Predicting Sentiment Polarity of Microblogs using an LSTM – CNN Deep Learning Model

Mayank Kumar Nagda, Sankalp Sinha, Poovammal E

***Abstract*: In this paper we propose a novel supervised machine learning model to predict the polarity of sentiments expressed in microblogs. The proposed model has a stacked neural network structure consisting of Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) layers.**

In order to capture the long-term dependencies of sentiments in the text ordering of a microblog, the proposed model employs an LSTM layer. The encodings produced by the LSTM layer are then fed to a CNN layer, which generates localized patterns of higher accuracy. These patterns are capable of capturing both local and global long-term dependences in the text of the microblogs.

It was observed that the proposed model performs better and gives improved prediction accuracy when compared to semantic, machine learning and deep neural network approaches such as SVM, CNN, LSTM, CNN-LSTM, etc.

This paper utilizes the benchmark Stanford Large Movie Review dataset to show the significance of the new approach. The prediction accuracy of the proposed approach is comparable to other state-of-art approaches.

Keywords: Deep Learning, Convolutional Neural Networks, LSTM, Natural Language Processing, Sentiment Analysis.

I. INTRODUCTION

The area of computer science concerned with the ability of computer programs to understand, process, interpret and generate human languages is referred to as Natural Language Processing (NLP) [18]. NLP is used to interpret free text or speech to make it understandable and analyzable by a computer. 'Understanding languages' is not a simple task, as it involves both identifying the inherent structure of the text and extracting the sentiments, emotions and the overall contextual meaning expressed in it [18]. This task is further complicated as humans often use different styles of expressing emotions, feelings, thoughts, and opinions with a large and diverse vocabulary.

NLP deals with languages in both spoken and written domains. It has two major areas of focus namely *language processing* and *language generation*. Language processing deals with the analysis of human language to extract meaningful features such as opinions, emotions, sentiments, context, etc. While language generation aims at generating natural languages with meaningful features and context by artificial systems. This paper focuses on language processing in the written domain, especially on the classification of microblogs based on the sentiment expressed in them.

The internet, especially since the turn of the 21st century, has brought the world closer in a way as nothing has ever before [5]. In the last decade, the internet has seen a meteoric rise in its user base worldwide. With 3.2 Billion users in 2015 and counting, it has undoubtedly become a leading tool for expressing and sharing ideas, thoughts, and opinions. More recently social media and other content-based websites like Facebook, Twitter, YouTube, Amazon, and others have risen to become one of the most frequently visited websites on the internet [5]. These websites generate large amounts of data in the form of articles, posts, tweets, comments, reviews, and stories. This type of data is commonly referred to as microblogs [25]. Microblogs are a combination of both objective (facts) and subjective (views) text, the size of which is smaller than traditional blogs and usually does not exceed 1000 words. Microblogs have frequent use of colloquial words with informal and fragmented language [25]. This along with the limited number of words in a microblog poses a unique challenge in sentiment extraction and classification.

Sentiment analysis is a process that leverages NLP techniques to extract views, opinions, and emotions as features form some form of speech or text source [16]. These features are then used to summarize or categorize the data. Before any analysis can be done on the extracted sentiments, we must have a way to represent these sentiments. For sentiment representation, there are two popular approaches namely, *categorical* and the *dimensional* [25]. In the categorical approach emotions or opinions are represented as discrete classes. These classes could be binary like, 'positive' or 'negative' or contain multiple categories such as Ekman's basic emotions 'anger', 'disgust', 'fear', 'happiness', 'sadness' and 'surprise'[8]. While in the dimensional approach each emotion or opinion is expressed as a state that has numeric values in multiple dimensions. One such example is the Valence – Arousal (VA) space, in which the degree of positive and negative sentiment is measured by the dimensions of valence vector, while the degree of calm and excitement is measured by the dimensions of the arousal vector [25]. Table 1 shows the high-level visualization of sentiment analysis for the sentence 'The movie had a strong storyline and amazing special effects".

Table- I: High-Level Visualization of Sentiment Analysis

Subject	the movie
Features	the storyline and special effects
Opinions	strong and amazing
Polarity	positive

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Mayank Kumar Nagda*, Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, India.
Sankalp Sinha, Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, India.
Poovammal E, Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, India.

In today's world which is heavily influenced by social media and online content, such classification of microblogs is of great interest to businesses and companies around the world [5]. The analysis of microblogs allows businesses to make better advertising and marketing decisions, understand trends of the marketplace and judge the general consumer perception of their goods and services. Sentiment analysis also finds great use in the fields of user review analysis and public opinion analysis [25]. The remainder of this paper is structured into five sections. The discussion on the related work carried out on sentiment analysis of microblogs is presented in Section 2. A brief background of our approach and the proposed model is discussed in Section 3, while Section 4 outlines the structure of our model and describes its components in detail. Section 5 discusses the experimental setup and results followed by the concluding remarks in Section 6.

II. RELATED WORK

This In the last decade the area of opinion-oriented information retrieval and sentiment analysis has seen a lot of work by various researchers. Much of the early work is systematically summarized in [21]. Their survey extensively covers some of the early techniques and approaches for sentiment extraction and opinion mining.The sentiments expressed in a text can be analyzed from four different perspectives or levels namely *document level*, *sentence level*, *phrase level* and *word level* [21]. Our paper focuses on the approaches and techniques applied to sentiment extraction and analysis. Therefore, we focus on the evolution of techniques and approaches for sentiment analysis of text in microblogs. We categorize the techniques and approaches of sentiment analysis based on its nature and its evolution in chronological order, there are:

a) POS, Lexicon, and Dictionary-based Approaches.
b) Statistical and Machine Learning Approaches.
c) Deep Neural Network Approaches.

The traditional approach to sentiment analysis involves Part of Speech (POS) tagging along with the use of dictionaries and semantic rules. With the advent of machine learning, extensive research was carried out using machine learning models. It was observed that the machine learning models outperformed the traditional POS tagging and lexicon approaches. like [9] [20]. In recent years deep-neural network models in combination with traditional approaches have seen greater use in sentiment analysis. Deep neural network models have outperformed previous approaches. In subsection 2.1, 2.2 and 2.3, we briefly discuss the previous research work on sentiment analysis of microblogs using part of speech and lexicon approaches, machine learning approaches and deep neural network approaches.

A. POS, Lexicon, and Dictionary-based Approaches

In the age of social media, microblogs and other content-based websites have become very popular [5]. To capture and make sense of the thoughts, and opinions expressed in these microblogs, classifying these microblogs based on the polarity of the sentiments expressed in them has become increasingly important. Lexicon based methods approach the problem of classifying the sentiments expressed in the text by calculating the overall semantic orientation of the words and phrases present in it [5]. This semantic orientation is manually or automatically curated in dictionaries which store a word or phrase and its corresponding sentiment orientation [5].

A sentence-level, dictionary-based sentiment classification approach for the same was presented in [9] called *SentiWordNet*. Their approach used a collection of eight ternary classifiers having different behavior on each *synset* of the *WordNet*. The rsesults of these classifiers were used to calculate a score. The magnitude of this score would determine how positive, negative or objective the words in each *synset* were. In [20] the authors used POS tagging along with N-grams as features to propose a multinomial Naïve Bayes model for sentiment classification of tweets. Further showing that bigrams are a better feature when compared to unigrams and trigrams and inclusion of negation words improves the accuracy of their model.

In [1] the authors experimented with a feature model, a unigram model, and a tree-based model. Showing that combining prior polarity of words with their POS tags produces features that have a high weight when it comes to the classification of the tweet as 'positive', 'negative' or 'neutral'. Many variants of the lexicon and semantic approaches utilizing POS tagging and dictionary seem to be naïve in nature but have been shown to perform better than chance.

B. Statistical and Machine Learning Approaches

There has been a lot of research on using various machine learning techniques to extract sentiment polarity of tweets. In [22] the authors showed that a naive Bayesian classifier outperformed the Maximum Entropy model. In [10] the authors used POS tagging, unigrams and bigrams as their feature vectors to compare SVM's with naïve Bayesian and Maximum Entropy models and demonstrated that SVM's outperform other models. In [4] the authors showed that learning rate of stochastic gradient descent-based models, when adjusted to account for different feature vectors, perform better than multimodal naive Bayesian models at the binary classification of sentiments expressed in tweets.

In [2] the authors proposed a two-phase approach for classifying the sentiments expressed in tweets. They employed prior polarity of words and POS tagging along with hashtags and punctuation marks as their feature vectors. Instead of using clean labeled data they used noisy labels extracted from twitter to train their supervised machine learning model. In the first phase, the tweets are classified on the basis of their 'subjectivity' or 'objectivity'. In the second phase all the tweets that are marked as subjective, are then further classified as 'positive' or 'negative'.

It is quite evident that supervised machine learning models need large amounts of labeled data for training, acquiring which can be time and labor-intensive task. To overcome this issue in [15] the authors proposed a semi-supervised machine learning approach in which they leverage prior known lexical and semantic knowledge, along with the labeled data to train supervised machine learning models. Their approach proved to be effective when the number of labeled training samples is small.

In [19] the authors proposed a three-step process to extract a set of feature vector from tweets. The set of features vectors consists of eight features that include POS tags, special keywords, negation, emoticons, number of positive and negative hashtags and frequency positive and negative keywords (Neethu and Rajasree, 2013). They then applied Naïve Bayes, SVM, Maximum, Entropy and Ensemble classifiers to test the classification accuracy of their feature vector and illustrated that their approach outperforms traditional lexicon and dictionary-based approaches.

C. Deep Neural Network Approaches

Neural networks are also an area of continued interest for sentiment extraction from microblogs like tweets, online reviews and web posts. In [23] the authors showed that an RNTN (Recursive Neural Tensor Network) approach outperforms all previous methods and reaches an accuracy of 80.7% on fine-grained sentiment prediction. While also successfully capturing negation of varying scopes on different sentiments [23]. In [7] the authors proposed a deep convolutional neural network model that carries out a sentiment analysis on the sentence-level, word-level, and character-level. Validating their model on the Stanford Twitter Sentiment (STS) and Stanford Sentiment Treebank (SSTb) corpora. With random word embeddings, their CharSCNN model reaches an accuracy of 85.7% in binary classification. Proving that deep CNN can reach the accuracy level of Recursive Neural Tensor Networks when classifying sentiments in microblogs.

In [26] the authors proposed an LSTM recurrent network for twitter sentiment prediction and showed that their model is effective at learning sentence-level representations, with a flexible compositional structure reaching an accuracy of 84% on predicting the polarity of the tweets in the STS corpora.

In [25] the authors applied a CNN-LSTM model for dimensional sentiment analysis of the text. Their model performs fine-grain sentiment analysis of microblogs by predicting the VA rating of sections of text they called 'regions'. Their model outperforms lexicon and regression models proposed in previous studies.

III. BACKGROUND

In this paper, we present a supervised machine learning model that uses the binary categorical approach to represent and categorize sentiments in microblogs. The model uses an LSTM and a CNN layer in conjunction, to categorize movies reviews as 'positive' or 'negative' based on the sentiments expressed in them.

To better understand why our proposed model uses an LSTM and CNN layer in conjunction, we first briefly discuss the long-term dependency problem in Recurrent Neural Networks in subsection 3(A) and Convolutional Neural Networks in subsection 3(B).

A. The Long-Term Dependency Problem in Recurrent Neural Networks (RNNs)

RNNs are designed to be context preserving, this means that RNNs can remember relevant information about the data they have seen in the past and use it to make relevant prediction or classification [3]. RNNs work best when the distance between relevant information and the point of its application to make the prediction or classification is small [3]. Assume that an RNN model wants to predict the underlined word in the sentence 'I am a citizen of India'. The distance between the relevant information (i.e. 'citizen') and the predicted word 'India' is small and hence an RNN model has a higher chance of predicting the word 'India' correctly.

RNNs in practice fail to make accurate predictions as this distance increases. In essence, they are unable to reliably maintain long-term dependencies between the data they have seen as the temporal space increases [3] [13]. Therefore, when asked to predict the underlined word in the sentence 'My hometown is in Tamil Nadu, which is a state in India and I speak Tamil'. RNNs are less likely to remember the association of the language 'Tamil' to the state 'Tamil Nadu' as the temporal space between the two words is fairly large.

LSTM is a variant of RNN and was designed to able to remember long-term dependencies of words and their context in sentences [12]. They are an elegant solution to the long-term dependency problem in RNNs. Microblogs often have a high temporal separation between the relevant information and its point of use due to the fragmented and informal style of language used [11]. This is why LSTMs perform better than RNNs at the classification of microblogs based on the sentiments expressed in them.

B. Convolution Neural Network

Convolutional Neural Networks were inspired by the human visual cortex and designed to perceive and extract features from images [6][14]. CNN's require little pre-processing of the image and no predesigned complex feature detectors. CNN's can learn these complex feature detectors with the use of basic feature detectors like edge, corner, blob, etc. The importance of a complex feature is determined by a set of learnable weights [6].

CNN in its essence is a deep neural network that can be broken down into a series of convolution and pooling layers [14]. The convolution layers extract the relevant convoluted features while the pooling layer reduces the size of the convoluted feature by performing downsampling. The downsampling ensures the extraction of the dominant features of the data and reduces the computational power needed to train the network. The convolution and pooling layers produce high-level features of the data passed through it [14]. Each data point is a non-linear combination of these high-level features. Then a fully connected neural network can be used to learn the non-linear dependencies among these high-level features for classification [6].

IV. PROPOSED MODEL

The architecture of the proposed LSTM – CNN model is shown in Figure 1. The proposed model consists of the following layers, input layer, embedding layer, LSTM layer, CNN layer, and a fully connected layer. The input layer passes individual reviews as a tokenized string of integers, to the embedding layer. The embedding layer is initialed with uniform weights which are learned by the model during the training phase. Each word embedding t produced by the embedding layer is passed through an LSTM layer. We can define a unit of LSTM at each time step t as a collection of vectors in $\mathbb{R}^d$ consisting of an input gate i_t, forget gate f_t, output gate o_t, memory cell c_t and a hidden state h_t, where d is the magnitude of the memory dimension [12] [24]. The LSTM equations are given from (1) through (6).

$$i_t = \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\right) \quad (1)$$

$$f_t = \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\right) \quad (2)$$

$$o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\right) \quad (3)$$

$$u_t = \tanh\left(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}\right) \quad (4)$$

$$c_t = i_t \odot u_t + f_t \odot c_{t-1} \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

In the equations, W, U, and b denote the two weight matrices and the bias vector for the input gate, output gate, forget gate, tanh layer, memory cell, and the hidden layer. While $\boldsymbol{\sigma}$ denotes the logistic sigmoid function and $\odot$ denotes element-wise multiplication.

In an LSTM unit, the input gate is fed a new stream of data at each time step t and is responsible for making the decision on remembering the information it processes. While the forget gate is responsible for regulating the amount of information that should be removed from the memory cell.

We also apply a dropout function before each LSTM layer in order to curb overfitting. Finally, the encoding matrixes $S_t \in \mathbb{R}^{d \times n}$ produced at each time step t, as output by the last LSTM layer is passed through a CNN layer. The CNN layer performs convolution on each encoding matrix S_t by applying a linear filter X, which is a weight matrix having a length d and region size l. The application of the linear filter on all possible sub-matrices of S_t, produces a feature map vector M_t at each time step t [14].

$M_t = [m_1, m_2, m_3, \dots, m_i]$, where $m_i = X \cdot S_{i:i+l}$ and $i \in (1, n-l)$. (7)

On each feature map vector produced by the convolution layer, a one-dimensional global max pooling layer is applied, to generate the set of most important features . This is done by taking the maximum value from each feature map vector.

$F = [f_1, f_2, f_3 \dots f_i]$, where $f_i = \max(M_t)$ at each time step t. (8)

This feature set F is passed through a fully connected neural layer that uses a sigmoid activation function, to calculate the probability distribution over the two sentiment labels 'positive' or negative.

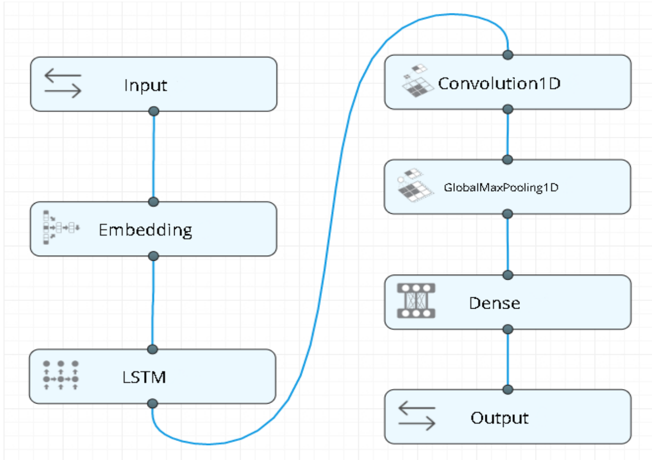

Fig. 1.The Architecture of the proposed LSTM – CNN model

V. DATASET AND EXPERIMENTAL SETUP

This section is divided into two subsections. The details of the dataset used and the preprocessing applied to the data is briefly summarized in subsection 5(A). While the details of the experimental setup are discussed in subsection 5(B).

A. The Dataset and Data Preprocessing

We utilize the popular Stanford Large Movie Review dataset [17]. The dataset is a collection of 50,000 movie reviews extracted from IMBD and labeled with the polarity of the reviews as positive or negative using a score out of 10. A review with a score ≥ 7 is marked as positive while a review with a score ≤ 4 is marked as negative. The dataset has even number of positive and negative reviews. This ensures that random guessing only yields a 50% prediction accuracy. Further, the dataset has no more than 30 reviews per movie. Using the Stanford Large Movie Review Dataset, we create our dataset by randomly extracting 1000 reviews.

Our dataset is further processed by removing all references to screen names, extra spaces, tabs, newlines, URLs, and punctuations from the reviews. This is done to reduce the size of each review by removing words and characters and text that do not contribute to the context or polarity of the tweet. The dataset is then shuffled to avoid any element of bias or pattern. Further, the words of each review are hashed and vectorized as a string of semicolon-separated numbers. The final step in data preprocessing involves splitting the dataset in 90:10 ratio to form the training set of 900 reviews and validation set of 100 reviews.

B. Experimental Setup

To evaluate our proposed model, we compare it against CNN, LSTM, and CNN-LSTM. For a fair and balanced evaluation, the parameters of the models are kept identical. The Convolution layer in CNN, CNN-LSTM and our model has the same hyper-parameters. It uses a 'ReLu' activation function with a linear filter of length 3 and has 64 output dimensions. All the models are trained for 10 epochs with a batch size of 32 and use the binary cross-entropy loss function. All the experimental parameters are summarized in Table 2.

Table- II: Experimental setup and parameters used

Parameters	CNN	LSTM	CNN-LSTM	Our Model
Training set	900 reviews	900 reviews	900 reviews	900 reviews
Validation set	100 reviews	100 reviews	100 reviews	100 reviews
Epoch	10	10	10	10
Batch Size	32	32	32	32
Learning Rate	0.001	0.001	0.001	0.001
Loss Function	Binary Crossentropy	Binary Crossentropy	Binary Crossentropy	Binary Crossentropy
Convolution layer	Activation: ReLu Filter Length: 3 Output Dim: 64	——	Activation: ReLu Filter Length: 3 Output Dim: 64	Activation: ReLu Filter Length: 3 Output Dim: 64

VI. RESULTS

We use average training accuracy and validation accuracy, as the parameters to compare the different models. The comparisons of the training accuracy of our model against other neural networks models we trained on our dataset are shown in Figure 2. As can be seen in Figure 2, our model achieves similar training accuracy to both LSTM and CNN-LSTM, while it performs much better in training than CNN.

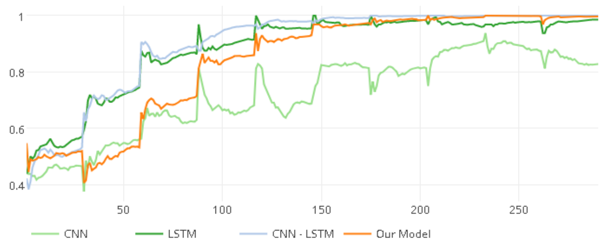

Fig. 2.Training Accuracy vs Sample of different models on our dataset

While Figure 3 shows a comparison of the validation accuracy of each model. It is evident from the comparison in Figure 3 that our model achieves comparatively higher validation accuracy overall epochs than the other models. Our model achieves a peek validation accuracy of 84.37% on our dataset. It consistently performs better than the popular CNN-LSTM model. Emboldened by the results we applied our model on the complete Stanford Large Movie Review dataset [17] and achieved a final validation accuracy of 87.92%. as shown in Figure 4.

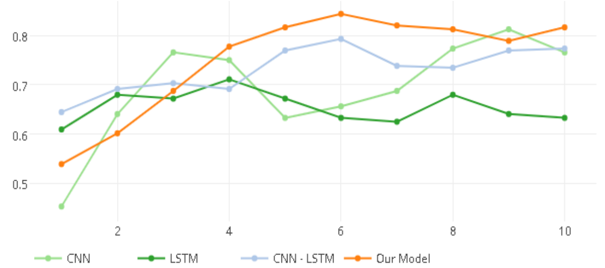

Fig. 3.Validation Accuracy vs Epoch of different models on our dataset

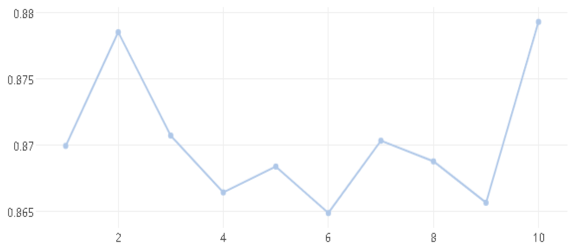

Fig. 4.Validation Accuracy vs Epochs of our model on the complete Stanford Large Movie Review Dataset (50,000 tweets)

The comparison of our model with previous approaches for sentiment classification of movie reviews is shown in Table 3. Our model outperforms other Maximum Entropy, Naïve Bayes, SVM and Markov Blanket classifier models utilizing high-level features like unigrams, bigrams, n-grams or minimum cut based subjectivity summarizations. Our model produces state-of-the-art results and outperforms lexicon, machine learning and simple neural network approaches with the use of pre-trained high-level feature vectors.

Table- II: Comparison of previous approaches for document-level sentiment classification of movie reviews extracted from IMDb.

Studies	Techniques Applied	Features	Accuracy
Pang et al. 2002	Maximum Entropy, Naïve Bayes and Support Vector Machine	Unigrams, Bigrams, frequency, position and feature presence	82.9%
Pang et al. 2004	Naïve Bayes and Support Vector Machine	Sentence level subjectivity summarizations based on minimum cuts.	86.4%
Bai et al. 2005	Two-stage Markov Blanket Classifier	Dependency among words	87.5%
Kennedy and Inkpen, 2005	Support Vector Machine and term counting	——	86.2%
A.L Mass et al. 2011	Hybrid supervised-unsupervised model	Without Additional Unlabeled data and Bag of Words	87.44%
This Work	**LSTM – CNN**	**Sentence-Level Summarization**	**87.92%**

VII. CONCLUSION

In this paper, we present a deep neural network model for document-level sentiment classification of movie reviews in the Stanford Large Movie Review Dataset [17]. Our model utilizes an LSTM layer to effectively capture the set of local and global contextual dependencies in a review. The CNN layer then extracts a set of features, using which the model learns to classify a review as 'positive' or 'negative' based on the sentiments expressed in it.

To the best of our knowledge, the previous state of the art approach on the Stanford Large Movie Review dataset is presented in [17] where their model without additional unlabeled data reached an accuracy of 87.44%. Our model produces state-of-the-art results on the same dataset achieving a validation accuracy of 87.92% without additional unlabeled data. It outperforms previous lexicon, machine learning and neural network approaches like Naïve Bayes, Maximum Entropy, Support Vector Machines, Markov Blanket classifiers, CNN, LSTM and CNN-LSTM that use pre-trained feature vectors. Further, the architecture of our model is simpler as it has fewer layer than other more complex multi-layer models. There is scope for improving the accuracy of our model by the application of pre-trained features and additional unlabeled data as also done in [17]. Further, the model can also be applied in other domains of application like customer reviews, tweets, etc.

REFERENCES

1. Agarwal, A., Xie, B., Vovsha, I., Rambow, O. and Passonneau, R., 2011, June. Sentiment analysis of twitter data. In Proceedings of the Workshop on Language in Social Media (LSM 2011) (pp. 30-38).
2. Barbosa, L. and Feng, J., 2010, August. Robust sentiment detection on twitter from biased and noisy data. In Proceedings of the 23rd international conference on computational linguistics: posters (pp. 36-44). Association for Computational Linguistics.
3. Bengio, Y., Simard, P. and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2), pp.157-166.
4. Bifet, A. and Frank, E., 2010, October. Sentiment knowledge discovery in twitter streaming data. In International conference on discovery science (pp. 1-15). Springer, Berlin, Heidelberg.
5. Chen, T., Xu, R., He, Y. and Wang, X., 2017. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. Expert Systems with Applications, 72, pp.221-230.
6. Cliche, M., 2017. Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. arXiv preprint arXiv:1704.06125.
7. Dos Santos, C. and Gatti, M., 2014, August. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (pp. 69-78).
8. Ekman, P., 1992. An argument for basic emotions. Cognition & emotion, 6(3-4), pp.169-200.
9. Esuli, A. and Sebastiani, F., 2006, May. Sentiwordnet: A publicly available lexical resource for opinion mining. In LREC (Vol. 6, pp. 417-422).
10. Go, A., Bhayani, R. and Huang, L., 2009. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1(12), p.2009.
11. Graves, A. and Schmidhuber, J., 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks, 18(5-6), pp.602-610.
12. Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.
13. Hochreiter, S., Bengio, Y., Frasconi, P. and Schmidhuber, J., 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.
14. Kim, Y., 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.
15. Li, S., Wang, Z., Zhou, G. and Lee, S.Y.M., 2011, June. Semi-supervised learning for imbalanced sentiment classification. In Twenty-Second International Joint Conference on Artificial Intelligence.
16. Liu, B., 2015. Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press.
17. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y. and Potts, C., 2011, June. Learning word vectors for sentiment analysis. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1 (pp. 142-150). Association for Computational Linguistics.
18. Nasukawa, Tetsuya, and Jeonghee Yi. "Sentiment analysis: Capturing favorability using natural language processing." Proceedings of the 2nd international conference on Knowledge capture. ACM, 2003.
19. Neethu, M.S. and Rajasree, R., 2013, July. Sentiment analysis in twitter using machine learning techniques. In 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.
20. Pak, A. and Paroubek, P., 2010, May. Twitter as a corpus for sentiment analysis and opinion mining. In LREc (Vol. 10, No. 2010, pp. 1320-1326).
21. Pang, B. and Lee, L., 2008. Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), pp.1-135.
22. Parikh, R. and Movassate, M., 2009. Sentiment analysis of user-generated twitter updates using various classification techniques. CS224N Final Report, 118.
23. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A. and Potts, C., 2013, October. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 1631-1642).
24. Tai, K.S., Socher, R. and Manning, C.D., 2015. Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075.
25. Wang, J., Yu, L.C., Lai, K.R. and Zhang, X., 2016, August. 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) (pp. 225-230).
26. Wang, X., Liu, Y., Chengjie, S.U.N., Wang, B. and Wang, X., 2015. Predicting polarities of tweets by composing word embeddings with long short-term memory. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (Vol. 1, pp. 1343-1353).

AUTHORS PROFILE

Mayank Kumar Nagda is an alumnus of SRM Institute of Science and Technology where he earned a bachelor's degree in the field of Computer Science and Engineering. Artificial Intelligence, Natural Language Processing, and Data Analytics are some of his areas of interests and expertise. Mayank is also fascinated by the idea of introducing automation in the regular lives of human beings so that it can assist and can create a new firm base for the next human evolution.

Sankalp Sinha is an alumnus of SRM Institute of Science and Technology where he earned his bachelor's degree in Computer Science and Engineering. Application of deep belief networks for sentiment analysis, text mining and natural language processing are his areas of interest. He is a full stack developer with hands on experience in building scalable high-performance systems and loves to automate the mundane tasks around him in his free time.

Dr. E. Poovammal is a Professor in the Department of Computer Science and Engineering at SRM Institute of Science and Technology. She joined in SRM in the year 1996. Before joining SRM, she worked in industry for five years. She obtained her B.E. Degree in Electrical and Electronics Engineering from Madurai Kamaraj University in the year 1990, M.E degree in Computer Science and Engineering from Madras University in the year 2002 and Ph.D. degree in Computer Science and Engineering from SRM University in 2011. Her research interests include data Big Data Analytics and machine learning.