several challenges in handling noises like special characters, informal words, etc. Apart from that, it also contains sentences which involve sarcasm, different kind of negation statements, ambiguous words, multi-polarity word, etc. Some of the solutions to handle sarcasm for sentiment analysis is described in work presented by Maynard et al. [7]. The other major challenges are the cleaning and preprocessing of the sheer volume of data based on the context of reviews. It thus needs to have domain-specific knowledge for feature engineering of those data and make a proper transformation in the preprocessing phase, which is a cumbersome task. We have also motivated by few papers based on domain-independent sentiment analysis which were authored by Biyani et al. [8] and Bagheri et al. [9].

- Social big data is found to be a potential resource for sentiment analysis as it involves human sentiments on a specific topic, or product. It involves lots of sarcasm and dependencies which need to be exploited for predicting sentiments accurately. It also consists of short text and proverb, where actual sentiment is quite challenging to predict. A number of statistical learning approaches already exist for sentiment analysis. However, its performance highly depends on the quality of features, extracted from the review. It usually requires expertise in feature engineering, and it is also expensive in terms of computational time and space. A neural network can reduce the burden of proper feature engineering. CNN can able to exploit the parallelism in extracting the local correlations and patterns from the text as the computation at a time step doesn't depend on the computation at the previous time step. In this paper, we have adopted CNN for better feature engineering for the big social review data. However, it may not be suitable for capturing the contextual information from a given review as it doesn't remember the past context. We have adopted LSTM which is mainly suitable to capture the temporal contextual information. It is best suited for capturing the dependencies of words inside the reviews. It is mainly used for sequence prediction.

Some of the other architectures like simple multi-layer perceptron (MLP) and probabilistic neural network can be used for feature extraction, but they are not suitable for processing a large set of big social review data. These architectures are not suitable to capture sequential dependencies which are the essential parameters for sentiment classification. In the second phase, simple RNN can be used for classification but It suffers from vanishing gradient problem due to which it is quite difficult to train for the problem which requires long term temporal dependencies. This has motivated us to use CNN in the first phase and LSTM in the second phase.

3. Related Work

In sentiment analysis, the given text or review is analyzed, and it captures the prevalent emotional opinion within that text to identify the reviewers attitude as positive, negative, or neutral. Technically, it is the process of extracting the sentiment orientation of a text unit by using Natural Language Processing (NLP), statistics, or machine learning methods. Sentiment analysis plays a crucial role in social media monitoring, as it captures the public opinion about certain topics. Some of the pioneer works related to sentiment analysis are presented below:

Dos Santos and Gatti [10] have proposed an efficient CNN model to exploit the character to sentence information to classify the emotional level of the short text. They have proposed their model consisting of two layers of CNN which they named as character Conventional Neural network (ChCNN). Zhou et al. [11] have proposed a bi-directional LSTM model for sentiment analysis in which a two-dimensional pooling layer has been adopted. They have experimented on Stanford Sentiment Treebank (SST) database which resulted in 88.7% accuracy. Ma et al. [12] have proposed an extension of LSTM model termed as sentic LSTM for targeted aspect-based sentiment analysis. Their work mainly concerns with combining tasks of target-dependent aspect detection and aspect-based polarity classification. In another work, they have embedded common sense knowledge in the recurrent encoder for targeted sentiment analysis [13]. Their model is a hybridization of the attention architecture and Sentic LSTM. Wang et al. [14] have proposed a hybrid version of CNN and RNN for opinion analysis of the sentences. As the CNN model is independent of the location of a word in the sentence, both of the models have been worked as a layering fashion, i.e., the output of the CNN is fed into the input to the RNN model. Rao et al. [15] have proposed document-level sentiment analysis which captures the semantic relationships between the sentences in the document. They have proposed SSR-LSTM and SR-LSTM which are based on deep recurrent neural networks.

Hussain et al. [16] have shown the potential of the semi-supervised model, which hybridize the random projection scaling and support vector machine to perform reasoning in big social media. Their model seems to be quite suitable for extracting the semantic information from emoticon representation and polarity identification in knowledge based on big social data. Cambria et al. [17] have developed a three-level representation for sentiment analysis termed as SenticNet 5 which able to discover conceptual primitives automatically and the commonsense knowledge is embedded. An ensemble of top down and bottom up learning has been embedded in senticNet 6 which is based in symbolic and subsymbolic AI [18]. They have trained their model using WordNet-affect emoticon list, which is freely available on the Internet.

Sentiment analysis of customer reviews is based on a procedure, which may be called as a dichotomous one. The procedures followed in it can be categorized into three types, such as the Supervised method, Lexicon based method, and Semantic-based sentiment analysis. These are described as follows:

3.1. Supervised methods

In supervised methods, sentiments of reviews are predicted based on the labelled sentiments associated with the available review data [19]. The overall procedure is to predict sentiments based on the classification model using different machine learning techniques, which are trained on these available data after going through proper feature engineering. Qiang et al. [20] have presented a comparison of different supervised machine learning techniques for sentiment classification of travel destination review in the USA. There are different techniques available to carry out feature engineering and data transformations, such as n-gram [21], POS tagging known as Part-Of-Speech tagging [19] methods based on semantic patterns [22] and word-based semantic concepts [23].

The major limitation of supervised learning is that they are domain-specific i.e., the classifier models trained on restaurant reviews may not perfectly work on movie or product review [24]. Another limitation may be noted that the classifier needs a large amount of training data to cover all possible cases. Araque et al. [25] has proposed deep learning-based ensemble techniques for classifying sentiment using in the social application. In their work, they hybridized surface classifiers with linear machine learning algorithms. The feature processing has been carried out by combining deep and surface features from different domains.

3.2. Lexicon based methods

Lexicon based methods use the sentiment orientation of words or phrases existing in a review to evaluate the overall sentiment score. Based on the obtained sentiment score, the review is termed as either positive or negative. Hence, lexicon-based methods are based on counting the sentiment lexicons rather than training the data. The model will be more effective if the lexicon dictionary is associated with more number of words. There exist various in-built dictionaries with terms and associated sentiment orientations like SentiWordNet [26], MPQA subjectivity lexicon [27] and LIWC lexicon [28], etc. The major disadvantage of this approach is the associated cost in searching the sentiment orientation of each word in the in-built dictionary. Also, the sentiment orientation of a word may vary from domain to domain. This problem can be tackled if the sentiment orientation of a word concerning the semantics of its context is being considered [29]. But in the case of most of the lexicon-based approaches, the existence of syntactical features or words explicitly reflects the sentiment independent of the context in the document. Deep learning has been a popular trend in sentence-level sentiment analysis. Yoon et al. [30] proposed a multi-channel lexicon-based model which hybridize CNN with bidirectional LSTM for sentiment classification. The performance of their model is based on the set of rules extracted from the sentiment orientation of lexicon present in the context, which is domain-dependent. In this paper, the proposed hybridized model is domain-independent in which training-based approach is adopted for sentiment analysis rather than the lexicon-based approach. In this paper, normal LSTM model is adopted instead of bidirectional LSTM as bidirectional LSTM are found to be more complex and needs huge computational power. They also need to scan the entire review text to capture the context dependency, which makes computationally inefficient while processing huge size social media reviews. In their work, the multi-channel embedding layer has been used, which is based on the Word2Vec model.

3.3. Semantic based methods

Various types of semantic-based sentiment approaches have been proposed by several authors, which can broadly be classified as conceptual semantic and contextual semantic [31]. Co-occurrence patterns of words in the text are used to evaluate the semantics in the case of contextual semantics, which is also known as statistical semantics [32]. External semantic knowledge bases like semantic networks are used with natural language processing to conceptually represent the words to convey sentiments. SenticNet [33] is an example of the conceptual lexicon for sentiment analysis. Although conceptual semantic approaches have outperformed the contextual approaches in many cases, they are limited to their knowledge base domain. We et al. [34] have proposed a semantic approach for clustering words in a text-based on the lexical chain and WordNet model. In their work, WordNet is integrated with lexical chains to exploit the ontological structure for capturing the semantic relationship between the words in a cluster.

3.4. Research questions

In this paper, the following research challenges have been identified, and an effort has been to resolve the same using deep learning algorithms.

RQ1:. Review data in social media often consists of noisy elements like incorrect spellings, grammatical errors, product ids, hyperlinks. Sometimes they are rich with emoticons which make the task more difficult for sentiment analysis. Emoticons are not natural text like language. These are the textual symbols consisting of various characters representing a specific smiley face. Each of them is associated with some kinds of emotions (happy, sad, irritate, etc.). Handling emoticon is found to be challenging as compared to handing text which represents emotions. Sailunaz et al. [35] have presented a model for sentiment analysis that can classify sentences based on emoticons associated with the text. Emoticons are the essential elements for short text or small reviews. Is the classifier model able to handle noisy data and emoticons?

To address this issue, proper feature engineering is desirable before training the classifier models. A huge text corpus has been referred to identify the incorrect spelling from the reviews. Google-1 (Billion word Corpus) [36] has been used to handle the word which are misspelled. It is then replaced with one or two-letter distance words available in the text corpus. All the numerical digits have been replaced with the newly introduced word "digit". The hyperlinks are filtered out using a regular expression. Emojis are handled through the package known as emoji-sentiment-lexicon for replacement of the emoticons available in the review text. LSTM architecture in the proposed model is able to capture the context in which emoticons are used in the reviews.

RQ2:. The processed string can't directly be fed to a model for training as most of the learning algorithms require numerical vectors as input. Traditional approaches like Tf-Idf [37] or one-hot encoding for converting a string into a numerical value, provide a random numerical index to a word or phrase. The random numerical vector may not able to capture the actual context involve in the text. The research question may be frame as how to capture the context of corresponding words or phrases in their numerical representation?

In this paper, a word embedding layer is being considered to create a numerical feature matrix for capturing actual context present in the review text. Each word is being assigned a one-dimensional numerical vector that is self-trainable. Here the numerical vector is being constructed by passing through several training steps rather than by random assignment.

RQ3:. The feature matrix obtained from the word embedding layer, is passed through the convolutional neural network. The output of the convolutional layer is then provided as input to the neural architecture to predict the sentiment as positive or negative. Most of the conventional model for classification treats every feature of an input independently, which is not in the case of human originated reviews. How to capture the dependency between the words in a sentence for predicting actual sentiments?

To capture the sequential dependency or semantic representation of a review, a Long Short Term Memory (LSTM) layer is used. LSTM seems to be able to capture the long term dependency of words in the text with its unique architecture of having memory at each network.

4. Background Details

4.1. Word Embedding Techniques

Word embedding is the technique of converting text into numbers so that it can be used as input to the machine learning algorithms [38]. The same text is converted to different numerical formats following different procedures depending on the context it is used. The word embedding process is quite important in text processing as various machine learning or neural network techniques do not support operation on plain texts but only numbers. Technically, word embedding method maps a word to a vector, based on a dictionary, which may be trained over a text corpus using a neural network. Vector representation of a word can be of various types. One-hot encoding is a popular vector representation technique of words consists of binary number only. In this representation, if the position of a word in a sentence is n, the n^{th} position of the vector corresponding to the word is one, and rest values will be zero. For example, considering the sentence "social media research", the one-hot encoded vector for 'media' will be [0, 1, 0] since the word 'media' exists only in the second position of the sentence. Various types of word embedding techniques can be categorized into two classes, such as frequency-based embedding and prediction based embedding. Frequency-based word embedding techniques are based upon how frequently a word is used in the sentence [39]. Count-vectorizer, Tf-Idf vectorizer and co-occurrence matrix are some of the examples of frequency-based techniques [40]. The prediction-based techniques use previous information and neural network models to prepare the word vector based on the context [31]. CBOW (Continuous bag of words) and skip-gram model are the examples of this category [33].

4.2. Deep Learning Techniques

Deep learning is a representation learning technique that can itself process the raw input to be suitable for the classification or regression eliminating the use of feature engineering as in the case of conventional machine learning techniques. There are various deep learning models like the convolutional neural network (CNN), probabilistic neural network (PNN), recurrent neural network (RNN), etc.

4.2.1. Convolutional Neural Network (CNN)

CNN generally operates based on the convolution and sub-sampling process carried out through a series of layers [31]. It is then followed by one or more fully connected layers. All the operations performed in the CNN model passes through three sequential layers as follows:

- Convolution Layer: CNN has got such a name mainly because of the convolution operation performed. The Convolution process primarily helps in extracting features from input data. For example, if an image is considered as input then the convolution process extracts the features from the image with preserving the spatial relationship between pixels by learning image features using small squares (2-D filters) of input data. When it is applied in text classification, it helps in extracting the feature matrix by preserving high-level word or phrase representation.

- Pooling Layer: It is a good practice that when the size of the input is too large, it is desirable to reduce the number of trainable parameters. The feature dimension needs to be reduced without losing any important information. Pooling layers are periodically introduced between subsequent convolution layers. Pooling (also called sub-sampling or down-sampling) reduces the spatial size of each feature map but retains the most important information. Spatial Pooling can be of different types: max, average, sum, etc. In the case of max pooling, a spatial neighborhood (for example, a 22 window) is defined and take the largest element from the rectified feature map within that window. Instead of taking the largest element, average (average pooling) or sum of all elements in that window may also be considered for average and sum pooling respectively. In this paper, the max-pooling approach has been considered.

- Fully Connected Layer: The fully connected layer is a traditional multi-layer perceptron that uses a softmax activation function in the output layer. The term "fully connected" implies that every neuron in the previous layer is connected to every other neuron on the next layer. The output from the convolutional and pooling layers represent high-level features of the input data. The intuition behind the fully connected layer is to use these features for classifying the input into various classes based on the training dataset. Most of the features from convolutional and pooling layers seem to be good for the classification task.

4.2.2. Recurrent Neural Network (RNN)

In real-world scenarios, semantic information of one word often depends on the meaning associated with previous words in a text. CNN fails to process this dependency as they consider every word in the text independently. RNN may be the appropriate solution to capture the dependency. RNNs perform the sequential analysis by carrying out the same process recurrently for every element in the sequence. RNN possesses a memory to capture the information that has already been calculated which influences the result to be evaluated. The schematic diagram of RNN can be depicted as in Figure 1.

The process of RNN may be well represented through an example. Considering a text which consists of a sequence of three words. The network is unfolded to three layers (one layer for each word) as shown in, Figure 1. To visualize the computation consider X_t be the one-hot encoded vector of a word to be input at timestamp t. C_t be the cell state at timestamp t which acts as a memory for the network. Y_t is the output at timestamp t.

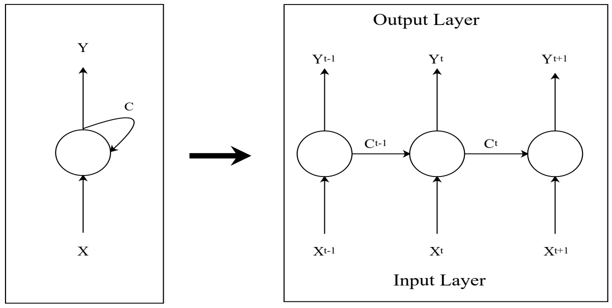


Figure 1: Schematic Flow Diagram for Recurrent Neural Network

4.2.3. *Long Short Term Memory (LSTM)*

LSTM is a sophisticated version of RNN used for sequential modeling mainly on text data. It can be considered as a special case of RNN where only the essential portion of data is being passed to the next layer instead of passing whole data. One of the major problems in a simple RNN network is the vanishing gradient problem [41] [42]. Gradient descent method is often used in neural networks to minimize the error by optimizing the weight value at each neuron. Usually, the gradient of the loss function decreases exponentially at subsequent steps through back-propagation in RNN, which is also known as gradient vanishing problem. For example, considering sentences like "I play cricket, and I am good at bowling", the word 'bowling' depends on the word 'cricket', which is far behind the former one in position. With the increase in distance between two dependent words, the performance of RNN often decreases, and also the gradient value vanishes significantly. The Long Short Term Memory (LSTM) overcomes this problem and performs well in long term dependency case.

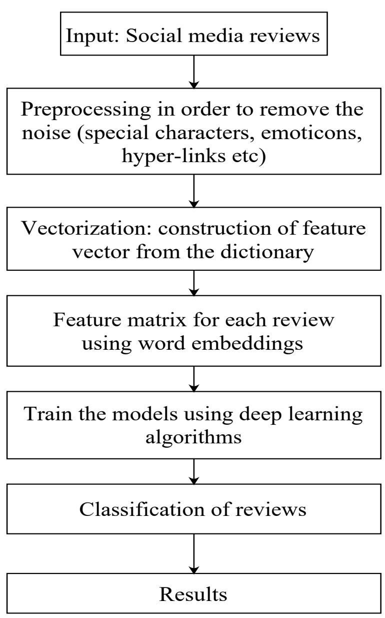


Figure 2: Schematic diagram of the proposed approach

5. Proposed model for sentiment analysis

The presented approach passes through the three layers, such as word embedding, Convolution, and LSTM layer. The schematic diagram of the proposed approach for sentiment analysis is presented in Figure 2. In the first layer, word-embedding is applied to embed the words in the review, which eradicates the domain dependency of the review features. The second phase uses the convolution layer and the pooling process in order to identify the important local and deep features in the sentence [43]. The third layer applies the LSTM network on the output obtained from the second layer to capture their sequential dependency from left to right. The combination of three layers helps in realizing the behavior of the sentence. The output of the LSTM is then supplied to the fully connected sigmoid layer to evaluate the result by considering binary cross-entropy as the loss function. The overall architecture of the classifier is shown in Figure. 7. The steps of the proposed approach are presented as follows:

Step 1: Preprocessing. Social media reviews often in the form of text which contain noisy data such as special characters, symbols, and hyperlinks, etc. The noisy information are filtered out with the help of regular expression. In the preprocessing stage, all the reviews are broken into tokens in the form of words. The duplicate words are then eliminated to construct a unique representation for each word. A vocabulary dictionary is then constructed with unique words as keys and words indices as values. Two new words such as “digit” and “unknown” are introduced to represent all the numerical digits and the words, which are not present in the dictionary, respectively. The process of vocabulary dictionary construction is shown in Figure. 3.

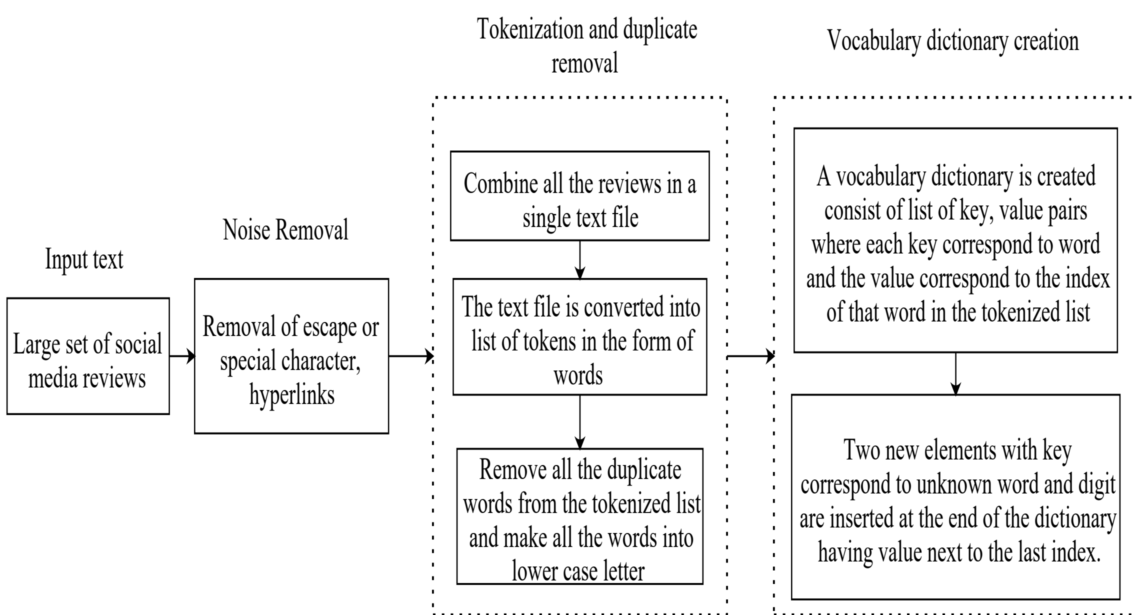


Figure 3: Vocabulary dictionary creation before feature vectorization

After preprocessing, text vectorization process has been carried out for each review. Each element of the vector representation of review corresponds to the index of the word in the vocabulary dictionary. The length of the vector has been fixed to 25. As most of the reviews are having word-length less than 25, the index of the newly introduced word unknown is padded at the end to make length 25. If the word-length of any review exceeds 25, the less significant features are removed, i.e., the word-length of the review is truncated to 25. The insignificant words are identified with the process of lemmatization and the stop word removal using the NLT package available in python. Most of the words in English have several alternative words with similar meaning. Lemmatization is the process of transforming alternative form to the base form which inherently reduces the number of words. The feature vectorization process is shown in Figure. 4.

Sometimes dimensional reduction is necessary to filter features for reducing the computational complexity. One intuitive example can be “awesomely amazing” may be mapped as only “amazing” as it reduces the input size without losing semantic information. We have adopted PCA for dimensional reduction of the feature metric, which is then passed into CNN and LSTM model as input. A novel architecture of convolution and pooling process has also been considered in order to feature filtering.

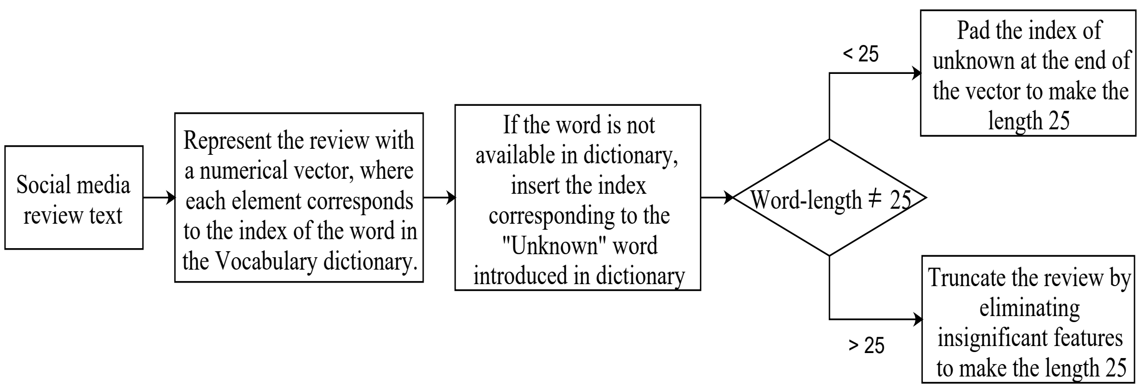


Figure 4: Process of feature vectorization

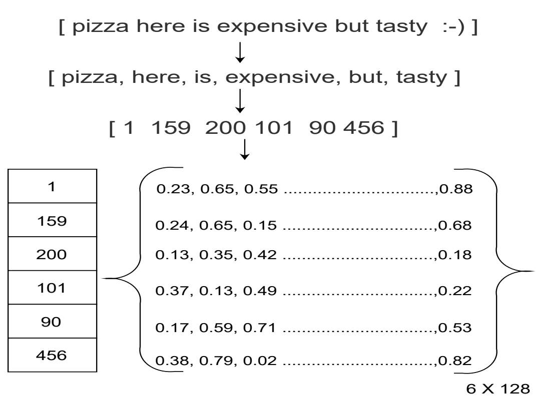


Figure 5: wordembedding for feature matrix construction

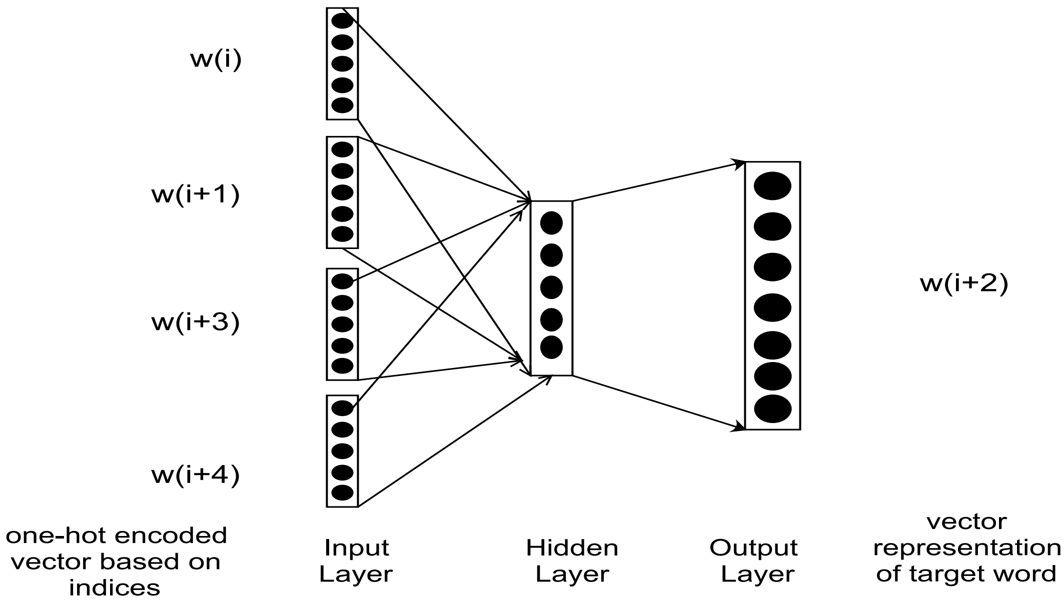


Figure 6: Schematic diagram of Word embedding model (CBOW)

Step 2: Word-embedding model. Each word in the list of texts is embedded to a vector of dimension 128 which is trained through the backpropagation process. Word2vec algorithm has been used for training the word embedding as it is simple and more efficient for vector representation. Word embedding is a model used to represent the review in textual format into numerical vector space which can be further process through neural networks. Prior to the representation, the vocabulary dictionary is created for the datasets considered. In vocabulary dictionary, each word is associated with a index which represents the position of the word in the dictionary. As the position of each word is unique in nature, we have leveraged it for vector representation of each review available in the dataset. The indices are used to represent the words in the vocabulary dictionary. These are used to construct the one-hot encoding representation, which are treated as input for word embedding model. A sample feature matrix constructed from the word embedding model is presented in Figure 5. The indices value for each of the word in Figure 5 are just an example. It may be varied from dataset to dataset. The values present inside the matrix are the randomly assigned weights for the embedding layers, which are adjusted through the backpropagation process. In word embedding model, CBOW is used which takes the context of the word as input and tries to predict the representation for the target word. Internally it uses three-layer feed forward neural network for constructing the numerical representation for words.The architectural diagram for word embedding model (CBOW) is presented in Figure 6. The schematic diagram for deep learning process is presented in Figure. 7.

Step 3: Convolutional Layer. In the convolution layer, seven filters each of size 3X3 with stride one are traversed over the input feature matrix to get the required features. Multiple filters have been used for extracting different types of features. For example; if a matrix of size 8x128 is traversed with the filter of

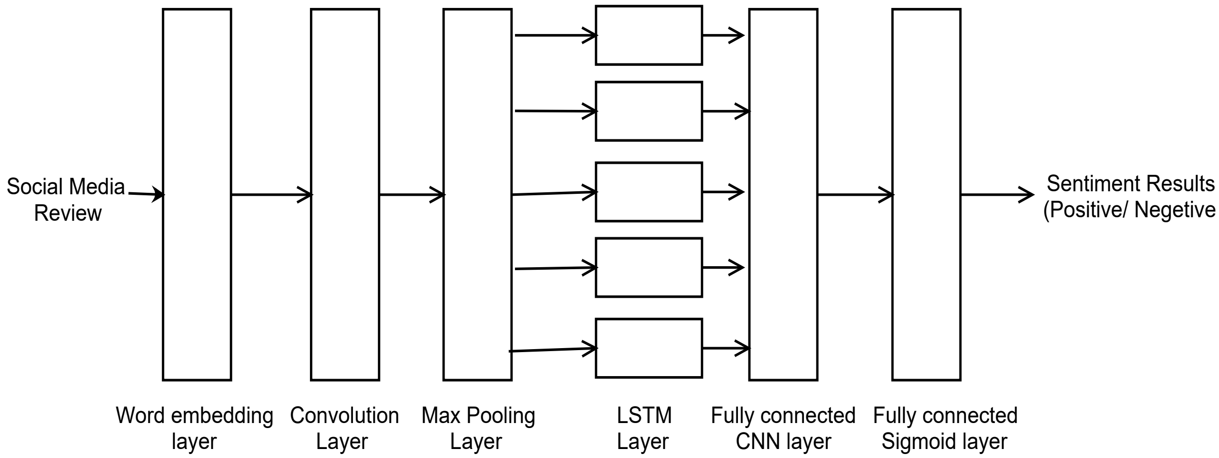


Figure 7: Schematic diagram of deep learning steps in the proposed sentiment analysis model

dimension 3x3, the convolution process will deliver a feature matrix of size 6x126. It captures all the local hidden features as shown in Figure 8. Rectified Linear Unit (ReLU) activation function has been used in the fully connected layer of CNN as it is found to be six times faster than the sigmoid and tanh activation function [44]. However, in the last layer, the sigmoid function is used to get the class label. The inputs to the last layer is the output of the last LSTM layer.

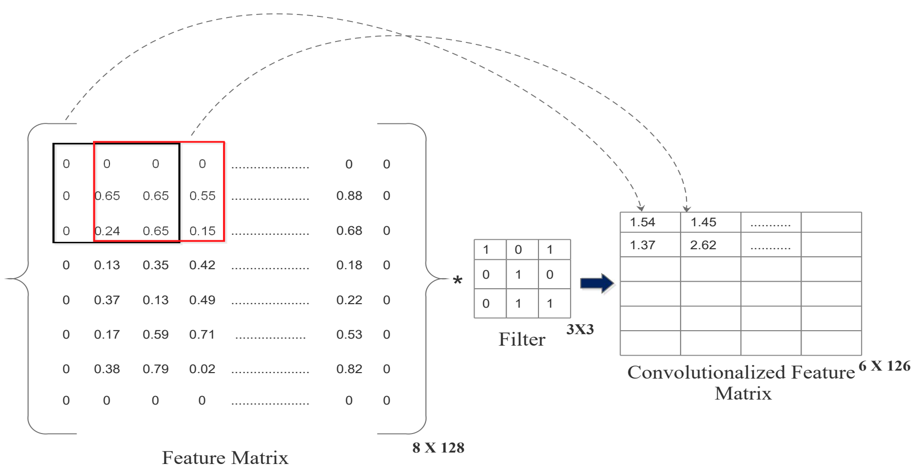


Figure 8: Convolution Process

Step 4: Maxpooling Layer. After getting a feature matrix of size u x v from the convolution layer, the max-pooling is performed with a filter of dimension 2x2. In max-pooling, the maximum feature value is selected at each position of the filter while traversing. The stride of size 2 is considered for traversing the filter. The obtained feature matrix is of dimension $\frac{u}{2}$x $\frac{v}{2}$. Max-pooling operation is performed for each convolution filter independently. Figure 9 shows the schematic structure of the Max-Pooling layer.

Step 5: Long Short Term Memory (LSTM) Network. The output from the max-pooling layer is passed to the LSTM layer to sequentially analyze the generated feature vectors from left to right. Since the important

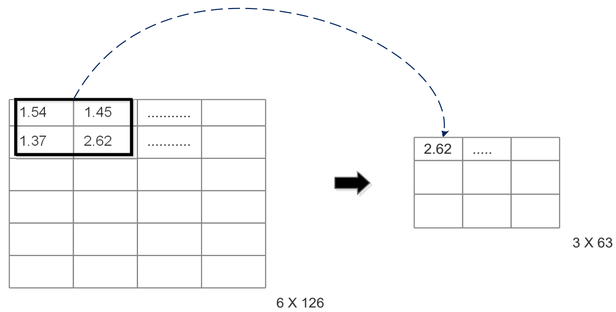


Figure 9: Max-Pooling Layer

local features have been extracted at the output of the max-pooling layer, the LSTM network is able to check the long term dependencies to detect the global features. The output of the LSTM layer is flattened to reduce the features, which is then passed through the fully connected CNN layer to predict the actual sentiment. In this work, a hundred number of LSTM networks have been applied with a ten percent dropout to avoid the over-fitting condition.

Step 6: Sigmoid Layer. The feature vectors obtained from the output of LSTM layer are passed to a fully connected sigmoid layer to find the probability distribution of each category. It can be mathematically defined as follows:

$$P_{sigmoid}(C_j) = \frac{e^{o_j}}{1 + e^{o_j}} \tag{1}$$

where $P_{sigmoid}(C_j)$ is the probability distribution for the category j and o_j represents the output corresponding to the category j. The Sigmoid activation function is used to normalize the confidence score of the classifier between zero to one. After getting the probability distribution from sigmoid layer, binary cross entropy is applied as loss function to calculate disparity between actual sentiments and predicted sentiments.

$$loss = -\sum_{i=1}^{k} R(C_i) \times log P_{sigmoid}(C_i) \tag{2}$$

where k is the number of categories and $R(C_i)$ is the actual sentiment associated with the text. It can take discrete value from the set $L = \{0, 1\}$, where L is the sentiment label of review text (Negative, Positive). It is similar to the likelihood function which seek to minimize the difference between probability distribution in the training set and the models predicted probability distribution of the testing dataset.

6. Implementation

6.1. Dataset used for Experiment

In this paper, four review datasets from diverse domains such as Movie review, Airline review, US presidential election review and self-driving car review have been considered for the experiment. As all of these are from different domains, the writing style of reviews are totally different from each other. Different kind of word dependencies may be available inside the post of reviews. As one of the contributions in this paper is to build a domain-independent sentiment analysis model, the model has been trained by merging the training set from all the datasets and evaluation has been carried out for each of the datasets separately. The confusion matrices presented in the result section is based on the testing part of individual dataset. All of these datasets are balanced in nature, i.e., the ratio of the number of samples belonging to positive, negative or neutral classes is equal or almost equal to each other. The description of the datasets are explained as follows:

1. Movie Review: The Large Movie Review Dataset (often referred to as the IMDB dataset) contains 25,000 highly polar moving reviews (good or bad) for training and the same amount again for testing. The problem is to determine whether a given moving review has a positive or negative sentiment. The data was collected by Stanford researchers and was used in a 2011 paper, where a split of 70:30 of the data was used for training and test [21].
2. Airline Review Dataset: This data originally came from Crowdflower's Data for Everyone library. It contains reviews about major U.S. airlines. The Twitter data was scraped from February 2013 to January 2014 in a paper by Wan et al. [45], and it is supervised as to classify positive, negative, and neutral tweets, followed by categorizing negative reasons (such as "late flight" or "rude service"). It contains whether the sentiment of the tweets in this set was positive, neutral, or negative for six US airlines:
3. Self Driving Car dataset: This dataset has been collected from the website "*https://www.kaggle.com/*" [46]. It has three attributes such as Twitter id, review text, and the polarity associated with the sentiment.
4. US Presidential Election Dataset: This data is collected from the website "*https://www.kaggle.com/*" [21]. It is the first GOP debate Twitter sentiment data that analyze tweets on the first 2016 GOP Presidential Debate. It consists of 21 attributes and 13871 number of reviews.

6.2. Performance Evaluation Parameters

The results obtained from the experiment have been discussed in this section. The proposed Co-LSTM model has been compared with the other machine learning models like SVM, Naive Bayes, Linear Regression, Random Forest, CNN, and RNN for validation. The performance of the proposed algorithm has been accessed in terms of accuracy, precision, recall, and F-measure which have been measured from the confusion matrix. The statistical test like a t-test has also been used to show how the proposed algorithm is significantly different from other algorithms. The ROC curve and AUC value are also presented for analyzing the performance of the proposed algorithm.

Table 1: Confusion Matrix

		Correct label	
		Positive	Negetive
Predicted label	Positive	True Positive (TP)	False Poitive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Confusion Matrix. Confusion matrix, also known as error matrix or contingency matrix is the visual representation of statistical values, obtained through experiments. It shows the statistics about the actual and predicted level for each review in the text for the classifier. It is used to evaluate the performance of most of the supervised machine learning algorithms. The confusion matrix for binary classification can be represented in the form, as shown in Table 1. In this study, the classification of reviews is labeled as either positive or negative sentiments. The confusion matrix has four components with the help of which the different performance parameters can be evaluated:

- True Positive (TP): It represents the reviews that are originally labeled as positive and also predicted as positive by the classifier.
- False Positive (FP): It represents the reviews that are originally labeled as negative but predicted as positive by the classifier.
- True Negative (TN): It represents the reviews that are originally labeled as negative and also predicted as negative by the classifier.
- False Negative (FN): It represents the reviews that are originally labeled as positive but predicted as negative by the classifier.

The performance of the proposed classifier has been evaluated based on the following parameters.

i. **Precision**: It is defined the ratio of true positive prediction to the total number of positive prediction. It measures the exactness of the classifier. It can be expressed as:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

ii. **Recall**: It is defined as the ratio between the number of true positive prediction to the total number of actual positive sample. It is also known as sensitivity.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

iii. **F-measure**: It is the harmonic mean of Precision and Recall.

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

iv. **Accuracy**: It is defined as the fraction of samples that are predicted correctly.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

6.3. Result Analysis and Discussion

Table 2: Confusion Matrix, Evaluation Parameters for Movie Review Dataset

Models		Confusion Matrix		Evaluation Parameter			
		Predicted Yes	Predicted No	Precision	Recall	F-Measure	Accuracy
SVM	Actual Yes	329	66	0.8329	0.8266	0.8298	0.8311
	Actual No	69	336				
		Predicted Yes	Predicted No				
Naive Bayes	Actual Yes	355	40	**0.8987**	0.7230	0.8014	0.7800
	Actual No	136	269				
		Predicted Yes	Predicted No				
Linear Regression	Actual Yes	318	77	0.8051	0.8010	0.8030	0.8050
	Actual No	79	326				
		Predicted Yes	Predicted No				
Random Forest	Actual Yes	302	93	0.7646	0.6028	0.6741	0.6350
	Actual No	199	206				
		Predicted Yes	Predicted No				
CNN	Actual Yes	316	79	0.8000	0.8294	0.8144	0.8200
	Actual No	65	340				
		Predicted Yes	Predicted No				
RNN	Actual Yes	296	99	0.7494	0.7810	0.7649	0.7725
	Actual No	83	322				
		Predicted Yes	Predicted No				
Co-LSTM	Actual Yes	330	65	0.8354	**0.8350**	**0.8302**	**0.8313**
	Actual No	70	335				

Standard machine learning models such as SVM, linear regression, random forest, and Naive Bayes are being considered for experimental comparison. Deep learning models are found to be more effective than machine learning algorithms. CNN and LSTM networks are considered as the basic framework for the proposed model, i.e., Co-LSTM. For classification of sentiment reviews efficiently, researchers have frequently come up with ensemble systems based on these architectures, and the experimental results reported in literature reflect the viability of the different techniques. Although the extensive study has been carried out using traditional models, a good amount of work has been carried out using deep learning models too in recent years. The latter is found to outperform the traditional systems in most of the cases, thereby establishing its utility in the field of natural language processing, including sentiment analysis. The performance results of various machine learning techniques have been presented in the confusion matrix form along with the evaluation parameters for each of the datasets.

Comparative analysis of the classification models based on precision, recall, f-measure, and accuracy for the movie review dataset is presented in Table 2. It can be observed that accuracy and f-measure for the proposed Co-LSTM model yield better results as compared to other algorithms. Naive Bayes and CNN model have better precision and recall value respectively for the movie review dataset as they are biased more towards positive sentiments. The top three models for movie review datasets in term of accuracy are found to be Co-LSTM, SVM, and CNN with 83.13%, 83.11%, and 82% respectively.

Table 3: Confusion Matrix, Evaluation Parameters for Airline Dataset

Models		Confusion Matrix		Evaluation Parameters			
		Predicted Yes	Predicted No	Precision	Recall	F-Measure	Accuracy
SVM	Actual Yes	3419	230	0.9370	0.9529	0.9449	0.9136
	Actual No	169	799				
		Predicted Yes	Predicted No				
Naive Bayes	Actual Yes	3646	3	**0.9992**	0.8135	0.8968	0.8183
	Actual No	836	132				
		Predicted Yes	Predicted No				
Linear Regression	Actual Yes	3611	38	0.9896	0.9007	0.9431	0.9056
	Actual No	398	570				
		Predicted Yes	Predicted No				
Random Forest	Actual Yes	3589	60	0.9836	0.8680	0.9221	0.8687
	Actual No	546	422				
		Predicted Yes	Predicted No				
CNN	Actual Yes	3553	96	0.9737	0.9449	0.9591	0.9344
	Actual No	207	761				
		Predicted Yes	Predicted No				
RNN	Actual Yes	3541	108	0.9704	0.9651	0.9678	0.9489
	Actual No	128	840				
		Predicted Yes	Predicted No				
Co-LSTM	Actual Yes	3442	207	0.9433	**0.9860**	**0.9681**	**0.9496**
	Actual No	49	919				

The experimental results for accuracy, precision, recall, and f-measure for the Airline review dataset are presented in Table 3. Like the movie review dataset, the Naive Bayes algorithm is more inclined towards positive sentiment. The precision value for the Naive Bayes algorithm is found to be 0.9992. Co-LSTM model has better accuracy, f-measure and recall value as compared to all other classifiers for the Airline review dataset. It can be observed that Co-LSTM and CNN seem to have very close performance results with RNN. The accuracy for Co-LSTM, RNN and CNN is found to be 94.96%, 94.89%, 93.44% respectively.

The performance results in the self-driving car dataset are presented in Table 4. It can be observed that Co-LSTM performs better in terms of accuracy, f-measure, and recall for self-driving car reviews. Precision value for the Naive Bayes algorithm is found to be 100%. The deep learning models such as RNN and CNN have accuracy 83.62% and 83.44% respectively. Unlike other datasets, the performance of SVM is satisfactory for self-driving car reviews in term of precision, recall and f-measure. Table 5 shows the performance result of all the models in US presidential election data. In this dataset, the accuracy of Co-LSTM is found to be 90.45%. It outperforms all other models in terms of accuracy, f-measure, and recall. Like the self-driving dataset, here the precision value for the Naive Bayes model is 1.0 due to more biasness towards positive sentiments.

Table 4: Confusion Matrix, Evaluation Parameters for Self Driving Car Dataset

Models		Confusion Matrix		Evaluation Parameters			
		Predicted Yes	Predicted No	Precision	Recall	F-Measure	Accuracy
SVM	Actual Yes	1615	398	0.9023	0.8549	0.8278	0.8081
	Actual No	274	491				
		Predicted Yes	Predicted No				
Naive Bayes	Actual Yes	2013	0	**1.0000**	0.7265	0.8416	0.7271
	Actual No	758	7				
		Predicted Yes	Predicted No				
Linear Regression	Actual Yes	1956	57	0.9717	0.7878	0.8701	0.7898
	Actual No	527	238				
		Predicted Yes	Predicted No				
Random Forest	Actual Yes	1907	106	0.9473	0.7583	0.8423	0.7430
	Actual No	608	157				
		Predicted Yes	Predicted No				
CNN	Actual Yes	1884	129	0.9359	0.8506	0.8912	0.8344
	Actual No	331	434				
		Predicted Yes	Predicted No				
RNN	Actual Yes	1916	97	0.9518	0.8426	0.8939	0.8362
	Actual No	358	407				
		Predicted Yes	Predicted No				
Co-LSTM	Actual Yes	1895	118	0.9414	**0.8798**	**0.9095**	**0.8643**
	Actual No	259	506				

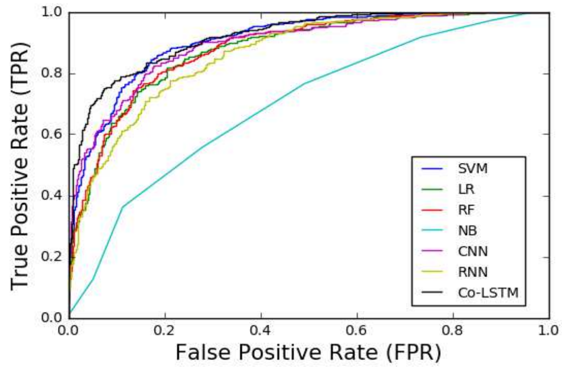
The observed value from the sentiment classification has been plotted through the Receiving Operator Characteristics (ROC) curve. This curve represents the trade-off between the false positive rate (FPR) and the true positive rate (TPR). The FPR is defined as the ratio between the number of false-positive to the total number of actual negative available in the dataset. Similarly, the TPR is defined as the ratio between the number of True positive value to the total number of actual positive in the dataset. It is same as the recall or sensitivity. The ROC curve is plotted against the false positive rate (x-axis) and the true positive rate (y-axis), which ranges from 0 to 1. It is one of the suitable approaches to find out the best model for the classification task. The classification model is said to have better performance if the curve is more inclined towards a true positive rate. The best prediction for a classifier will have curve towards (0,1), i.e., at the top-right region. So the performance of a model can be evaluated through the area under the curve of the ROC line. More is the area under curve, better is the performance of the model. Figure 10a, 10b, 10c and

Table 5: Confusion Matrix, Evaluation Parameters for GOP Datasets

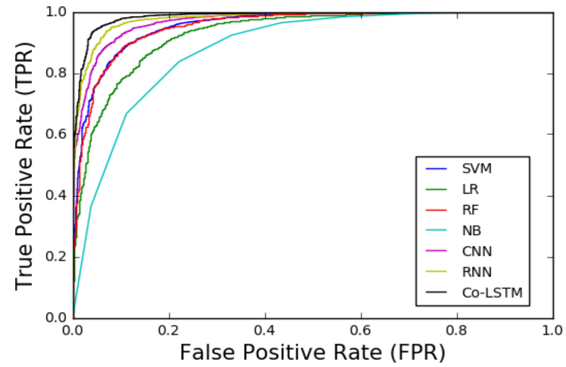
Models		Confusion Matrix		Evaluation Parameters			
		Predicted Yes	Predicted No	Precision	Recall	F-Measure	Accuracy
SVM	Actual Yes	2937	451	0.8669	0.9167	0.8911	0.8327
	Actual No	267	637				
		Predicted Yes	Predicted No				
Naive Bayes	Actual Yes	3388	0	**1.0000**	0.808	0.8938	0.8124
	Actual No	805	99				
		Predicted Yes	Predicted No				
Linear Regression	Actual Yes	3323	65	0.9808	0.8518	0.9118	0.8502
	Actual No	578	326				
		Predicted Yes	Predicted No				
Random Forest	Actual Yes	3241	147	0.9566	0.8529	0.9018	0.8355
	Actual No	559	345				
		Predicted Yes	Predicted No				
CNN	Actual Yes	3258	130	0.9616	0.9075	0.9338	0.8924
	Actual No	332	572				
		Predicted Yes	Predicted No				
RNN	Actual Yes	3333	55	0.9838	0.8686	0.9226	0.8698
	Actual No	504	400				
		Predicted Yes	Predicted No				
Co-LSTM	Actual Yes	3256	132	0.961	**0.9213**	**0.9408**	**0.9045**
	Actual No	278	626				

10d show the ROC curves of different classifiers for the movie, airline, self driving car and GOP datasets respectively. It can be observed that the black line, i.e., the ROC curve for the proposed model Co-LSTM is positioned more closed to TPR, which indicates that it has high TPR and low FPR. Naive Nayes is found to have more false-positive as compared to other classifiers in most of the datasets. For the Airline and Self-driving car dataset, Co-LSTM has a better distinguishable ROC curve as compared to other models. It can be observed that the ROC curve for deep learning models like RNN and CNN are more close to each other. The Area under the curve (AUC) for the models are listed in Table 6. It can be noted that AUC is more for Co-LSTM in all the datasets.

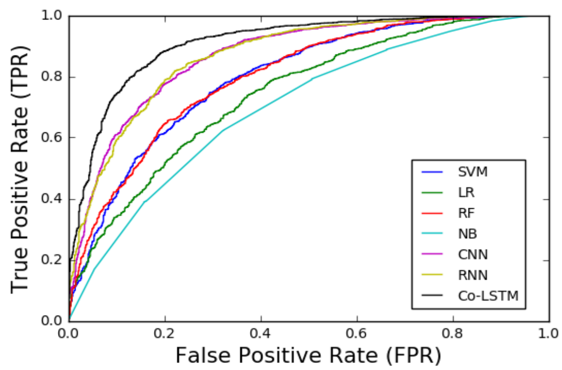
The paired t-test analysis has been performed for each pair of classification models for each of the evaluation parameters i.e., accuracy, precision, recall, and f-measure. It is used to check whether the performance of proposed model is significantly different from others or not. The t-test analysis has been performed for each data set for 5-fold cross-validation. The major parameters evaluated in t-test analysis is the p-value. The classifier is said to be significantly different than others if the p-value is less than 0.05. It can be observed from Table 7 that for accuracy, recall, and f-measure, the Co-LSTM is significantly different from all other classification models, i.e., the value obtained for accuracy, recall, and f-measure, is not due to randomness.



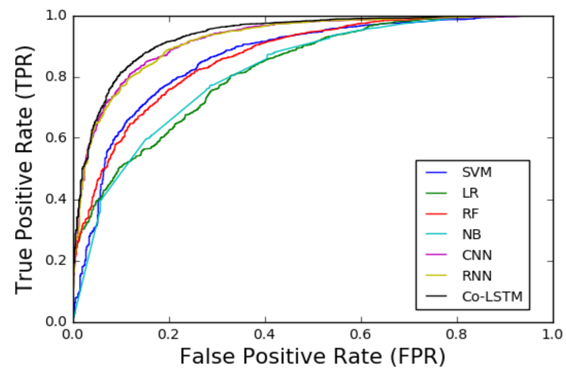
(a) Receiving Operator Characteristics (ROC) for Movie



(b) Receiving Operator Characteristics (ROC) for Airline



(c) Receiving Operator Characteristics (ROC) for Self Driving Car



(d) Receiving Operator Characteristics (ROC) for GOP

Figure 10: ROC Comparison for Different Classification Models

Table 6: Area under curve (AUC) value for ROC Curve

Models	Movie Review	Airline Dataset	Self Driving Car	GOP
SVM	0.905	0.955	0.795	0.862
Naive Bayes	0.877	0.930	0.743	0.820
Linear Regression	0.881	0.954	0.799	0.865
Random Forest	0.695	0.881	0.700	0.817
CNN	0.894	0.970	0.867	0.922
RNN	0.862	0.978	0.868	0.920
Co-LSTM	**0.920**	**0.984**	**0.909**	**0.934**

7. Conclusion and Future Work

A neural network architecture comprised of both CNN and LSTM has been proposed to predict the sentiment of customer reviews. The major advantage of this model is that it is not limited to a specific domain. Thus, the same model can be trained for product reviews as well as service reviews without

Table 7: t-test analysis (p-value) for various evaluation parameters

	Accuracy							**Precision**						
	SVM	NB	LR	RF	CNN	RNN	Co-LSTM	SVM	NB	LR	RF	CNN	RNN	Co-LSTM
SVM	-	0.058	0.792	0.257	0.162	0.486	0.010	-	0.037	0.166	0.324	0.208	0.375	0.017
NB	0.058	-	0.031	0.774	0.015	0.100	0.013	0.037	-	0.142	0.095	0.039	0.139	0.002
LR	0.792	0.031	-	0.151	0.017	0.370	0.019	0.166	0.142	-	0.043	0.058	0.155	0.039
RF	0.257	0.774	0.151	-	0.043	0.027	0.029	0.324	0.095	0.043	-	0.689	0.939	0.017
CNN	0.162	0.015	0.017	0.043	-	0.399	0.043	0.208	0.039	0.058	0.689	-	0.825	0.018
RNN	0.486	0.100	0.370	0.027	0.399	-	0.030	0.375	0.139	0.155	0.939	0.825	-	0.018
Co-LSTM	0.010	0.013	0.019	0.029	0.043	0.030	-	0.017	0.002	0.039	0.017	0.018	0.018	-
	Recall							**F-Measure**						
	SVM	NB	LR	RF	CNN	RNN	Co-LSTM	SVM	NB	LR	RF	CNN	RNN	Co-LSTM
SVM	-	0.001	0.012	0.048	0.179	0.202	0.016	-	0.368	0.602	0.409	0.223	0.652	0.012
NB	0.001	-	0.006	0.951	0.001	0.024	0.004	0.368	-	0.086	0.554	0.029	0.305	0.011
LR	0.012	0.006	-	0.246	0.008	0.230	0.021	0.602	0.086	-	0.188	0.006	0.745	0.005
RF	0.048	0.951	0.246	-	0.062	0.067	0.026	0.409	0.554	0.188	-	0.085	0.037	0.039
CNN	0.179	0.001	0.008	0.062	-	0.315	0.013	0.223	0.029	0.006	0.085	-	0.415	0.038
RNN	0.202	0.024	0.230	0.067	0.315	-	0.010	0.652	0.305	0.745	0.037	0.415	-	0.020
Co-LSTM	0.016	0.004	0.021	0.026	0.013	0.010	-	0.012	0.011	0.005	0.039	0.038	0.020	-

degrading the performance. No sophisticated manual feature engineering is required, thus avoiding domain-specific expertise. It is all due to the use of the pre-trained word-embedding model for embedding the input feature vector. In the next step, the use of CNN layer before the LSTM network helps to identify the important features only from the embedded vector, thus greatly improving the training time and hence makes it computationally feasible. At the last stage, the use of LSTM network layer helps to build the model by studying the sequential arrangements in the review rather than just considering words or phrases alone. Thus the model also incorporates the context study of the review and performs better in case of context such as negation as well as sarcasm.

In this study, an application of hybrid neural network architecture of both Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) built on the top of the word embedding model has been presented. The main advantage of this architecture is the sequential study of the important features in a review to predict the sentiment. Due to the application of the word embedding model and LSTM network, performance is quite better in multiple domains (as we experimented with movie reviews and airline tweets) without any domain-specific feature engineering. It can also be verified in other sentence classification activities.

8. Threat to Validation

The proposed architecture is based on the convolutional deep neural networks in the context of natural language processing. Few limitations of the proposed convolutional LSTM model may be as follows:

- The deep learning model requires a huge amount of data for proper training and is computationally intensive too.

- In feature matrix creation, the word embedding model is trained on the pre-trained data corpus. Pre-trained data corpus should be large enough to cover all frequently used words. If the pre-trained corpus is not sufficient; some of the important features might be missing while training the model.

- If the initial convolutional layer of the Co-LSTM model is unable to capture some of the texts order or sequence information, then the convolutional layer may fail to capture the sequential dependency of the words. Thus, the LSTM layer may act as just a fully connected layer without any memory.

- In this work, the word embedding model based on a pre-trained corpus has been considered. Sometimes, it is quite difficult to deal with misspellings or other irregularities found on the language used in social media. However, this can be improvised by building a social media-specific word-embeddings model.

Acknowledgement

This research work was supported by Fund for Improvement of S&T Infrastructure in Universities and Higher Educational Institutions (FIST) Scheme under Department of Science and Technology (DST), Govt. of India The authors wish to express their gratitude and heartiest thanks to the department of computer science & engineering, National Institute of Technology, Rourkela, India for providing their research support.

References

[1] E. Cambria, Affective computing and sentiment analysis, IEEE Intelligent Systems 31 (2) (2016) 102–107.

[2] A. Severyn, A. Moschitti, Twitter sentiment analysis with deep convolutional neural networks, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 959–962.

[3] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based lstm for aspect-level sentiment classification, in: Proceedings of the 2016 conference on empirical methods in natural language processing, 2016, pp. 606–615.

[4] W. Yin, K. Kann, M. Yu, H. Schütze, Comparative study of cnn and rnn for natural language processing, arXiv preprint arXiv:1702.01923.

[5] G. Vinodhini, R. Chandrasekaran, Sentiment analysis and opinion mining: a survey, International Journal 2 (6) (2012) 282–292.

[6] B. Liu, Sentiment analysis and opinion mining, Synthesis lectures on human language technologies 5 (1) (2012) 1–167.

[7] D. Maynard, M. A. Greenwood, Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis, in: LREC 2014 Proceedings, ELRA, 2014, pp. 26–31.

[8] P. Biyani, C. Caragea, P. Mitra, C. Zhou, J. Yen, G. E. Greer, K. Portier, Co-training over domain-independent and domain-dependent features for sentiment analysis of an online cancer support community, in: International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), IEEE, 2013, pp. 413–417.

[9] A. Bagheri, M. Saraee, F. De Jong, Care more about customers: Unsupervised domain-independent aspect detection for sentiment analysis of customer reviews, Knowledge-Based Systems 52 (2013) 201–213.

[10] C. Dos Santos, M. Gatti, Deep convolutional neural networks for sentiment analysis of short texts, in: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 69–78.

[11] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, B. Xu, Text classification improved by integrating bidirectional lstm with two-dimensional max pooling, arXiv preprint arXiv:1611.06639.

[12] Y. Ma, H. Peng, T. Khan, E. Cambria, A. Hussain, Sentic lstm: a hybrid network for targeted aspect-based sentiment analysis, Cognitive Computation 10 (4) (2018) 639–650.

[13] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm, in: Thirty-second AAAI conference on artificial intelligence, 2018, pp. 5876–5883.

[14] X. Wang, W. Jiang, Z. Luo, Combination of convolutional and recurrent neural network for sentiment analysis of short texts, in: Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers, 2016, pp. 2428–2437.

[15] G. Rao, W. Huang, Z. Feng, Q. Cong, Lstm with sentence representations for document-level sentiment classification, Neurocomputing 308 (2018) 49–57.

[16] A. Hussain, E. Cambria, Semi-supervised learning for big social data analysis, Neurocomputing 275 (2018) 1662–1673.

[17] E. Cambria, S. Poria, D. Hazarika, K. Kwok, Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 1795–1802.

[18] E. Cambria, Y. Li, F. Z. Xing, S. Poria, K. Kwok, Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 105–114.

[19] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, Sentiment analysis of twitter data, in: Proceedings of the workshop on languages in social media, Association for Computational Linguistics, 2011, pp. 30–38.

[20] Q. Ye, Z. Zhang, R. Law, Sentiment classification of online reviews to travel destinations by supervised machine learning approaches, Expert systems with applications 36 (3) (2009) 6527–6535.

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

[22] H. Saif, Y. He, H. Alani, Semantic sentiment analysis of twitter, in: International semantic web conference, Springer, 2012, pp. 508–524.

[23] R. K. Behera, S. K. Rath, S. Misra, R. Damaševičius, R. Maskeliūnas, Large scale community detection using a small world model, Applied Sciences 7 (11) (2017) 1173.

[24] A. Aue, M. Gamon, Customizing sentiment classifiers to new domains: A case study, in: Proceedings of recent advances in natural language processing (RANLP), Vol. 1, Citeseer, 2005, pp. 2–1.

[25] O. Araque, I. Corcuera-Platas, J. F. Sanchez-Rada, C. A. Iglesias, Enhancing deep learning sentiment analysis with ensemble techniques in social applications, Expert Systems with Applications 77 (2017) 236–246.

[26] S. Baccianella, A. Esuli, F. Sebastiani, Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining., in: LREC, Vol. 10, 2010, pp. 2200–2204.

[27] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of the conference on human language technology and empirical methods in natural language processing, Association for Computational Linguistics, 2005, pp. 347–354.

[28] J. W. Pennebaker, M. R. Mehl, K. G. Niederhoffer, Psychological aspects of natural language use: Our words, our selves, Annual review of psychology 54 (1) (2003) 547–577.

[29] E. Cambria, An introduction to concept-level sentiment analysis, in: Mexican International Conference on Artificial Intelligence, Springer, 2013, pp. 478–483.

[30] J. Yoon, H. Kim, Multi-channel lexicon integrated cnn-bilstm models for sentiment analysis, in: Proceedings of the 29th Conference on Computational Linguistics and Speech Processing (ROCLING 2017), 2017, pp. 244–253.

[31] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, Journal of Machine Learning Research 12 (Aug) (2011) 2493–2537.

[32] P. D. Turney, P. Pantel, From frequency to meaning: Vector space models of semantics, Journal of artificial intelligence research 37 (2010) 141–188.

[33] E. Cambria, C. Havasi, A. Hussain, Senticnet 2: A semantic and affective resource for opinion mining and sentiment analysis., in: FLAIRS conference, 2012, pp. 202–207.

[34] T. Wei, Y. Lu, H. Chang, Q. Zhou, X. Bao, A semantic approach for text clustering using wordnet and lexical chains, Expert Systems with Applications 42 (4) (2015) 2264–2275.

[35] K. Sailunaz, R. Alhajj, Emotion and sentiment analysis from twitter text, Journal of Computational Science 36 (2019) 101003.

[36] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, T. Robinson, One billion word benchmark for measuring progress in statistical language modeling, arXiv preprint arXiv:1312.3005.

[37] J. Ramos, et al., Using tf-idf to determine word relevance in document queries, in: Proceedings of the first instructional conference on machine learning, Vol. 242, Piscataway, NJ, 2003, pp. 133–142.

[38] O. Melamud, O. Levy, I. Dagan, A simple word embedding model for lexical substitution, in: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, 2015, pp. 1–7.

[39] O. Levy, Y. Goldberg, Neural word embedding as implicit matrix factorization, in: Advances in neural information processing systems, 2014, pp. 2177–2185.

[40] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, B. Qin, Learning sentiment-specific word embedding for twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2014, pp. 1555–1565.

[41] J. Wang, L.-C. Yu, K. R. Lai, X. Zhang, Dimensional sentiment analysis using a regional cnn-lstm model, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016, pp. 225–230.

[42] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[43] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882.

[44] C.-N. Chou, C.-K. Shie, F.-C. Chang, J. Chang, E. Y. Chang, Representation learning on large and small data, Big Data Anal. Large-Scale Multimed. Search. Wiley, Hoboken, NJ (2019) 3–30.

[45] Y. Wan, Q. Gao, An ensemble sentiment classification system of twitter data for airline services analysis, in: Data Mining Workshop (ICDMW), 2015 IEEE International Conference on, IEEE, 2015, pp. 1318–1325.

[46] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, A. L. Yuille, 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, pp. 4545–4554.