Sentiment Analysis on Bangladesh Cricket with Support Vector Machine

Shamsul Arafin Mahtab
Department of CSE
Shahjalal University of Science and Technology,
Sylhet, Bangladesh
arafinmahtab@gmail.com

Nazmul Islam
Department of CSE
Shahjalal University of Science and Technology,
Sylhet, Bangladesh
nazmul.islam.6978@gmail.com

Md Mahfuzur Rahaman
Department of CSE
Shahjalal University of Science and Technology,
Sylhet, Bangladesh
mahfuzsustbd@gmail.com

***Abstract*—While social platform and news portal play a big role in Internet today, it also becomes the valuable medium for public opinions. We want to perform sentiment analysis in these public opinions. Many works have been done on sentiment analysis in different sectors for English language. But works in Bengali language are limited to only Bengali corpus and micro-blogging. So we have targeted a special sector which is Bangladesh Cricket where people express their opinions in their native Bengali languages on social medias in every moment. So we have prepared a dataset of three sentiment classes about Bangladesh Cricket from real people sentiments. We have processed our dataset by removing unnecessary words from the Bengali texts. Then we have used TF-IDF Vectorizer for vectorization and the classifier Support Vector Machine to classify our data.**

***Index Terms*—Sentiment Analysis, TF-IDF, SVM**

I. Introduction

Sentiment analysis, is the field of study for analyzing peoples opinion, sentiments and emotions for different sources like products, organizations, services, events, social issues. It is critical because it helps us see what people like and dislike about us, our brands, names or other aspects. User feedback from social media, website, call center agents, or any other source contains a treasure trove of useful information. But, it is not enough to know what users are talking about. We must also know how they feel. Sentiment analysis is one way to uncover those feelings. According to the Oxford dictionary, it is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether a person's attitude towards a particular topic, product, etc. is positive, negative, or neutral. Although full understanding of natural text language is well beyond capabilities of machines, statistical analysis can provide meaningful categorization of sentiments.

In our country people love cricket like a religion. So they have diverse sentiments for this game. In the world of cricket people are regularly expressing their various kinds emotions every time in various aspects. So this section has become an interesting area for us to analyze with real people emotions for cricket. Overall maximum of our data are well structured Bengali sentiments as people express their feelings for Cricket in their native languages. But there is a challenge to work on Bengali language because of the shortage of resources for Bengali Language Processing.

In this research we have extracted sentiments or opinions of people from news portal and social platform and then identified the overall polarity of texts as positive, negative or neutral. And then we shifted our work to detect praise, criticism and sadness. At first we have created a Google forms and labeled the data with these three classes. Then we preprocessed and extracted our data as features using TF-IDF and applied machine learning models. For classification, our first preference is Support Vector Machine as it gives good performance for low shape dataset. Besides we have used Decision Tree and Multinomial Naive Bayes. We have also gathered a lots of knowledge on Deep learning. We are holding it for our future works.

II. Related Works

Our work is inspired by some of previous work in these related fields and some of them are for our knowledge gaining. Our idea mostly inspired from [10] where they tried to identify different classes such as violence and emotions. And one of the notable works we have followed [18]. They work on finding emotions from text messages. They have used TF-IDF for increasing classification accuracy and Support Vector Machine for classification. Their approach is quite similar to our first research beginning. But, they have also used Vector Space Model (VSM) as the document representation model. In this paper [9], they proposed the work where they utilizes the naive Bayes and fuzzy classifier to classify tweets into positive, negative or neutral behavior of a particular person. In [12], they consider the problem of classifying documents not by topic but by overall sentiment determining whether a review is positive or negative. In [8], they have done their sentiment analysis in detecting insults and flames. This inspired us to choose and work on our class criticism in Bangladesh Cricket in our dataset.

Also most recent work [1] for Bengali language on Twitter data to find the polarity of a Bengali text if it is positive or negative. They performed Bangla Pos-Tagger Package for POS Tagging and Support Vector Machine and Maximum Entropy to do a comparative analysis on the performance of these two

978-1-5386-8207-4/18/$31.00

algorithms by experimenting with a various sets of features. We are interested to work with the POS-Tagger they have used for our future work. In article [2], they proposed multiple computational techniques like WordNet based, dictionary based, corpus based or generative approaches for generating SentiWordNet(s) for three Indian languages: Bengali, Hindi and Telugu. And in report [16], they aim to automatically extract the sentiment or polarity by using HMM to perform POS tagging and SVM classifier.

Also work [17] on finding sentiment such as positive and negative reviews over 2000 movie data. They have used TF-IDF and their classification is performed using Support Vector Machine provided by weka tool. They have considered unigram, bigram, POS tags of words and function words as feature set. In paper [14], they prepared gold standard Bengali-English code-mixed data with language and polarity tag for sentiment analysis purposes and discussed the systems they prepared to collect and filter raw Twitter data.

Another notable paper we have followed [7] where they have done the opinion mining and mood extraction where they classified the polarity of text like positive, negative and neutral. [5] has done the survey on sentiment analysis analyzing text classification on opinion mining. Besides [19] also done the survey of the sentiment analysis text data. [3] analyzed and tracked the emotions of English and Bengali texts. From these literature, we have learned the Bengali language Processing.

In addition, [11] focused on twitter micro blogging data and classified positive, negative and neutral from the data. And in [6], they focused on restaurants reviews and classified the polarity of the text. [13] applied several common machine learning techniques on twitter micro-blogging, including various forms of a Naive Bayes and a Maximum Entropy Model. We have also done research on finding emotions of people about Bangladesh cricket from Facebook and online newspaper Prothom-Alo. From all of the literature learning, we have come to the approach to work on using TF-IDF Vectorizer and SVM to classify our data.

III. Methodology

Our initial preprocessing for Bengali text data is performed using Python Natural Language Toolkitc (NLTK)[1] and for vectorization and classification using machine learning model, we have imported the tools of scikit-learn[2]. Our whole system, outlining the whole process, is stated below.

A. Dataset

From the beginning of our research, we tried to find a proper way to collect and prepare our main dataset as data collection is time expensive sometimes. For this reason, we want to optimize the time in data collection with proper labeling. Use of web scrapping can give us lots of data but the problem is that some data might not be well structured, noisy or standard for our thesis. So we have split our works by using two same standard datasets. Initially we have collected a dataset which is referred as Bengali dataset ABSA [15]. This dataset is about Bangladesh cricket related comments which is quite similar to the dataset we want to build. So we have primarily chosen this dataset to train and build our machine learning system for sentiment analysis.

This ABSA dataset contains 2979 data with 5 columns. All of the data are crawled and tagged from BBC Bangla. Though this dataset is made for aspect-based sentiment analysis, we ignored the aspect columns as we do not need to train aspect-based. We have only chosen the comment column and the target column containing **Positive**, **Negative** and **Neutral** classes, then trained this dataset.

Besides training ABSA dataset with our system, we have also built our main dataset. We have previously mentioned that we have collected the sentiments from public posts or comments from facebook group of Bangladesh Cricket and sports section of Prothom-Alo newspaper. Then we have created a Google form to label all these opinions with the classes **Praise**, **Criticism** and **Sadness**. As we have used this manual process, our overall data have become well structured and less noisy. The shape of our dataset is 1601 which contains 3 classes including praise with 513, criticism with 604 and sadness with 484 labeled data.

B. Preprocessing

As our research is in natural language data, we need to process the data as a cleaned version. So data formatting and data cleaning play a significant role for our system.

In the beginning we have separated all the words as tokens from Bengali texts. We have used Python NLTK for tokenizing. Our preprocessing ends after splitting all the words from natural language sentences as token. After tokenizing, we have collected a huge array which contains the Bengali stopwords [4]. This array includes the Bengali stopwords of so, in, they, but, or all of these as we do not need these words while training our model. In addition, we have manually listed an array for punctuations and Bengali numbers. All of these unnecessary list of words, numbers and punctuation marks are filtered in the initialization of TF-IDF vectorizer which is explained in Feature Extraction section.

C. Feature Extraction

Our text data requires special customized preparation before starting using it for predictive modeling. Text must be parsed and tokenized before using predictive model. All of the words need to be converted into number or floating point number to use as input for the machine which is referred as vectorization. In this sections, scikit-learn library provides easy-to-use tools to perform feature extraction of text data. A simple and effective model for thinking about text documents in machine learning is called the Bag-of-Words Model, or BoW. The model is so simple that it throws away all of the order information in the words and focuses on the occurrence of words in a document. We have previously used CountVectorizer to extract our data as feature which counts the occurrence of

[1]Leading platform for building Python programs to work with human language data.

[2]Simple and efficient open source tools for data mining and data analysis.

words. But TF-IDF is one of the most powerful to vectorize the data as feature. With TF-IDF, words are given weight TF-IDF measures relevance, not frequency. So we have replaced the word counts with TF-IDF scores across the whole dataset. The below parameters we have used to accurately filter our data with TF-IDF vectorizer.

```
ngram_range=(1,2),
analyzer='word',
lowercase=False,
stop_words=bangla_stopwords,
tokenizer=nltk_tokenizing,
sublinear_tf = True,
use_idf = True
```

Here we have used ngram with lower and upper boundary to affect the vocabulary. Besides the stopwords parameter of TF-IDF is assigned with our Bengali stop words. The tokenizing part is also done in this section and initialized with Python NLTK which is quite good for tokenizing. We have also applied sub-linear tf scaling with 1 + log(tf) and enabled inverse-document-frequency re-weighting. These all parameters have properly processed our dataset and enriched our current system. For this implementation, we have got an enriched accuracy in our lower size dataset.

D. Classifier Selection

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observations. This dataset may simply be bi-class (like identifying whether the sentiments is positive or negative or that the mail is spam or non-spam) or it may be multi-class too. Like in our analysis, we are using three classes in each dataset for classification. There are many machine-learning classifiers for text classification. After the proper study, we have decided to use Support Vector Machine with Linear kernel which gives better performance in lower shape dataset. The kernel defines the similarity or a distance measure between new data and the support vectors. We can use other kernels also such as a Polynomial Kernel and a Radial Kernel that transform the input space into higher dimensions. This is called the Kernel Trick. SVM has been found providing better accuracy in the case of classifying text. As SVM are binary classifiers, they are better suited in classifying polarity of a sentence. Since our work is to identify different types of emotions which is like binary classification. So Support Vector Machine is our chosen method. Besides we have also used default Decision Tree and for probabilistic model-based approach, we have used Multinomial Naive Bayes classifiers to compare and analyze our results.

We have chosen 10% data as our random test sets. Then we have trained our machine learning model with the rest 90% of the dataset. The trained model predicts from the test sets whether a public opinion is praise, criticism or sadness related.

IV. Experiment & Result Analysis

We have done experiment and result analysis for two of our datasets and for the machine learning models we have chosen to experiment. Here the precision, recall, f1-score and support result are given for each dataset.

Precision is the number of sentence in the test set that is correctly labeled by the classifier from the total sentences in the test set that are classified by the classifier for a particular class.

Precision = True Positive / (True Positive + False Positive)

Recall is the number of sentences in the test set that is correctly labeled by the classifier from the total sentences in the test set that are actually labeled for a particular class.

Recall = True Positive / (True Positive + False Negative)

F-measure is the weighted harmonic mean of precision and recall for a particular class.

F1-Score = 2 * ((Precision * Recall) / (Precision + Recall))

A. Trained on Our Dataset

Though our dataset shape is small, our system implementation and its result are quite worthy to show here. **Accuracy 64.596 %**.

TABLE I
Classification Report

Label	Precision	Recall	F1-Score	Support
Praise	0.80	0.73	0.76	51
Criticism	0.56	0.81	0.67	59
Sadness	0.63	0.37	0.47	51
avg / total	0.66	0.65	0.63	161

From our observations, we have discovered that criticism and sadness is similar in some cases. So we must improve our result with more data. Around 2000 labeled data for each classes will definitely give a satisfactory result for our current system.

B. Trained on ABSA Dataset

We have built a model but as our own dataset size is small, we have used same topic based ABSA dataset to be sure that our model performs well. **Accuracy 73.490 %**.

TABLE II
Classification Report

Label	Precision	Recall	F1-Score	Support
Praise	0.73	0.25	0.37	64
Criticism	0.74	0.97	0.84	208
Sadness	0.33	0.04	0.07	26
avg / total	0.70	0.73	0.67	298

From the above report we see that our current system performs good for ABSA dataset though we have not done enough preprocessing.

Besides using Support Vector Machine, we have used Decision Tree and Multinomial Naive Bayes to recheck our accuracy levels. We have used the models for both of the datasets. And the results are close.

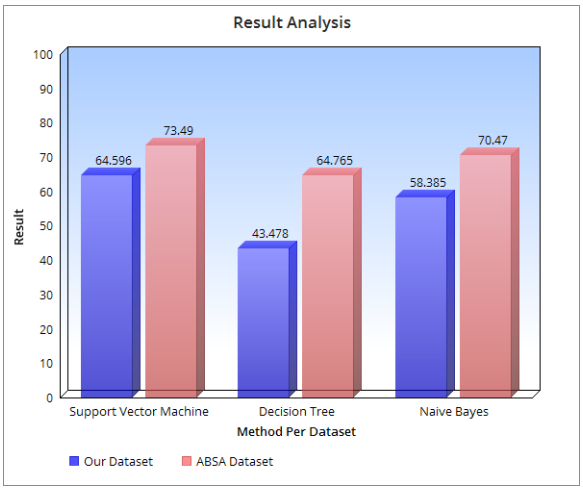

Fig. 1. Comparison between two datasets

This Analysis Shows that ABSA dataset result is a bit high for Support Vector Machine, Decision Tree and Multinomial Naive Bayes. This is because our dataset shape is small where ABSA dataset is double larger than our dataset. So the proper learning of our system with large data can give us better output.

V. Future Work and Conclusion

For future we need some improvement in our research. First of all, we have a limited amount of data where we have used 10% of our dataset as test set and found around 64% accuracy. Now our first target is to increase our dataset. Also, we will increase our target classes which is now only three. We will also try to improve our approach for better accuracy of our result. And of course we will apply deep learning theory for our existing system. The most important part is that we are working on Bengali language where we have not done the stemming, spell-checking and Bengali parts-of-speech tagging for our current research. Use of the proper natural language processing will highly improve our system. So we will definitely go on to workout with the accurate natural language processing.

Acknowledgment

Our deep thanks to our supervisor Md Mahfuzur Rahaman for his guidance, giving flexibility and continuous support throughout the work. We are also thankful to our family, friends for their support and encouragement. Finally, we thank SUST NLP Research Group and department of CSE, SUST for giving us their support throughout the thesis.

References

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

[2] A. Das and S. Bandyopadhyay. Sentiwordnet for indian languages. In *Proceedings of the Eighth Workshop on Asian Language Resouces*, pages 56–63, 2010.

[3] D. Das. Analysis and tracking of emotions in english and bengali texts: a computational approach. In *Proceedings of the 20th international conference companion on World wide web*, pages 343–348. ACM, 2011.

[4] G. Diaz. (2018, Oct.) Bengali stopwords. [Online]. Available: https://github.com/stopwords-iso/stopwords-bn.

[5] D. M. E. D. M. Hussein. A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences*, 2016.

[6] H. Kang, S. J. Yoo, and D. Han. Senti-lexicon and improved naïve bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, 39(5):6000–6010, 2012.

[7] A. Kaur and V. Gupta. A survey on sentiment analysis and opinion mining techniques. *Journal of Emerging Technologies in Web Intelligence*, 5(4):367–371, 2013.

[8] A. Mahmud, K. Z. Ahmed, and M. Khan. Detecting flames and insults in text. 2008.

[9] R. Mehra, M. K. Bedi, G. Singh, R. Arora, T. Bala, and S. Saxena. Sentimental analysis using fuzzy and naive bayes. In *Computing Methodologies and Communication (ICCMC), 2017 International Conference on*, pages 945–950. IEEE, 2017.

[10] S. M. Mohammad. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In *Emotion measurement*, pages 201–237. Elsevier, 2016.

[11] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[12] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[13] R. Parikh and M. Movassate. Sentiment analysis of user-generated twitter updates using various classification techniques. *CS224N Final Report*, 118, 2009.

[14] B. G. Patra, D. Das, and A. Das. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task@ icon-2017. *arXiv preprint arXiv:1803.06745*, 2018.

[15] A. Rahman. (2018, Oct.) Bengali ABSA dataset. [Online]. Available: https://github.com/AtikRahman/Bangla_Datasets_ABSA.

[16] A. Roy and A. A. Singh. (2018, Oct.) Sentiment Analysis ANLP Research Report. [Online]. Available: https://github.com/abhie19/Sentiment-Analysis-Bangla-Language/.

[17] P. H. Shahana and B. Omman. Evaluation of features on sentimental analysis. *Procedia Computer Science*, 46:1585–1592, 2015.

[18] J. D. Silva and P. S. Haddela. A term weighting method for identifying emotions from text content. In *Industrial and Information Systems (ICIIS), 2013 8th IEEE International Conference on*, pages 381–386. IEEE, 2013.

[19] H. Tang, S. Tan, and X. Cheng. A survey on sentiment detection of reviews. *Expert Systems with Applications*, 36(7):10760–10773, 2009.