



Sentiment Analysis of Bengali Tweets Using Deep Learning

Kamal Sarkar

► To cite this version:

Kamal Sarkar. Sentiment Analysis of Bengali Tweets Using Deep Learning. 3rd International Conference on Computational Intelligence in Data Science (ICCIDS), Feb 2020, Chennai, India. pp.71-84, 10.1007/978-3-030-63467-4_6 . hal-03434785

HAL Id: hal-03434785

<https://inria.hal.science/hal-03434785>

Submitted on 18 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Sentiment Analysis of Bengali Tweets Using Deep Learning

Kamal Sarkar

Department of Computer Science and Engineering
Jadavpur University, Kolkata, India

jukamal2001@yahoo.com

<https://orcid.org/0000-0002-0689-3976>

Abstract. Sentiment analysis is the research area that deals with analysis of sentiments expressed in the social media texts written by the internet users. Sentiments of the users are expressed in various forms such as feelings, emotions and opinions. Tweet sentiment polarity detection is an important sentiment analysis task which is to classify an input tweet as one of three classes: positive, negative and neutral. In this study, we compare various deep learning methods that use LSTM, BILSTM and CNN for the sentiment polarity classification of Bengali tweets. We also present in this paper a comparative study on the Bengali tweet sentiment polarity classification performances of the traditional machine learning methods and the deep learning methods.

Keywords: Sentiment polarity classification, Machine learning, Deep learning, Bengali Tweets, LSTM, BILSTM, CNN, Naïve Bayes, Support Vector Machines

1 Introduction

There has been a tremendous surge of data on the internet after the social media boom. Since the advent of the Social Web, the amount of textual content on the Internet has exponentially grown and provides a great deal of potential for analysis. Since millions of tweets are generated every day, machine learning can be more useful for efficient and fast data analysis to better understand the activities of the social media users as well as the users' feedback on various social and political issues. The online merchants allows the customers to give feedback on various products. The opinions and comments on the other issues related to crickets, sports, movies, music etc. are also available on the Internet.

After collecting a vast amount of users' feedback or opinions/comments, the next task is to analyze them to derive knowledge that can help the online merchants in improving the customer services and launching new products to satisfy customer needs. The opinions and comments related to various political and social events can be useful in policy making as well as enriching the various

academic fields such as politics, political science, sociology, psychology. Sentiment of a text refers to the expression of people's thought, judgment or attitude prompted by feelings [1] in the text. Sentiment polarity classification is one of the important sentiment analysis tasks. In sentiment polarity classification task, sentiment expressed in an input tweet is classified into one of three categories -positive, negative or neutral.

Since manual processing of the vast amount of social media texts for knowledge extraction is a difficult task, there is a need for an automatic system which can determine the polarity of the sentiments of social media texts coming from multiple heterogeneous sources.

Since the Bengali language is the 7th most spoken language in the world, with approximately 215 million speakers all over the world. Bengali is a primary language in Bangladesh, and in India - West Bengal, Tripura and Assam and recently Bengali tweets are becoming available on the various social media, there is a need for a tool for sentiment classification of Bengali tweets. But the most existing sentiment classification systems have been developed for processing English texts and a limited numbers of research works on Bengali sentiment classification are reported in the literature.

In this paper, we focus on developing sentiment classification system for Bengali tweets and report the results obtained for Bengali sentiment classification using the several deep learning algorithms such as LSTM (Long Term and Short Term Memory), BiLSTM (Bidirectional Long Term and Short Term Memory) and CNN (Convolutional Neural Networks).

In the rest of the paper, section 2 discusses the existing works related to our work. In section 3, we present descriptions of the model. In section 4, we discuss the experimental setup, evaluation and results. We conclude in section 5.

2 Related Work

The earliest works on sentiment polarity classification used lexicon based sentiment polarity detection methods [2] that used information retrieved from sentiment lexicon to determine sentiment polarity (positive class or negative class) of the text. The simpler version of sentiment lexicon is a mere collection of positive and negative terms [3]. Since manual construction of such lexicon is a laborious task, automatic construction of sentiment lexicon has also been proposed [4] [5]. Some researchers have studied on automatic prediction of semantic orientation of adjectives and adjectival phrases [6] [7]. The approach proposed in [8] determines polarity of words. SentiWordNet [9] is a more informative sentiment lexicon which is created by analyzing the glosses (associated with synsets) retrieved from WordNet [10]. The main criticism of the lexicon based sentiment analysis is that maintaining a universal sentiment lexicon is difficult.

Compared to sentiment lexicon based approach, machine learning based approach has been more popular due to the fact that the machine learning based approach does not require in-depth domain knowledge for crafting rules from data

and it automatically learns the relationship between input and the output from the data provided to the system. So, the most existing works on sentiment polarity detection considers this task as a classification problem and use a number of machine learning algorithms for coarse grained (document level) sentiment polarity classification [11-18]. The more fine-grained analysis has also been done using the linguistic features combined with machine learning methods [19-22]. Though sentence level sentiment polarity classification [23] is not uncommon today, the substantial portion of previous research works focused on sentiment classification of product and movie reviews [24] [25] [26] [27].

In supervised learning method for sentiment polarity detection, a labeled corpus converted to a collection of labeled feature vectors is provided to the learning algorithms. This labeled data is used to train a machine learning (ML) algorithm and then the trained model is applied to classify new unseen texts. Since ML algorithm requires each instance of text to be transformed into feature vector, the common features used for feature representation are individual words, n-grams, surrounding words, punctuations and the popular supervised learning algorithms used for this task are SVM (Support Vector Machines), K-NN (K-nearest neighbor), decision tree and Naïve Bayes [28] [29] [30] [31] [32] and ANN (Artificial Neural Networks) [33].

Some approaches to sentiment polarity classification are the hybrid approaches [34] that combine both supervised and unsupervised learning for analyzing reviews. Sentiment analysis of micro-blog posts and tweets have been presented in [35] [36].

Within the last several years, deep learning based approaches to sentiment analysis have been applied and many research papers have reported efficacy of deep learning based approaches applied to sentiment polarity detection tasks [37] [38] [39].

Most of the existing works on sentiment polarity detection discussed above has dealt with detecting polarity of non-Indian language texts. Since many Indians are now actively participating in writing their opinions and comments on Twitter or other micro-blog sites using Indian languages, opinionated texts in Indian languages are now available on the Internet. But there is a limited number of research works on sentiment analysis of tweets or comments or opinions written in Indian languages. So, there is a need for a system that can classify a large volume of Indian social media texts.

For sentiment analysis of tweets in Indian languages, a few attempts have been made so far. A contest on sentiment analysis in Indian languages (SAIL) tweets @ MIKE 2015 was held in 2015 at IIIT Hyderabad, India [40].

Some studies on sentiment analysis for Hindi language have been presented in [41][42][43].

Some baseline results on aspect-based sentiment analysis has been reported in [44]. Deep learning based sentiment analysis system that uses LSTM and CNN for sentiment analysis of Bengali tweets has been presented in [45] and [38].

Unlike the prior work on tweet sentiment analysis for Bengali tweets, which applied Naïve Bayes and SVM [35] [46], we have used in this present work the

bidirectional recurrent neural networks named LSTM (Long Term Short Term Memory for sentiment classification. We have also compared the sentiment classification performances of LSTM, BiLSTM and CNN.

3 Model Development

Our proposed system for sentiment polarity detection uses a special kind of deep learning models. Our proposed system has several components- (1) preprocessing (2) development of deep learning models

3.1 Preprocessing

Since tweet data is noisy, we preprocess our data to clean unimportant characters. From tweets in our corpus, the characters [, -, _, =, :, +, \$, @, ~, !, ,, ;, /, ^,), (, ., {, }, <, > are removed.

At this stage, we use SentiWordNet to augment the tweet terms by searching terms in SentiWordNet and retrieving the corresponding sentiment tags. We have used for our present work, the SentiWordNet [47], downloaded from <http://amitavadas.com/sentiwordnet.php>. This is basically a collection of three types of polarity words-positive, negative and neutral words. How the terms of a tweet are augmented is illustrated as follows. For example, the following two tweets:

Tweet1: *অতি অসাধারণ <Positive> ! ক্যামেরার কাহিনী দেখে পুরাই চমৎকার হইয়া গেলাম!* (Awesome! I was totally amazed by the camera's story!)

Tweet2: *ঝড়ন্য মুভি, আমার কাছে মনে হয়েছে পুরো ৩ টা ঘন্টা নষ্ট করেছি* (Damn movie, I feel like I've wasted an entire hour)

They are augmented as follows:

Tweet1: *অতি অসাধারণ <Positive> ! ক্যামেরার কাহিনী দেখে পুরাই চমৎকার <positive> হইয়া গেলাম!*

Tweet2: *ঝড়ন্য <negative> মুভি, আমার কাছে মনে হয়েছে পুরো ৩ টা ঘন্টা নষ্ট <negative> করেছি*

Since the words “অসাধারণ (Awesome)” and “চমৎকার (Excellent)” of tweet1 are found in SentiWordNet, they are augmented with <positive>. Since the words “ঝড়ন্য (Damn)” and “নষ্ট (wasted)” of tweet2 are found in SentiWordNet, they are augmented with <negative>.

This augmentation is done because we observe that the external knowledge of word level sentiments can help in classifying overall polarity of tweets. After processing all the training tweets in the above mentioned way, they are submitted to the deep learning algorithms for model development.

3.2 Deep Learning Models

For implementing our proposed sentiment classification system, we have considered three deep learning algorithms - (1) Recurrent Neural Networks called LSTM, (2) Bidirectional Recurrent Neural Networks called BiLSTM and (3) Convolutional Neural Networks (CNN). Unlike the traditional sentiment analysis model that uses multinomial Naïve Bayes and SVM along with bag-of-words tweet representation [35] [46], the deep learning based models can take into account the entire sequence of tokens contained in a tweet while detecting sentiment polarity of the tweet. Since the deep learning models with word embedding layer transforms the input words into the dense word vectors, the data sparseness problem arising out from the bag-of-words tweet representation used by the traditional machine learning based models is overcome to some extent by the deep learning models.

3.2.1. LSTM and BiLSTM based Models

The important feature of LSTM based deep learning based model is that it can process sequence of tokens in the input tweet to encode it to a contextual embedding that passes through the final *softmax* layer that classifies the tweet into one of the three sentiment classes - positive, negative and neutral.

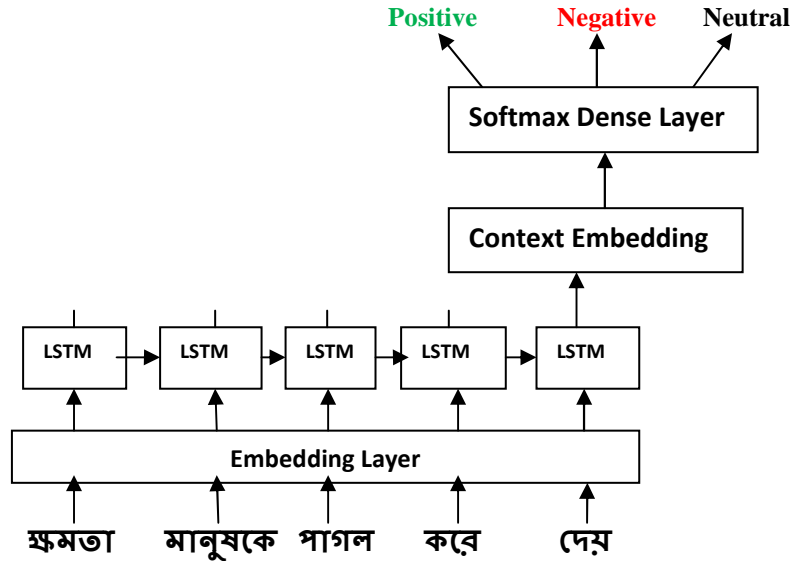


Figure 1. Unrolled LSTM over a sample input tweet
 “ক্ষমতা মানুষকে পাগল করে দেয়” (Power makes a man mad)

Thus the LSTM based model can make use of the appropriate context words in the sequence of tweet words for sentiment polarity prediction.

A LSTM (Long Term Short Term Memory) is a special type of recurrent neural networks, which can process relatively long sequence of data in better way because it does not suffer from so-called gradient vanishing problem [48]. The details of LSTM with its working principle can be found in [49].

The outline of the broad architecture that we have used for developing our LSTM based sentiment polarity detection model is given in Figure 1. The output of LSTM unit goes to the softmax layer which predicts the class of the input tweet.

While LSTM can process the input in only forward direction (from the left to right of the input word sequence), the bidirectional LSTM (BiLSTM) can capture the better contextual information by processing the input word sequence in both the forward and backward directions.

In practice, architecture involves two recurrent layers side-by-side, then providing the input sequence as input to the first layer and providing a reversed copy of the input sequence to the second. The outputs of two parallel LSTM layers are finally merged for improving the performance of Recurrent Neural Networks (RNNs). Figure 2 shows how bidirectional LSTM works.

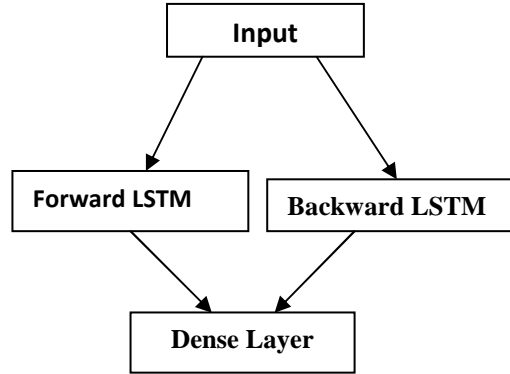


Figure 2. Bidirectional LSTM Architecture

For implementation of our LSTM based deep learning models for sentiment polarity classification, we have used Keras open source library written in Python for deep learning.

3.2.2. CNN based Model

We have also used one dimensional CNN (Convolutional Neural Networks) for tweet polarity classification. The input to CNN is formed by concatenating the word vectors of the tokens contained in a tweet. To deal with the variable tweet length problem, zeros are added for equalizing the length. CNN effectively helps

in convolving the effect of local (word level) sentiment polarity with the semantics of tweet words. Moreover convolution helps to incorporate the surrounding information in feature computation.

The architecture of the CNN model that we have used for sentiment polarity classification of Bengali tweets is shown in Figure 3. This architecture is similar to that is given in [38]. In Figure 3, the input to CNN is an $m \times k$ matrix representation of a tweet where m = the number of words in the tweet and k = the embedding dimension, that is, k is the size of the dense word vector representing each tweet word. The convolutional layer convolves the input using a filter whose size is $W = h \times k$, where h is the window size and k is the word vector's dimension. This convolutional operation gives a new feature f_i when a filter W is applied to the window of h words:

$$f_i = g(W \cdot X_{(i:i+h-1)} + b)$$

Where $X_{(i:i+h-1)}$ is the vector formed by concatenating the word vectors corresponding to the words falling in the window that spans from the i^{th} word to the $(i+h-1)^{\text{th}}$ word in the tweet, b is the bias weight and g is the non-linear transformation functions such as hyperbolic tangent or rectilinear.

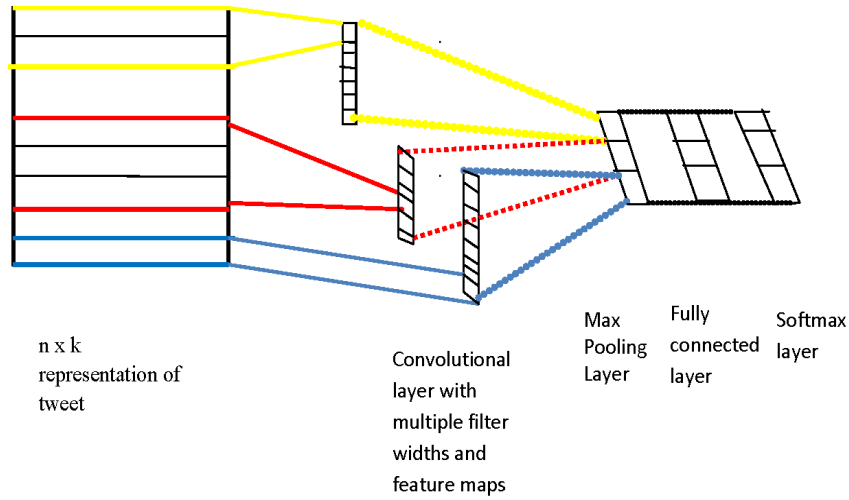


Figure 3. CNN Architecture

The filter is slid right with stride equal to 1 and each possible window of word vectors are convolved to a new feature. This gives a feature map:

$$f = [f_1, f_2, \dots, f_{(m-h+1)}]$$

Then max pooling [35] is applied on all the obtained feature maps (the number of filters determine number of feature maps) to select the most salient

feature per feature map f . Using pooling the problems related to the variable tweet length are also overcome.

Thus each filter can give one convoluted feature and we obtain a feature vector of length N if N filters are used (For the different filters, the different window sizes can be used).

The output of the convolution layer is then connected to a fully connected (FC) layer which receives as input a vector of dimension N . Here the convolutional layer acts as feature learning layer. The output of this fully connected layer is then passed to *softmax* layer which gives the probability distribution over three class labels-positive, negative and neutral.

For implementing the CNN based model, we have used Keras open source library written in Python for deep learning. The model is developed using Keras sequential model. Figure 4 shows the various layers used for implementing the CNN model.

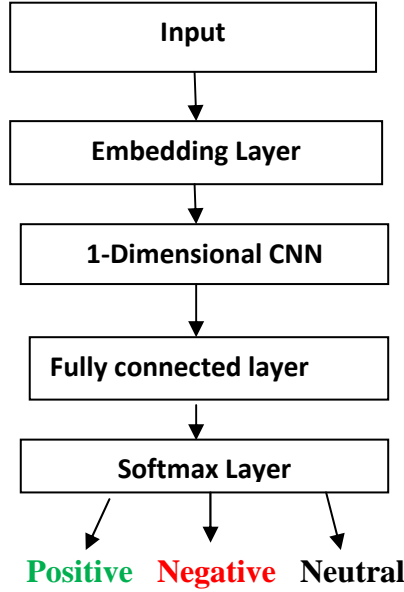


Figure 4. Steps of our implementation of CNN model

4 Evaluation and experimental results

We have used Bengali sentiment analysis datasets used in the shared task named SAIL@MIKE 2015, held at IIIT Hyderabad, India [40]. The distribution of positive, negative and neutral tweets in the training set and the size of the test set is shown in Table 1. The training set contains a total of 1000 tweets and the test contains a total of 500 tweets.

Since the deep learning methods are data hungry, this dataset is not sufficient for training deep learning models. However, we have used this dataset because this is the only dataset on which many existing Bengali sentiment analysis techniques had been tested.

Table 1. Description of SAIL 2015 Bengali dataset

Training set			Test set
<i>Negative</i>	<i>Positive</i>	<i>Neutral</i>	
356	368	276	500

4.1. Experimental Results and Comparisons

Since training set is relatively small, we combine the MIKE training set and test set to form a single dataset consisting of 1500 tweets and 10-fold cross validation is done for model performance comparisons. For each of our developed models, we compute average accuracy over 10 folds and the models have been compared based on the obtained average accuracy.

We conducted several experiments to judge effectiveness of deep learning models for sentiment polarity classification.

Experiment 1: In this experiment, we have developed a simple LSTM based model which has been discussed in section 3.

Experiment 2: In this experiment, we have also developed an bidirectional LSTM model as discussed in section 3.

Experiment 3: In this experiment, we have developed a CNN based model as discussed in section 3.

4.1.1. Model Configuration

The deep learning models have several tunable hyperparameters which need to be properly tuned to obtain the best possible results. We have tuned the hyperparameters in sequential manner. For example, initially hyperparameters are set to default values as per Keras implementation of LSTM based model, then batch size is tuned by varying its value from low values to high values and the batch size that gives the best results is noted. Now by setting the batch size to the obtained optimal value, other hyperparameters except embedding size are set to the default values and embedding size is tuned. Thus the best configuration of each model is found.

We have obtained the best results for our LSTM based model with the following configurations:

embedding_size = 100
 SpatialDropout = 0.4
 Dropout = 0.2 and recurrent_dropout = 0.2
 dimension of the output space = 100

Loss function ="categorical_crossentropy"
 optimizer= "Adam"
 epochs = 25
 batch_size =100

For the bidirectional LSTM model, we obtained the best results with the following settings of the model hyperparameters:

embedding_size = 32
 Dropout =0.2
 dimension of the output space = 32
 Loss function ="categorical_crossentropy"
 optimizer= "RmsProp"
 epochs = 20
 batch_size =64

For the CNN base model, the best results are obtained with the following model configuration:

embedding_size = 32
 Activation of CNN layer="relu"
 Number of filters=32, kernel_size=3, strides=1
 Max pooling size=3
 Size of the first dense layer = 32
 Activations of the dense layer="tanh"
 Dropout =0.2
 Loss function ="categorical_crossentropy"
 optimizer= "Adam"
 epochs = 20
 batch_size =64

The 10-fold cross validated results of our developed deep learning models with the above mentioned configurations are shown in Table 2.

Table 2. Performances Comparisons of Our Proposed Deep Learning based Models for Bengali Sentiment Classification

Models	Accuracy (%)
The BiLSTM based model	55.73
The LSTM based model	55.27
The CNN based model	51.93

As we can see from Table 2, BiLSTM performs the best among the three deep learning models we have implemented.

4.2 Model Comparisons and Discussion

By comparing the results of all the deep learning based models, we can observe that, for Bengali sentiment classification task, the BiLSTM based model can perform slightly better than the other deep learning models.

To judge the effectiveness of the deep learning models, we have compared them with the traditional machine learning algorithms. Since some previous works suggest that two machine learning algorithms- Multinomial Naive Bayes [35] and SVM with the linear kernel[46] are effective in Bengali sentiment polarity classification, we have chosen these two algorithms for comparisons with our developed deep learning based models. Multinomial Naive Bayes and SVM based models have been developed on WEKA¹ platform with the same feature set and parameter settings as mentioned in [35] and [46] respectively. The comparisons of the results are shown in Table 3.

Table 3. Performances Comparisons of Our Proposed Deep Learning based Model and Two Existing Machine Learning based Models for Bengali Sentiment Classification

Models	Accuracy (%)
The BILSTM based model	55.73
The LSTM based model	55.27
The CNN based model	51.93
SVM with unigram features [46]	53.73
Multinomial Naïve Bayes with unigram and bigram features[35]	53.07

It is evident from Table 3 that performances of LSTM and BILSTM based models are slightly better than the traditional machine learning algorithms- Multinomial Naive Bayes and SVM though the CNN based model performs like the traditional machine learning algorithms. BILSTM performs the best because of the fact that it analyzes the tweet from both sides: left-to-right and right-to-left, whereas the LSTM processes the tweet from left-to-right only.

We can also see that deep learning models outperform the traditional machine learning models by small margin. It is also unclear to us why CNN gives relatively poor performance. One of the possible reasons for such results of deep learning models may be the nature of SAIL 2015 dataset and the size of the dataset. Since deep learning model is highly data hungry and SAIL 2015 dataset is insufficient for training the deep learning models, the deep learning models could not outperform the traditional machine learning models by large margin. Though SAIL 2015 dataset is small, we have applied deep learning on this dataset for several reasons - (1) to the best of our knowledge, this is only benchmark dataset for Bengali tweet sentiment analysis, (2) *No free lunch* theorem states that no learning algorithm can outperform another when evaluated over all possible classification problems and (3) to investigate whether deep learning can be the automatic choice for Bengali

¹ <http://www.cs.waikato.ac.nz/ml/weka/>

tweet sentiment analysis task or not.

5 Conclusion

In this work, we have presented several deep learning based models for sentiment polarity classification of Bengali tweets. SentiWordNet has been incorporated in each model to utilize external knowledge.

We feel that we need more training data for improving accuracy of the deep learning models for sentiment polarity classification. With the availability of relatively large training data and the hybrid deep learning models such as BiLSTM with additional attention layers can be used for developing more enhanced deep learning model. For improving the performance of the CNN based model, transfer learning mechanism can be integrated with the model in future. With slight modifications, our proposed deep learning models can easily be applied to the tweet data sets for other Indian languages.

Acknowledgment. This research work has received support from the project entitled ‘‘Indian Social Media Sensor: an Indian Social Media Text Mining System for Topic Detection, Topic Sentiment Analysis and Opinion Summarization’’ funded by the Department of Science and Technology, Government of India under the SERB scheme.

References

1. Bowker, J.: The Oxford dictionary of world religions. Oxford University Press, USA, (1997)
2. Melville, P., Gryc, W., Lawrence, R. D.: Sentiment analysis of blogs by combining lexical knowledge with text classification. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 1275-1284 (2009)
3. Ramakrishnan, G., Jadhav, A., Joshi, A., Chakrabarti, S., Bhattacharyya, P.: Question answering via Bayesian inference on lexical relations. In Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering, Association for Computational Linguistics, Vol. 12, pp. 1-10 (2003)
4. Jiao, J., Zhou, Y.: Sentiment Polarity Analysis based multi-dictionary. Physics Procedia, 22, 590-596 (2011)
5. Macdonald, C. , Ounis, I.: The TREC Blogs06 collection: Creating and analysing a blog test collection. Department of Computer Science, University of Glasgow Tech Report TR-2006-224, 1, 3-1 (2006)
6. Hatzivassiloglou, V., McKeown, K. R.: Predicting the semantic orientation of adjectives. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, pp. 174-181, Association for Computational Linguistics, July, (1997)

7. Wiebe, J.: Learning subjective adjectives from corpora. In AAAI/IAAI, pp. 735-740, July (2000).
8. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In Proceedings of the 2003 conference on Empirical methods in natural language processing, Association for Computational Linguistics, pp. 105-112, July (2003)
9. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In Proceedings of LREC, Vol. 6, pp. 417-422, May (2006)
10. Fellbaum, "WordNet," Blackwell Publishing Ltd., 1999.
11. Zhao, J., Liu, K., Wang, G.: Adding redundant features for CRFs-based sentence sentiment classification. In Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics, pp. 117-126 (2008)
12. Joachims, T.: Making large scale SVM learning practical. Published in book-Advances in kernel methods, pp. 169-184 (1999)
13. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques, In Proceedings of the ACL-02 conference on Empirical methods in natural language processing, Association for Computational Linguistics, Vol. 10, pp. 79-86 (2002)
14. Dave, K., Lawrence, S., Pennock, D. M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings of the 12th international conference on World Wide Web, ACM, pp. 519-528 (2003)
15. Mullen, T., Collier, N.: Sentiment Analysis using Support Vector Machines with Diverse Information Sources. In EMNLP, Vol. 4, pp. 412-418 (2004)
16. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and trends in information retrieval. 2(1-2), 1-135 (2008)
17. Goldberg, B., Zhu, X.: Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, Association for Computational Linguistics, pp. 45-52 (2006)
18. Miao, Q., Li, Q., Zeng, D.: Fine grained opinion mining by integrating multiple review sources. Journal of the American society for information science and technology. 61(11), pp. 2288-2299 (2010)
19. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In Proceedings of the 2003 conference on Empirical methods in natural language processing, Association for Computational Linguistics, pp. 105-112 (2003)
20. Prabowo, R., Thelwall, M.: Sentiment analysis: A combined approach. Journal of Informetrics, 3(2), 143-157 (2009)
21. Narayanan, R., Liu, B., Choudhary, A.: Sentiment analysis of conditional sentences. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Vol. 1, pp. 180-189 (2009)
22. Wiegand, M., Balahur, A., Roth, B., Klakow, D., Montoyo, A.: A survey on the role of negation in sentiment analysis. In Proceedings of the workshop on negation and speculation in natural language processing, Association for Computational Linguistics, pp. 60-68 (2010)
23. Yu, H., Hatzivassiloglou, V.: Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In Proceedings of the 2003 conference on Empirical methods in natural language processing, Association for Computational Linguistics, pp. 129-136, July (2003)
24. Ku, L., Liang, Y., Chen, H.: Opinion Extraction, Summarization and Tracking in News and Blog Corpora. In AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, (2006)
25. Kim, J., Chern, G., Feng, D., Shaw, E., Hovy, E.: Mining and assessing discussions on the web through speech act analysis. In Proceedings of the Workshop on Web

- Content Mining with Human Language Technologies at the 5th International Semantic Web Conference (2006)
26. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd annual meeting on Association for Computational Linguistics (p. 271) (2004)
 27. Zhu, F., Zhang, X.: Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics. *Journal of Marketing*. 74(2), 133-148 (2010)
 28. Pang, B., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In proceedings of ACL (2004)
 29. Chen, C., Tseng, Y. D.: Quality evaluation of product reviews using an information quality framework. *Decision Support Systems*. 50(4), 755-768 (2011)
 30. Kang, H., Yoo, S. J., Han, D.: Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*. 39(5), 6000-6010 (2012)
 31. Clarke, D., Lane, P., Hender, P.: Developing robust models for favourability analysis. In Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, Association for Computational Linguistics, pp. 44-52 (2011)
 32. Reyes, A., Rosso, P.: Making objective decisions from subjective data: Detecting irony in customer reviews,” *Decision Support Systems*, 53(4), 754-760 (2012)
 33. Moraes, R., Valiati, J. F., Neto, W. P. G.: Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*. 40(2), 621-633 (2013)
 34. Martín-Valdivia, M. T., Martínez-Cámara, E., Perea-Ortega, J. M., Ureña-López, L. A.: Sentiment polarity detection in Spanish reviews combining supervised and unsupervised approaches. *Expert Systems with Applications*, 40(10), 3934-3942 (2013)
 35. Sarkar, K., Chakraborty, S.: A Sentiment Analysis System for Indian Language Tweets. In: Mining Intelligence and Knowledge Exploration, Springer, LNAI, Vol. No. 9468, pp. 694-702 (2015)
 36. Li, Y. M., Li, T. Y.: Deriving market intelligence from microblogs. *Decision Support Systems*, 55(1), 206-217 (2013)
 37. Ouyang, X., Zhou, P., Li, C., Hua, L., Liu, L.: Sentiment analysis using convolutional neural network. In: proceedings of IEEE International Conference on Computer and Information Technology, Ubiquitous Computing and Communications, Dependable, Autonomic and Secure Computing, Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM). pp. 2359-2364 (2015)
 38. Sarkar, K.: Sentiment Polarity Detection in Bengali Tweets Using Deep Convolutional Neural Networks. *Journal of Intelligent Systems*. 28(3), 377-386 (2019)
 39. Wang, X., Jiang, W., Luo, Z.: 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, pp. 2428-2437 (2016)
 40. Patra, B. G., Das, D., Das, A., Prasath, R.: Shared task on sentiment analysis in indian languages (sail) tweets-an overview. In International Conference on Mining Intelligence and Knowledge Exploration, pp. 650-655, Springer, December (2015)
 41. Arora, P.: Sentiment analysis for Hindi language. MS by Research in Computer Science. (2013)
 42. Joshi, A., Balamurali, A. R., Bhattacharvya, P.: A fall-back strategy for sentiment analysis in hindi: a case study. Proceedings of the 8th ICON (2010)
 43. Sharma, Y., Mangat, V., Kaur, M.: A practical approach to sentiment analysis of Hindi tweets. In 2015 1st International Conference on Next Generation Computing Technologies (NGCT). pp. 677-680 (2015, September)

44. Rahman, M., Kumar Dev, E. Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation. *Data*, 3(2), 15 (2018)
45. Sarkar, K.: Sentiment Polarity Detection in Bengali Tweets Using LSTM Recurrent Neural Networks. 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 2019, pp. 1-6. doi: 10.1109/ICACCP.2019.8883010 (2019)
46. Sarkar, K., Bhowmik, M.: Sentiment Polarity Detection in Bengali Tweets Using Multinomial Naïve Bayes and Support Vector Machines. *CALCON 2017*, Kolkata, (2017)
47. Das, A., Bandyopadhyay, S.: SentiWordNet for Indian languages. In: *Proceedings of 8th Workshop on Asian Language Resources (COLING 2010)*, Beijing, China, pp. 56–63 (2010)
48. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*. 5(2), pp-157-166 (1994)
49. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Computation*. 9 (8), doi:10.1162/neco.1997.9.8.1735, pp.1735–1780 (1997)