4th International Conference on Electrical Information and Communication Technology (EICT), 20-22 December 2019, Khulna, Bangladesh

Emotion Detection from Bangla Text Corpus Using Naïve Bayes Classifier

Sara Azmin
Department of Computer Science & Engineering
Premier University
Chittagong, Bangladesh
azmin.sara17@gmail.com

Kingshuk Dhar
Department of Computer Science & Engineering
Premier University
Chittagong, Bangladesh
kingshuk2006@yahoo.com

***Abstract*— Emotions are an important part of everyday human interaction. Emotions can be expressed by means of written text, verbal speech or facial expressions. In recent years, the practice of expressing emotion in social media or blogs have increased rapidly. People write about their feelings and opinions on any political or global issues. All these social activities have made it essential to gather and analyze human emotion from the text. Although the field of emotion detection has been explored extensively for English language, the investigation of this domain for Bangla language still now in its infant stages. Our paper aims at detecting multi-class emotions from Bangla text using Multinomial Naïve Bayes (NB) classifier along with various features such as stemmer, parts-of-speech (POS) tagger, n-grams, term frequency-inverse document frequency (tf-idf). Our final model was able to classify the text into three emotion classes (happy, sad and angry) with an overall accuracy of 78.6%.**

Keywords — Emotion Detection, Machine Learning, Natural Language Processing, Bangla Text Processing, Naïve Bayes.

I. INTRODUCTION

Human emotion has always been a core interest to study in psychology. As they are an important element that helps understand any human nature. In psychology, emotion has been defined by many professors and specialists. Professor of Psychology, David G. Myers says, human emotion involves "...physiological arousal, expressive behaviors, and conscious experience." [1].

Nowadays the internet has made it easier to connect to the people of any part of the world. The recent growing usage of social media has caught the attention of computer science researchers. Especially in the study of human-computer interaction. Social media like Facebook, Twitter has created huge opportunities for its users to convey their feelings, opinions, feedbacks, and emotions through text. This has made it possible to analyze the emotion of people living in any part of the world on serious issues or crisis. It is also beneficial in the case of product reviews, market analysis.

Emotion detection from text is one of the core applications of artificial intelligence (AI) and Natural Language Processing (NLP). This is an important area of study to improve the interaction between humans and machines. Although this topic has been widely studied in the English language but it is still a less explored area in the case of Bangla language. Bangla is the 7th most spoken language in the world with nearly 228 million native speakers. Bangla is the official language of Bangladesh and a major part of Indian people also uses this language. At present, the number of internet subscriber of Bangladesh has reached a total of 91.421 million[1]. The current number of people in Bangladesh using Facebook is around 33,996,000[2]. Nowadays, a large number of people use Bangla to write on social media. For this rapid growth of Bangla users, it is quite important to focus on the study of emotion detection in Bangla Language. Most of the recent works in Bangla focuses on binary sentiment analysis. A lot of works has been done where positive is tagged as happiness emotion and negative as sadness emotion. But this is not sufficient to analyze a text. According to Ekman [2] - happiness, fear, anger, sadness, surprise, and disgust are the six basic human emotions. In social media, people do not just share their happy and sad feelings. They comment on different posts with anger, they write about their fear about any uncertainty and so on. Therefore, being one of the most widely used languages, the need for understanding the meaning of anything written in Bangla should be taken into account. This can be utilized in many areas such as market analysis, predicting public reaction in addition to so on.

In this paper, we have worked with a Bangla text corpus that contains comments from Facebook users on different topics. The proposed method preprocesses the corpus to simplify the classification process. It then classifies the data into three emotion class namely happy, sad, and angry using Multinomial Naïve Bayes Classifier.

II. RELATED WORKS

Not much of the work has been done for emotion detection in Bangla. Rather most of the paper has focused on sentiment analysis or binary sentiment polarity. A sentiment detecting approach using machine learning technique has been proposed in [3], in this paper they also analyzed some features but did not actually use them in their research. They focused on binary classification by using tf-idf classifier to find out the most informative words and got 83% accuracy using this approach.

A sentiment polarity detection on Bangla tweets using word and character n-grams with Naïve Bayes has been proposed in [4]. Authors also looked at the SentiWordNet feature, which is a lexical resource of sentiment polarity analysis. They classified the tweets using Multinomial NB. Using 1000 training and 500 test data, they achieved 48.5% accuracy.

[1] http://www.btrc.gov.bd/content/internet-subscribers-bangladesh-january-2019

[2] https://napoleoncat.com/stats/facebook-users-in-bangladesh/2019/09

In another paper [5], a lexicon-based backtracking approach has been employed over 301 test sentences for binary emotion classification. In this paper, they first classified the sentiment of the data and then the emotion. The dataset was mainly collected from Facebook status, news headlines, textbook, and direct speech. They claimed an accuracy of 77.16% using this approach.

A good study on emotion tagging has been done on paper [6]. In this paper, authors aimed at doing manual annotations of sentence-level text from web-based Bengali blog corpus and observed the classification results on the corpus they annotated. With 1200 training instances, The Conditional Random Fields (CRF) classifier gave them an average accuracy score of 58.7% and Support Vector Machine (SVM) managed to get them to 70.4%.

A computational approach of analyzing and tracking emotions is studied in the paper [7]. This paper focuses on the identification of the emotional expressions at word, phrase, sentence, and document level along with the emotion holders and events. Emotions has been tracked on the basis of subject or event. They observed a micro F-score of 0.63 on 200 test sentences that are collected from Bangla news and blogs for sentential emotion tagging.

On a case study for Bengali [8], authors developed a blog-based emotion analysis system. The blog posts are collected from Bengali web blog archive. They considered 1100 sentences and used a morphological analyzer for identifying lexical keywords. The average evaluation results they got for precision, recall, and F1-score was 0.59, 0.64 and 0.62 respectively. And for the morphological based system, they achieved an F1-score of 0.65.

A multilabel sentiment and emotion detecting approach has been proposed in [9]. Authors have considered a dataset containing Bangla, English and Romanized Bangla comments on different YouTube videos. They proposed a deep learning-based approach that classifies Bangla comments with a three-class and five class sentiment labels. They built models to extract emotions as well. Results showed 65.97% and 54.24% accuracy for three class and five class sentiment labels. Also, they extracted emotions with an accuracy of 59.23%.

In one of the most recent works [10] about fine-grained Bangla emotion classification, authors compared the results for five different classical machine learning techniques. They introduced a dataset having six different emotion categories and showed that a non-linear SVM with RBF kernel achieves an average accuracy of 52.98% and an overall F1 macro score of 0.33. One remarkable contribution of this paper is the exploration of manifold preprocessing and feature selection techniques. We are also using the same dataset produced by them but only considering three classes because of the highly imbalanced nature of this dataset.

III. METHODOLOGY

Given a set of comments and emotion labels (e.g. happy, sad, anger) for each of those comments, our main objective was to classify the comments with the appropriate emotion label using a supervised machine learning algorithm called Naïve Bayes classifier. We have gone through a number of pre-processing steps before classifying the data to remove any unnecessary information. Variants of feature selection techniques were also examined to improve the classification performance. Each of these techniques was important since these can affect the overall result of the supervised approach that we took. Here in this section, we provided an overview of our approaches. Figure 1 shows the detailed architecture of the proposed method.

Our system works in two phases: Training phase and Test phase. At the very beginning, the dataset was divided into training and test data. The training set is then processed with several pre-processing steps followed by various feature selection techniques before feeding them to the classifier. The test data goes through this same process afterward and predicts probable emotion labels for each document. The predefined emotion labels are then compared with the predicted ones to evaluate the efficiency of the system.

The proposed method can be divided into four sequential phases:

1. Dataset Preparation
2. Pre-processing
3. Feature Selection and Extraction
4. Classification

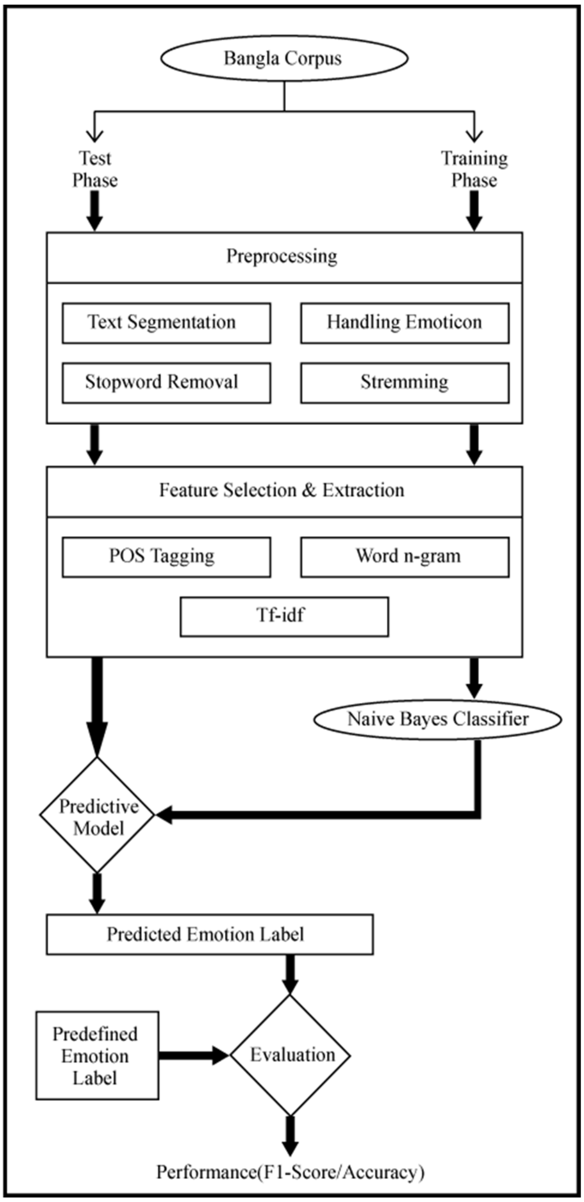

Fig 1: Architecture of the proposed method

A. Dataset Preparation

The dataset we used for this work contains a large number of user comments from different Facebook groups and some public posts of popular bloggers. The comments were collected based on different socio-political issues. We adopted the same corpus developed by the authors of this paper [10]. The dataset was annotated by the authors themselves based on the presence of words in addition to phrases corresponding to emotional content as well as the overall emotion lying in that comment. Among six different categories, we only considered 4200 comments of three emotion classes - happy, sad and angry. Owing to the fact that the dataset was too much imbalanced and the number of annotations for the other classes were quite a few as compared to the above three classes. This in terms affected their performance (to 52.98%) because most of the misclassification was due to the reason of training imbalanced dataset. We split the above dataset where 3780 of the comments were used for training and 420 comments as test data. Table I provides the class distribution of the selected dataset.

TABLE I. CLASS DISTRIBUTION IN THE DATASET

Label	***Training Set***	***Test Set***
Happy	1582	230
Sad	1062	104
Angry	1136	86
Total	3780	420

B. Pre-processing

The comments in the dataset contained a lot of useless and duplicate data such as stop words, punctuations, digits, and symbols. In order to simplify the further steps, we processed the data and cleaned them as follows:

a) *Text Segmentation:* At first any kind of extra dot, comma, hyphen, and other symbols were deleted. Punctuations and digits were removed too. Then the sentences were tokenized into separate words. We have used the native python string split function to tokenize the words of sentences.

b) *Handling Emoticon:* At this point, we removed all the emoticons from the comments since we aimed to consider text data only. A string of punctuations and emoticons was defined to strip out every single one of them from the data.

c) *Stop Word Removal:* We removed the stop words from each data. Since stop words are actually the most frequently repeated words. We filtered stop words (such as হয়, অথচ, অথবা, এবং) and removed them. We have considered the Bangla stop word list from an open source project[3].

d) *Stemming:* A word may vary in different forms. We stemmed each token into its root word to make it easier to process the data. Stemming Bangla words is quite difficult since it has a huge number of inflected words. Here, we made lists of prefix, suffix, nouns, verbs and article inflections. And then trimmed the left or right part of the word to extract the stems. For example, these three words করছেন, করবেন, করছে would be converted into the root word 'কর'.

C. Feature Selection and Extraction

Feature selection and extraction is the most important step to detect emotion because it affects the overall result of the work. A good feature selection results in a good prediction. So, selecting features properly to enhance the classification is very important. After the completion of the pre-processing phase, we applied several features to evaluate our processed data. We combined different techniques to observe the best possible result.

a) *POS Tagging:* POS tagger labels a word based on its grammatical category. It assigns each of the words in a document to its corresponding parts of speech. Different approaches can be used for POS tagging. We have used a Hidden Markov Model (HMM) based POS tagger which is supervised. The POS tagger we used can tag words with 32 tags of parts-of-speech and its subclasses[4]. We have used the same POS tagger used in the paper [10] which has a claimed accuracy of 75% over the POS tagged dataset they used. In our experiment we have only considered 3 versions of tagging, one with only *JJ ('adjective')*, another with five tags - *'JJ', 'CX', 'VM', 'NP' and 'AMN'*, and finally with 'all tags' for the purpose of comparison. The reason behind choosing these specific tags is that, most of the emotion related words falls under these parts-of speech categories. We looked upon other combinations too but those actually did not change the results too much.

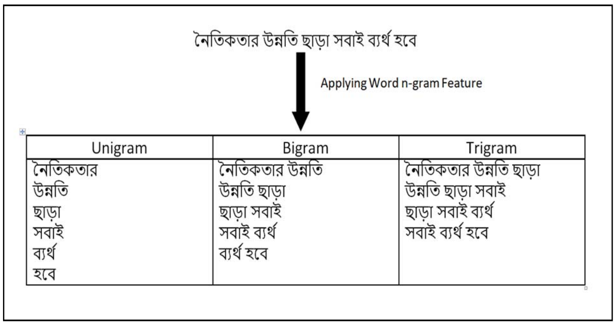

Fig 2: Example of n-grams as a feature.

b) *Word n-grams:* n-grams feature is considered to be very useful for classifying texts. Basically, this is a combination of n subsequent words or characters. In our work have used word n-grams. Here, we have observed the performance for unigram, bi-gram, and tri-gram to get the best model. We combined the n-grams feature with other features as well to get the overall scenario. Bi-gram provides a relatively better result than unigram and tri-gram in our work.

[3] https://github.com/stopwords-iso/stopwords-bn

[4] https://github.com/shaoncsecu/Bangla_POS_Tagger

Therefore, we used bi-grams for further evaluation. For our experimentation with features, we used scikit-learn [11]. Figure 2 shows an example of unigram, bigram, and trigram in Bangla text.

c) *Tf-idf Vectorizer:* Tf-idf simply corresponds to *term frequency times inverse document frequency.* One good input representation in NLP task is the tf-idf vectorizer which weights the occurrence of a word in a document instead of taking only raw counts. In our work, we combined both tf-idf and n-gram from scikit-learn [11]. The *term frequency* counts how many times a particular word appears in a given document. Whereas, *inverse document frequency* accounts for all the documents which have that word in it. The formula to calculate tf-idf is given below:

$$TF(w) = \frac{No.of\ times\ word\ w\ appears\ in\ a\ doc}{Total\ no.of\ words\ in\ the\ doc} \quad (1)$$

$$IDF(w) = \log_e\left(\frac{Total\ no.of\ documents}{No.of\ documents\ with\ word\ w}\right) \quad (2)$$

$$TF\text{-}IDF(w) = TF(w) * IDF(w) \quad (3)$$

D. Classification

The classifier uses the representation of the data that goes through all the pre-processing and feature selection steps. For classification, we have used a Multinomial Naïve Bayes classifier to predict the emotions from the text. Naïve Bayes is a probabilistic classifier which relies on the Bayesian theorem [12] to build a predictive classification model based on every pair of features. We have applied it in our work using scikit-learn [11]. It is a widely used library for text classification in python. In our implementation, we used the Multinomial version of the NB which is defined by the function in scikit-learn as `MultinomialNB()`. It uses the `fit(trainDoc, trainClass)` method to train the classifier. In our case the training data contains the comments and the training class are the corresponding emotion labels. NB algorithm uses the following Bayes rule to calculate the class probability given a document which is also termed as posterior probability. The class that has the maximum probability among all the classes will be selected as the most probable class for that document. It determines the probability of each word or word n-grams with respect to the classes and uses the chain rule to produce the full probability for a document given a class.

$$P(A|B) = \frac{P(B|A)\times P(A)}{P(B)} \quad (4)$$

A classification model based on Multinomial Naïve Bayes classifier has been designed in this work to classify Bangla Language texts into three different classes namely happy, sad and angry. The model was first trained with the training data and then the test labels were classified with this model. The predicted labels of the test data were then compared to the gold emotion labels to evaluate the performance of the classification model.

IV. Evaluation And Result

In order to evaluate the performance of our proposed method, we have considered precision, recall, and F1-score for each emotion class and the average accuracy.

We have experimented with different combinations of pre-processing and features to find out which combination produces the best score. Table II shows the performance of each of these models in different combinations.

TABLE II. Classification Results Based on Different Features.

Feature Combination	*Accuracy*
emoticon removal + tf-idf + stemmer	0.775
Stopword and emoticon removal + stemmer + tf-idf + POS tagger	0.773
Stopword and emoticon removal + stemmer + tf-idf + unigram + POS tagger	0.776
Stopword and emoticon removal + stemmer + tf-idf + bigram + POS tagger	**0.786**
Stopword and emoticon removal + stemmer + tf-idf + trigram + POS tagger	0.774

Based on all the experiments given, we choose the model with the best features. Our best model uses bigram based tf-idf with POS features and both stopword and emoticon processor. With this combination, we were able to gain an overall accuracy score of 78.6%. For this same combination, we also evaluated the result by using Support Vector Machine (SVM) for classification. The overall score for SVM was not sufficient enough compared to MNB. Table III shows the result of both classifiers for the best combination of features.

TABLE III. Classification Results BasED ON CLASSIFIER.

Classifier	*Accuracy*
MNB	0.786
SVM	0.716

Figure 3 shows the confusion matrix of this classification model on the test set.

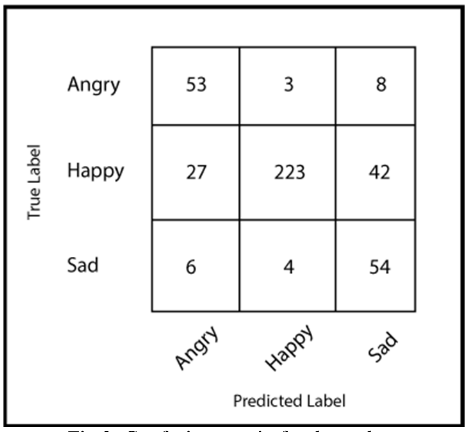

Fig 3: Confusion matrix for three class

Table IV shows the detailed evaluation scores for our best model. We can see that the model's prediction for the happy class was really good (F1 of *0.843*) while for sad categories, our model performed poorly. This is due to the fact that our training dataset was imbalanced and the sad class had the lowest number of training examples. However, this, in essence, represents the real-world scenario making our model more generalized.

TABLE IV. DETAILED EVALUATION USING BEST MODEL.

Emotion Class	*Precision*	*Recall*	*Macro F1-Score*	*Accuracy*
Happy	0.764	0.968	0.843	0.786
Sad	0.843	0.519	0.643	
Angry	0.828	0.616	0.707	

We also compared our best model with some of the existing works that we have mentioned in the literature review section. Table V shows the comparison with related works in the aspect of techniques and results. While evaluating, we have seen that the size of dataset, features and classifier used for the classification purpose greatly affects the overall result of the work.

TABLE V. COMPARISON WITH RELEVENT STATE OF ART WORKS

Paper	*Approach/Algorithm*	*Result (Accuracy/ F1-score)*
M. Mahmudun, 2016	• Tf-Idf on training data • Frequency of Important data	83% (A)
K. Sarkar, 2018	• NB	48.5% (A)
T. Rabeya, 2017	• Lexicon Approach • Backtracking Technique	77.16% (A)
S. Bandyopadhyay, 2010	• Data Annotation • Conditional Random Field (CRF) • SVM	CRF-58.7% SVM-70.4% (A)
D. Das, 2011	• Emotion, Holder and Topic • Sense-based affect estimation • SVM/CRF	63.26% (F1)
S. Roy, 2012	• Lexical word level keyword spotting • Ordering of timestamps	65.3% (F1)
N. I. Tripto, 2018	• LSTM/CNN • SVM/NB	(LSTM) 65.97% 54.24% 59.23% (A)
M.A. Rahman, 2019	• NB/SVM/KNN/K-means clustering/ Decision tree	(SVM) 52.98% (A)
Our best model	• **NB**	**78.6%** **(A)**

V. CONCLUSION

In this paper, we have tried to detect emotions from Bangla text. We have used a dataset that is comprised of a large number of user comments from Facebook posts. We processed the data to remove any kind of unnecessary information noise from it and to make the classification easier. We applied a variety of features like n-grams, POS tagger, and tf-idf to enhance the efficiency of the classifier. We used a Multinomial Naïve Bayes classifier to classify the data and compared the predicted label with the gold label. Our final model was able to classify the test data with an overall accuracy of 78.6%.

While doing this work, we faced some problems with the dataset since it was imbalanced. The original dataset consists of six emotion class but we only considered three classes that had a relatively sufficient number of data. We believe, there is some inconsistency with the annotations and it makes the model disagree more. Also, the overall evaluation would have been better if we had more data. Moreover, Bangla is a morphologically rich language so we faced some difficulties while working with some of the features. In a future study, we also would like to handle negations for better performance.

REFERENCES

[1] D. G. Myers, “Theories of emotion,” *Psychol. Seventh Ed. New York, NY Worth Publ.*, vol. 500, 2004.

[2] P. Ekman, “An argument for basic emotions,” *Cogn. Emot.*, vol. 6, no. 3–4, pp. 169–200, 1992.

[3] M. Mahmudun, M. T. Altaf, and S. Ismail, “Detecting Sentiment from Bangla Text using Machine Learning Technique and Feature Analysis,” *Int. J. Comput. Appl.*, vol. 975, p. 8887.

[4] K. Sarkar, “Using Character N-gram Features and Multinomial Naive Bayes for Sentiment Polarity Detection in Bengali Tweets,” in *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, 2018, pp. 1–4.

[5] T. Rabeya, S. Ferdous, H. S. Ali, and N. R. Chakraborty, “A survey on emotion detection: A lexicon based backtracking approach for detecting emotion from Bengali text,” in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, 2017, pp. 1–7.

[6] D. Das and S. Bandyopadhyay, “Labeling emotion in Bengali blog corpus--a fine grained tagging at sentence level,” in *Proceedings of the Eighth Workshop on Asian Language Resouces*, 2010, pp. 47–55.

[7] 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*, 2011, pp. 343–348.

[8] D. Das, S. Roy, and S. Bandyopadhyay, “Emotion tracking on blogs-a case study for bengali,” in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2012, pp. 447–456.

[9] N. I. Tripto and M. E. Ali, “Detecting Multilabel Sentiment and Emotions from Bangla YouTube Comments,” in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018, pp. 1–6.

[10] M. Rahman, M. Seddiqui, and others, “Comparison of Classical Machine Learning Approaches on Bangla Textual Emotion Analysis,” *arXiv Prepr. arXiv1907.07826*, 2019.

[11] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[12] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.