Evaluation of Naïve Bayes and Support Vector Machines on Bangla Textual Movie Reviews

Nayan Banik
Department of Computer Science & Engineering
Comilla University
Comilla - 3506, Bangladesh
cse.nayan@gmail.com

Md. Hasan Hafizur Rahman
Department of Computer Science & Engineering
Comilla University
Comilla - 3506, Bangladesh
hhr@cou.ac.bd

***Abstract*—The ever-growing World Wide Web (WWW) containing textual movie reviews from Facebook groups, personal blogs, newspapers, dedicated movie review sites has crucial impacts on intended users and review providers where users can find the desired movies in his preference by checking the reviews and review providers can build a better recommendation system that influences the marketing policy and revenues generated from advertising campaigns. The manual approach to analyze this textual reviews is complex and time-consuming and hence it requires specialized automated systems. Sentiment Analysis (SA) is such a tool which is the computational study of extracting opinions, emotions from textual data to build such automated systems and it has been implemented in many languages for movie review domain except Bangla; which is continuously increasing due to the emergence of Bangladeshi reviewers on WWW. Since none of the existing SA works addressed this domain, in this paper, we have developed a polarity detection system on textual movie reviews in Bangla by using two popular machine learning algorithms named Naïve Bayes (NB) and Support Vector Machines (SVM) and provided a comparative results where SVM performed slightly better than NB by considering stemmed unigram as feature with an excellent precision of 0.86.**

Keywords*—*Sentiment Analysis*, *Polarity Detection*, *Natural Language Processing

I. Introduction

Watching movies is a common pastime for people to diversify themselves from the day-to-day monotonous life where a person can enjoy good movies in a cinema hall and modern cineplex distributed in a city. For this purpose, the movie industry is continuously booming to increase their revenue by selling movie tickets during the play. From a viewers perspective, buying movie ticket costs money as well as the time to finish watching it. The personal preference to select a movie to watch is generally influenced by the trailers provided earlier by the movie makers where they try to demonstrate the movie as a must watch and the best one. But sometimes the original movie may not be so entertaining as the trailers and the viewer may realize that the time and money is wasted. In order to overcome this scenario, it is required to analyze unbiased textual movie reviews from other viewers providing accurate and overall pictures of the movie which includes both positive and negatives aspects including actors, their performances, direction standards, script and story, quality of screenplay and background scores [1].

The recent outburst of WWW provides a tremendous amount of textual movie reviews submitted by existing viewers from diverse sources including Facebook groups, personal blogs, newspapers, dedicated movie review sites and so on where a new viewer can browse through his desired movie categories and check the reviews posted by others. Among these sources, some are so popular that they get huge traffic on the Internet every day and some of them are online for several years [2]. The success of such sites depend on proper analysis of posted reviews which is important for both user and site management because if there are lack of clear, complete and unbiased reviews, a user may not visit the site and from the business perspective, better review analysis is crucial for creating a better recommendation system which positively affects the sites marketing policy and revenues from the advertising campaigns.

Manual analysis of textual reviews is quite impossible and time-consuming because the number of reviews posted every day is huge and there are no strict rules for the reviewers to follow. Information extraction from these vast amount of textual data to build automated system can be done through Sentiment Analysis (SA), which plays a great role as a Natural Language Processing (NLP) tool to detect some major properties of the text like subjectivity detection (whether the text is subjective or objective), subjective texts polarity detection (whether the text is positive, negative or neutral), emotion detection etc. SA is the computational study of peoples opinions, appraisals, attitudes and emotions toward entities, individuals, issues, events, topics and their attributes where it may use statistical as well as machine learning tools for analyzing the textual data which are important for organization and human decision making process [3] [4]. This analysis can be done from several perspectives including document level, sentence level, aspect level and the applied techniques can be supervised, unsupervised or case-based reasoning [5] [6]. For supervised approach, it uses classical machine learning algorithms like Naïve Bayes (NB), Support Vector Machines (SVM), Maximum Entropy (ME) whereas Latent Dirichlet Allocation (LDA), theme-based clustering are used in unsupervised approach and recently modern deep learning approaches are also incorporated to SA [7] [8] [9]. As a part of NLP, SA get notable importance to the research community for the last decade [10] [11] where many languages like Arabic, Hindi, Chinese, Czech, French, German, Italian, Japanese, Russian, Thai have implemented this analysis to enrich their linguistic area for knowledge extraction [12].

Existing and ongoing research on SA in Bangla language touches many sectors like food reviews, product reviews, horoscopes etc but none of them has addressed the movie review domain which is currently in a formative stage. In this paper, we have developed a polarity detection system on textual movie reviews in Bangla using supervised machine learning approach where we have manually collected 800 reviews by crawling some Bangla movie review sites and social media. In the preprocessing step, we have cleaned up the data by removing URLs, emoticons, punctuation characters. We have also stemmed the data to its root forms to minimize the feature space. Then the data are vectorized using the popular $TF-IDF$ vectorizer and feed to NB and SVM classifiers. The evaluation of our developed system has provided an acceptable precision of *0.86*.

This paper is organized as follows. In **Section II**, related works on SA in Bangla language is discussed. **Section III** describes our approach and methodology. Experimental results are provided in **Section IV** and the paper concludes with the conclusion in **Section V**.

II. Related Works

Bangla is a highly inflected language with free-form complex sentence structures and it lacks required NLP tools for doing research in this linguistic domain [13] [14]. In order to address these limitations, many researchers have done their research on SA task in Bangla like in [15], the authors come up with an idea of data standardization using the automatic translation of positive and negative words of SentiWordNet [16]. But Since no corpus is created, the work is limited to only word level sentiment which is less useful for complex analysis and it is also incapable of considering various inflected word terms, spelling errors, colloquial terms etc. Therefore a small dataset of 999 Bangla tweets for training and 499 for testing is used while the authors demonstrate the outcome of shared SA task of Indian languages in [17]. Despite some preprocessing and manual annotation, the small size of the dataset is a limiting factor for modern deep learning techniques. Moreover, in [18], the authors present a theme cluster-based approach to detect the subjectivity of a corpus from multiple domain perspectives for any less computerized languages like Bangla without having any vast linguistic knowledge. For sentiment lexicon generation, the authors use SentiWordNet [16] lexical resources from English to Bangla by utilizing a threshold for SentiWordNet scoring value. Moreover, they use the Samsad English-Bangla Dictionary [19] while the datasets used are manually annotated Multi-Perspective Question Answering (MPQA) [20] corpus and International Movie Database (IMDB) [21]. Using Parts Of Speech (POS), chunk, sentiment lexicon, stemming, frequency and positional aspect as features, the experimental result shows precision values of 72.16% and 74.6% for Bengali news and blog domains respectively. Relating to that, document level SA is done in [22], where the authors propose valency analysis for polarity detection in Bangla language texts using WordNet [23] and SentiWordNet [16]. First, word level senses are used to predict the sentence level polarity which is then used to detect the document level polarity and to evaluate the proposed system, they use expert analysis which justifies the approach to a satisfactory limit. But the drawbacks of this work is the requirement of Bangla text translation into English for analysis. In another work, Using Mutual Information (MI) [24] as a feature extractor, the authors in [25] show that Multinomial NB [26] provides slightly better performance for Bangla compare to English where the dataset is Amazon Watch's Reviews for English and the word by word translation of them for Bangla. With some preprocessing, the system's accuracy for Bangla is 84.78% (without negation) and 83.77% (with negation). In another work in [27], 9337 Bangla and Romanized Bangla social media texts are used in Recurrent Neural Network (RNN) [28]; specially Long Short Term Memory ($LSTM$) [29] considering binary cross entropy and categorical cross-entropy as loss functions. This is the first deep learning approach on SA task in Bangla where the proposed method scored 78% accuracy considering two categories as positive or negative. SA in Bangla horoscope domain is proposed in [30] where the authors provide a detail comparison on five classifiers named NB, SVM, K-Nearest Neighbours (KNN), Decision Tree (DT) and Random Forest (RF) with different parameters. They have also created a lexicon of 58 stop words in this process and the best sentiment classification model comprises of SVM with unigram features producing an accuracy of 98.7%. Polarity detection on product reviews for both Bangla and English is done in [31] where the authors translate 16000 watch reviews from Amazon into Bangla using Google Translator. They divide the polarity into weak, steady and strong where they achieve 85% accuracy for Bangla. Relating to that in [32], Facebook statuses are used for SA using NB models by collecting 3000 statuses from 70 users. The proposed approach removes the noise from data using some common preprocessing tasks and handling negation using antonym substitution. Considering bigram as feature the proposed system acquires F-score 0.72. In another work, SVM has outperformed ME while twitter posts are classified as positive or negative using semi-supervised bootstrapping methods for preparing the data in [33]. The authors have considered several features including word n-grams, stemming, emoticon, lexicon, POS tag, negation and combination of features. In [34], the authors predict the opinion polarity for Bangla news texts by considering several features in the SVM classifier. The approach first classifies the subjective texts from the objective ones using sentiment lexicons from SentiWordNet(Bangla) [35] and then it uses POS tags, chunk-level information, functional words, sentiment lexicon, stemming cluster, negative words and dependency tree feature as features in SVM classifier having a precision of 70.04%.

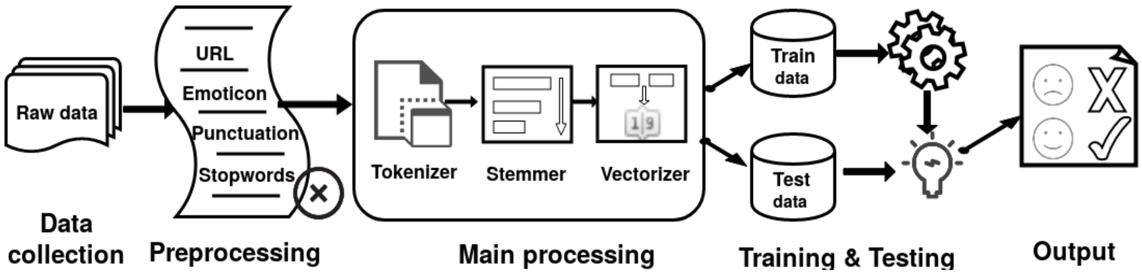

Fig. 1. System Structure

Without providing any evaluation metrics in [36], authors have claimed that dictionary-based approach gives better accuracy than lexicon-based approach while outperforming NB classifier for classifying the social media posts since they are not well written, formatted and standardized.

III. Methodology

Our automated polarity detection system is built using a supervised machine learning approach and in order to develop it, we have collected the raw data from different sources and manually annotate them. Then we have removed noise by applying some preprocessing steps and the processed data are then tokenized and stemmed for extracting features. Before feeding the data to the classifiers, we have vectorized them and finally generate the performance report. The overall system structure is depicted in Fig 1 and the whole process is described accordingly.

A. Data Collection

The Bangla movie review texts were collected from several independent sources including Facebook movie review related groups, pages, twitter posts of famous reviewers and Bangla Movie Database(BMDb)[1]. In order to collect them, we had used both custom rule-based crawler and manual collection through copy-pasting. The dataset contained 800 movie reviews in Bangla and each of them was manually annotated by two independent annotators so that we could get unbiased polarity. The raw dataset consisted of 400 positive reviews marked by 1 and 400 negative reviews marked by 0.

B. Preprocessing

The collected raw data from the Internet contained many noises since text-based reviews are not in fixed length, not properly structured and do not follow any specific standards. Sometimes the reviews may contain excessive and unimportant data for analysis and hence they require some processing. The Fig.2 represents a snippet of Bengali text document that consists of texts along with punctuations, hash tags and URLs. Our preprocessing steps included the following tasks:

a) URL removal: Many reviewers provide links on his reviews for either a specific point on the movie clip, media or related information. Since these URLs do not convey any sentiment from the reviews, we removed them in the first place.

'আয়নাবাজি' দেখে এলাম; যেমন ভেবেছিলাম, তার চেয়েও হাজার গুন ভালো লেগেছে :) ছবির এই দৃশ্য অসাধারণ ছিল http://images.myreviewsite.com/aynabaji.jpg। সময় করে দেখে নিন। #আয়নাবাজি #Aynabaji

Fig. 2. A snippet of document written in Bengali language

b) Emoticon removal: Social media texts include small emoticons where the user can show his feeling about a topic. But since we have considered only text-based reviews, we removed these emoticons.

c) Punctuation character removal: Bangla language uses many punctuation characters in texts which bears a little importance in sentiment polarity. So, a simple script was used to strip all punctuation characters from the data.

d) Stop words removal: A stop word is a word which occurs frequently in the dataset but has no sentiment polarity associated with it and thus can be filtered out before or after processing of natural language data (especially in text processing) [37]. In sentiment analysis, many of the words in English like 'the', 'for', 'and', 'on', 'to' etc do not convey any significance. Similarly, there are also some stop words in Bangla that We can discard from the input text as the overall polarity of a review does not depend on those words. Removing those stop-words in the early stage considerably improves the processing speed afterward. The obvious advantage of removing the stop words is the reduction in the number of features that the machine learning algorithms have to deal with. As there is no single universal list of stop words used by all processing of natural language tools, and indeed not all tools even use such a list, we had considered those words mentioned in an open source project work[2]. The Fig. 3 shows the output of Fig. 2 after removing stop-words, punctuations, hash tags and URLs to understand the significance of this process.

আয়নাবাজি দেখে এলাম ভেবেছিলাম হাজার গুন ভালো লেগেছে ছবির দৃশ্য অসাধারণ সময় দেখে নিন

Fig. 3. Stop-Words, Punctuation, Hash Tags and URL remove from Fig. 2

[1] https://bmdb.co/

[2] https://github.com/6/stopwords-json/edit/master/dist/bn.json

C. Main Processing

Main processing includes all the major steps required for our proposed system. The preprocessed data was first tokenized and stemmed from its root forms and features were selected from them. Selected features were then vectorized so that our classification algorithm could process them and generate the performance report. Details of each step are described below:

a) Tokenization: Text segmentation or Tokenization is the process of dividing the written text into meaningful units, such as words, sentences, or topics. In our case, we had implemented a bag-of-word approach for our classifier. Thus we had tokenized the input sentences into words which act as the primary features. We had used native python string split functions to tokenize the words in sentences. The Fig. 4 demonstrates the process of tokenization of texts after removing stop-words, punctuations, hash tags, URLs of Fig. 2 to close look the process of recognized individual words in a system.

'আয়নাবাজি', 'দেখে', 'এলাম', 'ভেবেছিলাম', 'হাজার', 'গুন', 'ভালো', 'লেগেছে', 'ছবির', 'দৃশ্য', 'অসাধারণ', 'সময়', 'দেখে', 'নিন'

Fig. 4. Tokenization of words from Fig. 3.

b) Stemming: Reducing variant terms of a word into its basic form is referred to as stemming. For grammatical reasons, documents or texts uses different forms of a word, such as 'stems', 'stemmer', 'stemming', 'stemmed' where the root word is 'stem'. Additionally, there are families of derivationally related words with similar meanings. In many situations, it would be useful for searching and comparison for one of these words to return documents that contain another word in the set because a list of similar words may denote the same sentiment. The goal of stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. In the Fig. 5, we portrayed stemmed words of sentences those are given in Fig. 2 to comprehend the essence of a stemmer in our work. For the purpose of stemming, we had used the stemmer proposed in [38].

'আয়না', 'দেখ', 'এল', 'ভাব', 'হাজার', 'গুন', 'ভালো', 'লাগ', 'ছবি', 'দৃশ্য', 'অসাধারণ', 'সময়', 'দেখ', 'নে'

Fig. 5. Stemmed words from sentences of Fig. 2

c) Vectorization: Vectorization is the general process of turning a collection of text documents into numerical feature vectors. This specific strategy (tokenization, counting, and normalization) is called the Bag of Words or Bag of n-grams representation. Documents are described by word occurrences while completely ignoring the relative position information of the words in the document. In a large text corpus, some words may be very frequent (e.g. the, a, is in English) hence carrying very little meaningful information about the actual contents of the document. If we had to feed the direct term counts into a classifier those very frequent terms would shadow the frequencies of rarer yet more interesting terms. In order to re-weight the count features into floating point values suitable for use by a classifier, it is very common to use the TF-IDF transformation. TF means term-frequency while TFIDF means term-frequency times inverse document-frequency:

$$tf - idf(t, d) = tf(t, d) \times idf(t)$$

The Fig. 6 illustrates vectors of words of Fig. 2 after removing less important words and performing tokenization operation.

(আয়নাবাজি, 9), (দেখে, 34), (এলাম, 69), (ভেবেছিলাম , 54), (হাজার, 21), (গুন, 60), (ভালো, 45), (লেগেছে, 11), (ছবির, 76), (দৃশ্য, 51), (অসাধারণ, 75), (সময়, 23), (দেখে, 34), (নিন, 97)

Fig. 6. Vectorization of text from Fig. 4

d) Features: In our work, we had considered several features while classifying the reviews. They were unigram, bigram, unigram and stemmed words, bigram and stemmed words.

e) Classification: For our sentiment polarity detection purpose, we had used the Multinomial Naive Bayes classifier [39] [40], which is an augmented version of simple Naive Bayes classifier and linear kernel based Support Vector Machines [41] [42] as supervised machine learning algorithms. Apart from other kernels like Polynomial, Radial Basis Function (RBF), we had used Linear kernel in SVM because most of the time texts are linearly separable, containing a lot of features, works faster and there are only a few parameters to optimize [43]. These two algorithms are widely used for text classification tasks and hence we had applied them in our system using scikit-learn[3], which is a popular, open source machine learning library in Python.

IV. Experimental Results

In order to evaluate our polarity detection system, we had considered three popular evaluation metrics: precision, recall, F1-score (or f-measure) and the overall accuracy. They assume a binary classification problem and two classes - a positive one and a negative one. In our system, the positive class was 1, while the negative class was 0. These metrics are calculated by the following equations where TP, TN, FP and FN denotes True Positive, True Negative, False Positive and False Negative respectively and m is the sample size (that is, $TP + TN + FP + FN$):

$$Accuracy = \frac{TP + TN}{m}$$

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

$$F1 - score = \frac{(2 * Precision * Recall)}{(Precision + Recall)}$$

The performance scores of our proposed system is provided in Table I and the corresponding comparison and trends is visualized in Figure 7 and 8.

[3]http://scikit-learn.org

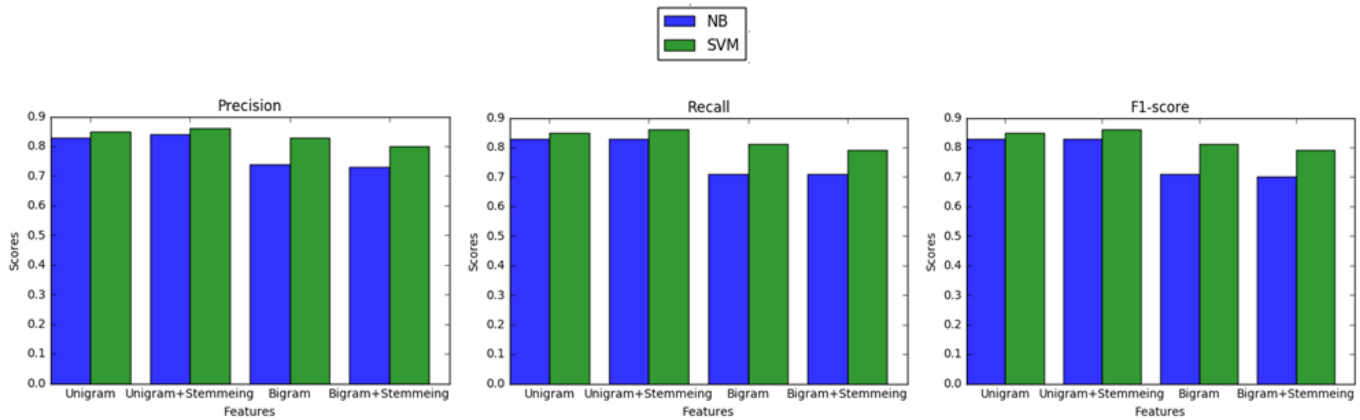

Fig. 7. Visualization of performance metrics

TABLE I
PERFORMANCE SCORES OF THE PROPOSED SYSTEM

Features	Precision	Recall	F1-score
Unigram + NB	0.83	0.83	0.83
Unigram + SVM	0.85	0.85	0.85
Bigram + NB	0.74	0.71	0.71
Bigram + SVM	0.83	0.81	0.81
Unigram + Stemming + NB	0.84	0.83	0.83
Unigram + Stemming + SVM	0.86	0.86	0.86
Bigram + Stemming + NB	0.73	0.71	0.70
Bigram + Stemming + SVM	0.80	0.79	0.79

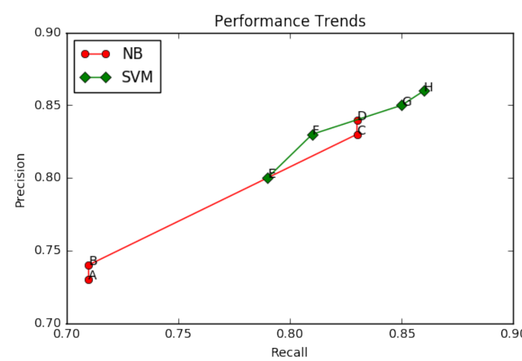

Fig. 8. Comparison and performance trends of the proposed system

In Figure 8, we have shown the performance of our proposed system for both classifiers in a precision-recall curve. Points marked in the figure denote our experimental results for different features. Point A and B on red line show that NB is not very sensitive to stemming as preprocessing step while considering bigram as features because the performance is similar in both cases. The same condition holds true while considering unigram as features in NB as marked by the points C and D on the red line. But from these observations, we can say that NB performs well with unigrams and stemming doesn't influence the performance. The precision for stemmed unigram in NB is 0.84 while the recall is 0.83. Point E on green line shows the performance of SVM with stemming as preprocessing and bigram as a feature. This is the lowest score for SVM with precision 0.80 and recalls 0.79. A slight increase is shown by point F when we remove the steaming step. The two nearest point G and H depicts the performance when SVM considers unigram features without and with stemming step. The best performance gets when we consider stemmed unigram as features resulting in the highest precision and recall of 0.86. From the Figure 7 it is seen that considering all features SVM performed better than NB and considering stemmed unigram as features in SVM provides the best precision and recall of 0.86. It can be said that bigram as the feature is providing the least performance scores in all scenarios. By further inspection, it is also seen that SVM is not influenced by stemming as the performance increase is very small. Recall and F1-score depict that the proposed system is also quite satisfactory for SVM compared to NB.

V. CONCLUSION

In this paper, an automated system is developed for detecting the polarity of textual movie reviews in Bangla. Due to the lack of standardized labeled dataset in this domain, reviews are collected from several online sources including Facebook and movie review sites using a custom crawler. After the manual annotation, the data is preprocessed for removing noise and reducing the feature space. Following that, the data is tokenized, stemmed and vectorized for classification. The developed system used $Na\ddot{v}e$ $Bayes$ (NB) and Linear kernal based Support Vector Machines (SVM) as a classifier where the experimental results showed that considering stemmed unigram as a feature in SVM performed best with a promising precision of 0.86. Since the system considered only Bag-of-words approach for a small dataset, the underlying semantic relationship among features is not captured properly. Semantic analysis of large data is required in this domain to explore better linguistic knowledge for sentiment extraction.

REFERENCES

[1] "Importance of movie reviews," http://www.abcarticledirectory.com/Article/Importance-of-Movie-Reviews/1379743, (Online: Accessed on June 14, 2018).

[2] "Alexa - top sites by category: Arts/movies/reviews," https://www.alexa.com/topsites/category/Top/Arts/Movies/Reviews, (Online: Accessed on June 16, 2018).

[3] B. Pang, L. Lee *et al.*, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[4] B. Liu, "Sentiment analysis and opinion mining," *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[5] A. Suriya and M. Prabakaran, "Opinion mining and sentiment analysis using data mining techniques-a survey."

[6] K. M. Rahman and A. Khamparia, "Techniques, applications and challenges of opinion mining."

[7] S. Sun, C. Luo, and J. Chen, "A review of natural language processing techniques for opinion mining systems," *Information Fusion*, vol. 36, pp. 10–25, 2017.

[8] K. D. Varathan, A. Giachanou, and F. Crestani, "Comparative opinion mining: a review," *Journal of the Association for Information Science and Technology*, vol. 68, no. 4, pp. 811–829, 2017.

[9] S. Mathapati, S. Manjula *et al.*, "Sentiment analysis and opinion mining from social media: A review," *Global Journal of Computer Science and Technology*, vol. 16, no. 5, 2017.

[10] T. Al-Moslmi, N. Omar, S. Abdullah, and M. Albared, "Approaches to cross-domain sentiment analysis: A systematic literature review," *IEEE Access*, 2017.

[11] R. Piryani, D. Madhavi, and V. K. Singh, "Analytical mapping of opinion mining and sentiment analysis research during 2000–2015," *Information Processing & Management*, vol. 53, no. 1, pp. 122–150, 2017.

[12] D. Vilares, M. A. Alonso, and C. Gómez-Rodr íguez, "Supervised sentiment analysis in multilingual environments," *Information Processing & Management*, vol. 53, no. 3, pp. 595–607, 2017.

[13] M. Karim, *Technical challenges and design issues in bangla language processing*. IGI Global, 2013.

[14] F. Alam, S. Habib, D. A. Sultana, and M. Khan, "Development of annotated bangla speech corpora," 2010.

[15] D. Das and S. Bandyopadhyay, "Developing bengali wordnet affect for analyzing emotion," in *International Conference on the Computer Processing of Oriental Languages*, 2010, pp. 35–40.

[16] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining." in *LREC*, vol. 10, 2010, pp. 2200–2204.

[17] B. G. Patra, D. Das, A. Das, and R. Prasath, "Shared task on sentiment analysis in indian languages (sail) tweets-an overview," in *International Conference on Mining Intelligence and Knowledge Exploration*. Springer, 2015, pp. 650–655.

[18] A. Das and S. Bandyopadhyay, "Subjectivity detection in english and bengali: A crf-based approach," 2009.

[19] "Digital dictionaries of south asia," http://dsal.uchicago.edu/dictionaries/, (Online: Accessed on June 18, 2018).

[20] "Mpqa resources," https://mpqa.cs.pitt.edu/, (Online: Accessed on June 18, 2018).

[21] "Data," http://www.cs.cornell.edu/people/pabo/movie-review-data/, (Accessed on June 18, 2018).

[22] K. A. Hasan, M. Rahman *et al.*, "Sentiment detection from bangla text using contextual valency analysis," in *Computer and Information Technology (ICCIT), 2014 17th International Conference on*. IEEE, 2014, pp. 292–295.

[23] A. Kilgarriff and C. Fellbaum, "Wordnet: An electronic lexical database," 2000.

[24] H. Liu, J. Sun, L. Liu, and H. Zhang, "Feature selection with dynamic mutual information," *Pattern Recognition*, vol. 42, no. 7, pp. 1330–1339, 2009.

[25] A. K. Paul and P. C. Shill, "Sentiment mining from bangla data using mutual information," in *Electrical, Computer & Telecommunication Engineering (ICECTE), International Conference on*. IEEE, 2016, pp. 1–4.

[26] D. Jurafsky and J. H. Martin, *Speech and language processing*. Pearson, 2014, vol. 3.

[27] A. Hassan, M. R. Amin, A. K. Al Azad, and N. Mohammed, "Sentiment analysis on bangla and romanized bangla text using deep recurrent models," in *Computational Intelligence (IWCI), International Workshop on*. IEEE, 2016, pp. 51–56.

[28] J. A. Bullinaria, "Recurrent neural networks," *Neural Computation: Lecture*, vol. 12, 2013.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] T. Ghosal, S. K. Das, and S. Bhattacharjee, "Sentiment analysis on (bengali horoscope) corpus," in *India Conference (INDICON), 2015 Annual IEEE*. IEEE, 2015, pp. 1–6.

[31] K. A. Hasan, M. S. Sabuj, and Z. Afrin, "Opinion mining using naïve bayes," in *Electrical and Computer Engineering (WIECON-ECE), 2015 IEEE International WIE Conference on*. IEEE, 2015, pp. 511–514.

[32] M. S. Islam, M. A. Islam, M. A. Hossain, and J. J. Dey, "Supervised approach of sentimentality extraction from bengali facebook status," in *Computer and Information Technology (ICCIT), 2016 19th International Conference on*. IEEE, 2016, pp. 383–387.

[33] S. Chowdhury and W. Chowdhury, "Performing sentiment analysis in bangla microblog posts," in *Informatics, Electronics & Vision (ICIEV), 2014 International Conference on*. IEEE, 2014, pp. 1–6.

[34] A. Das and S. Bandyopadhyay, "Opinion-polarity identification in bengali," in *International Conference on Computer Processing of Oriental Languages*, 2010, pp. 169–182.

[35] ——, "Sentiwordnet for bangla," *Knowledge Sharing Event-4: Task*, vol. 2, 2010.

[36] S. Akter and M. T. Aziz, "Sentiment analysis on facebook group using lexicon based approach," in *Electrical Engineering and Information Communication Technology (ICEEICT), 2016 3rd International Conference on*. IEEE, 2016, pp. 1–4.

[37] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2014.

[38] M. H. Seddiqui, A. A. M. Maruf, and A. N. Chy, "Recursive suffix stripping to augment bangla stemmer."

[39] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752. Citeseer, 1998, pp. 41–48.

[40] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?" in *CEAS*, vol. 17, 2006, pp. 28–69.

[41] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[42] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[43] "Linear kernel: Why is it recommended for text classification?" [Online]. Available: https://www.svm-tutorial.com/2014/10/svm-linear-kernel-good-text-classification/