Tokenization and Implementation of Stop-Word Function to Analyze Sentiment of Bengali Language

Pritam Chowdhury
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
ppcctulan@gmail.com

Fahmida Hassan
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
fahmidahassan996@gmail.com

Arthy Anjun
Department of Computer Science and Engineering
East West University
Dhaka, Bangladesh
arthy.anjun@gmail.com

Abstract--Most of the research work on sentiment analysis are performed on English Languages. To remove unnecessary and noisy words from English language dataset many applications are also executed and even there is a build in Stop-words function which can be used to preprocess the dataset. Many works also being performed to analyze the sentiment using Bengali language. But we all know that Bengali Language datasets are full of unnecessary and noisy words which are not relevant to express sentiments. But there is no works which can withdraw the additional words from Bengali sentences although which have not directly bearing on sentiment analysis. In this paper, we are trying to implement a function called stop-words to remove the unnecessary and noisy words from our data set and apply sentiment analysis to analyze the writer's sentiment using Bengali Sentences to identify either the sentence was positive or negative. For classification we use Random Forest, Support Vector Machine and MLP Neural Network to analyze the comparative accuracy differences before and after implement of Stop-word function.

Keywords--Stop-words function, Bengali Datasets, Random Forest, Support Vector Machine, MLP Neural Network, comparative, analyze, differences, implement

I. Introduction

In this modern era by technological advancement everything is upgrading. Sentiment analysis method is one of them. This method is already used in different sectors to analyze and carry out the author's emotion in many languages. Now, we are trying to apply this method to analyze the writer's sentiment using Bengali Sentences.

One of the most important parts of the Bangladesh government's Vision 2021 is "Digital Bangladesh". In this information based world the use of Internet is raising rapidly over the country and also uses different online platforms which are increasing among the common people in every spare of their daily lives. This inspired us to design datasets by analyzing the Bengali people's emotions and to extract their sentiments in different aspects. In this paper, we have constructed a dataset which contains some Bengali lines of common people's opinion.

In this work, the user will use a Bengali sentence as an input. The input size is not pre-defined here. the user can write the sentence as long as he/she wants to. After having input a function called stop-word implemented to pull out the additional and redundant words from dataset. After that, the system will analyze the sentence and predict the attitude of authors. Here we classify the writer's emotion in two motion, positive and negative. As we work with Bengali Language Some Example of Bengali sentences input and output are described below.

- If the user gives an input like **তাকে নিলে তার ভাল হবে** then the system will analyze this sentence and give a prediction that the writer is expressing a sentence in negative mood. But the sentence can be predicted positively by the system only based on **নিলে তার ভাল হবে**. So implementing Stop-word function we remove the word **তাকে** and make the sentence more efficient and simple to train the machine and predict it.
- Again, if the user gives an input like তিনি কোনো কাজ জানেন না then after analyzing this sentence it will give a prediction that the writer is expressing a sentence in negative mood. In same way this sentence can be predict based on only কোনো কাজ জানেন না as a negative

sentence .So by implementing the function the word তিনি has been eliminated.

II. PREVIOUS WORK

For information for retrieval and text mining, preprocessing is a very important task and vital step. A research work has been done with the Turkish language. Analyze the effect of preprocessing on text classification is the main purpose of that work. Two large datasets from Turkish newspapers using a crawler have been used on this work. A detailed analysis of preprocessing methods such as stemming, stop-word filtering and word weighting for Turkish text classification was performed in this work[1]. Another work has been done with English Language and found that accuracy raised from unprocessed dataset to stop-words removed dataset for Traditional Sentiment Classifiers [2]. One of the most significant preprocessing procedures is the elimination of functional words, also known as stop-words. Text processing tasks affects the performance very deeply. So it is important to remove stop-word from all text processing tasks. So there is proposal of a Stop-Word removal algorithm for Hindi Language which is using the concept of a Deterministic Finite Automata (DFA).[3] Using CNN as only a classifier gives lower accuracy than the accuracy has been found from CNN which can be used to extract trainable features for the SVM classifier. Two different method named feature level fusion and decision level fusion can also be used for better result.[4]

III. METHODOLOGY

Our system architecture, outlining the whole process, is shown in figure 1. All our experiments are performed using NLTK Python Toolkit1.

A. Dataset

In this dataset there are 7459 comments or line of gathered from different Bengali peoples which is used commonly when they communicate. The dataset is constructed through survey and it is found on (https://gist.github.com/Tulan01/8862aab93d101c0f3968056c664b01a0) .
The lines or comments are of different lengths and each line contains approximate 3-30 Bengali words. People generally used Bengali Language to express their feelings and emotions about any news, incidents or occurrences. We also observed that 5 – 10% of the time they use English language for the purpose of their communications. We also observed that they use English alphabets to write Bangla sentences. We did not consider that type of data in our dataset. In addition, we also omitted those data which contains only emotions and no other texts or words.

The following table shows some classifications module of our dataset. We have subtracted our dataset into 2 types of polarities that is positive and negative

Data	Polarity
তিনি সবার ভাল চান	Positive
তিনি আমাকে মারতে চান	Negative
আমাকে কেউ ভালবাসে না	Negative
তাকে সবাই পছন্দ করে	Positive
আমি তাকে ভয় পাই	Negative
তার পাওয়া লাগবে ভয়	Positive

Table 1: Determination of polarity

B. Learning and Testing Module

For learning and testing the data set, k fold method is used. We used a certain percentage of data for learning the machine about our desired six types of classes. And a certain amount of data is used for testing. The Percentage of test data is 40%, it indicates that 2984 number of rows have used for testing purpose. On the Contrary 4475 number of rows are used for learning the machine, which cover approximately sixty percent of our data set.

In the training phase, extracted feature sets were trained by the popular supervised machine learning algorithms, we trained our models by setting up multi-label output. We used machine learning algorithms. The following algorithms are used for implementation.

- Neural Network,
- Random Forest
- Decision Tree

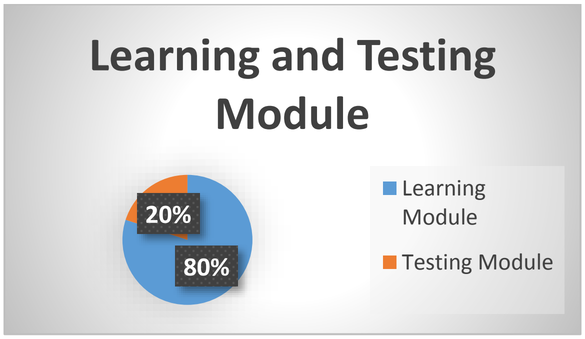

Figure 1: Learning and testing data module

C. Preprocessing and Feature Extraction

In the preprocessing phase, we have applied traditional preprocessing steps for the evaluation. Firstly, we have created a function called stop-words, corresponding to our dataset. We tokenize each Bangla word from our dataset. After that, stop words have been removed from each of the Bengali lines. Finally, we reassemble the tokenize words into a normal sentence. We have created a feature matrix for which each review was represented by a vector of that vocabulary. Term frequency–inverse document frequency (TF–IDF) is used for calculating the features. TF-IDF means Terms frequency and inverse term frequency. TF-IDF algorithm is made of 2 algorithms multiplied together. Term frequency (TF) is how often a word appears in a document, divided by how many words there are. We have used Label-Encoder for classify the dataset into classes. It converts classes into numerical values.[5**]** Sentiment classification in any dataset is usually affected by the noisy nature (abbreviations, irregular forms) of data. A popular procedure to reduce the noise of textual data is to eliminate stop-words by using pre-compiled Stop-word lists. However, in the few years in the context of sentiment analysis the effectiveness of eliminating of Stop-words has been debated.[3]

In this work, we analyze the sentiment of Bengali dataset. Generally, the Bengali dataset is full of unnecessary and irregular words. So, we made a stop words list as per our dataset and apply it in our dataset.

Stop-Words list is shown below,

['অনেকে','অনেকেই','অথবা','অথচ','অর্থাত','অন্য','আজ','আপনার','আপনি','আবার','আমরা','আমাকে','আমাদের','আমার','আমি','আর','আই','আগামী','অবধি','আদ্যভাগে','এই','একই','একে','একটি','এটি','এটা','এটাই','এবং','একবার','এবার','এদের','এঁদের','এমন','এমনকী','এল','এর','এরা','এঁরা','এস','এত','এতে','এসে','একে','এ','ঐ','ই','ইহা','ইত্যাদি','উনি','উপর','উপরে','ও','ওই','ওর','ওরা','ওঁর','ওঁরা','ওকে','ওদের','ওঁদের','ওখানে','কত','কবে','সেখানে','সে','স্বয়ং','কি','কী','তিনি','তিনঐ','তিনিও','তাঁদের','তাঁহারা','তাঁরা','তাঁর','তাঁকে','তাই','তাকে','তাহার','তাদের','তারা','তারৈ','তার','তিনি','তুমি','তোমার','তথা','দু','দুটি','দুটো','নিজে','নিজেই','নিজের','নিজেদের','প্রভৃতি','বার','বা','ভাবে','ভাবেই','মধ্যভাগে','যাকে','যার','যারা','যাঁর','যাঁরা','যাদের','যিনি','যে','সবার','সহ','সুতরাং','সহিত','সেই','সেটা','সেটি','সেটাই','সেটাও','সম্প্রতি','সেখান','হিসাবে','জন','জনকে','জনের','জন্যওজে','জে','তুলে','মোট','টি']

Below we represent some example of our dataset .The Data set we used before removable of Stop-Words:

তিনি কাজ জানেন
তিনি কাজ জানেন না
তিনি কোনো কাজ জানেন না
তিনি কাজ করতে চান
তিনি কাজ করতে ভয় পান
তিনি কাজ করতে অনেক ভালবাসেন

After removable of stop-words the dataset become more efficient and short The words from the listed stop-word function have been removed and the set looks like below:

[কাজ, জানেন]
[কাজ, জানেন, না]
[কোনো, কাজ, জানেন, না]
[কাজ, করতে, চান]
[কাজ, করতে, ভয়, পান]
[কাজ, করতে, অনেক, ভালবাসেন]

D. Data Analyzing and Implementation of Algorithms

To analyze the data and extracting the features we follow some criteria. At the beginning we divide the data into two classes manually and the data is created by us. Then we extract the data in a csv file for input, at the next step using label encoder the text data is formed into numerical data. Then using term frequency–inverse document frequency (TFIDF) we vectorize the data, so that the machine can learn it. TFIDF is a numerical statistic that is intended to reflect how important a word is to a document in the collection. After that, we use some classifier algorithm to learn and predict the best output for the given dataset.

- Extracting data from dataset and separate the whole data into 2 section column wise. Data section means the Bengali sentences and the other is polarity section which means different classes.

- Convert the whole data section into tokens.
- Prepare a Stop-Word list as per our dataset and eliminate the words from the data for having more efficient data.
- Employing Label Encoder [6] and TF-IDF vectorization [5] on the training, testing and cross-validation set to convert them into corresponding binary value to feed into machine learning approaches as input.
- Splitting them into train (80%), test (20%) and cross-validation set.
- Further applying classification approaches Support Vector Machine, Neural Network, Random Forest, Decision tree, K-Nearest Centroid and Naïve Bayes for performance comparison purpose.
- Finally, performances of all the classification approaches have been analyzed, compared and demonstrated based on accuracy, specificity, memory usage, F-measures and comparison of generated have been illustrated.

E. *Classifier*

Our proposed work basically based on preprocessing of data set. We used Bengali dataset and remove the stop-words to make the training dataset more simple and efficient. So we used python toolkit built in classifiers. We use Neural Network, Random Forest and Decision Tree Classifiers. Here a question can be arise why we use these three classifier? Actually we choose these three classifiers randomly based on popularity. In the field of any kind of machine learning approach, Neural Network classifier is very popular to all. So we use that and the other two classifier are used for verifying the result.

Random forest is a brand term for a classifier that subsist of many decision trees and outputs the class that is created individually from all the decision trees. It does not randomize the training data but it randomize the algorithm we use. Random forest is form of many trees which are slightly different from each other. The strength of random forest is that, it is capable to deal with missing and unbalanced data and its runtime is very fast compare to others.[7]A neural network is also called artificially neural network. A neural network is an array of software and it is inspired by biological neurons. By applying these algorithms, we create artificially intelligent programs. Neural network is a method of machine learning. For solving problems a program can be changed as it learns.[8] In Decision Tree, the attribute for the root node in each level identification is the major challenge. This process is called attribute selection. There are two popular attribute selection measures, they are information gain and Gini index [12].

F. *Working Procedure Flow Chart*

The hole working procedure have been described through a flow chart below

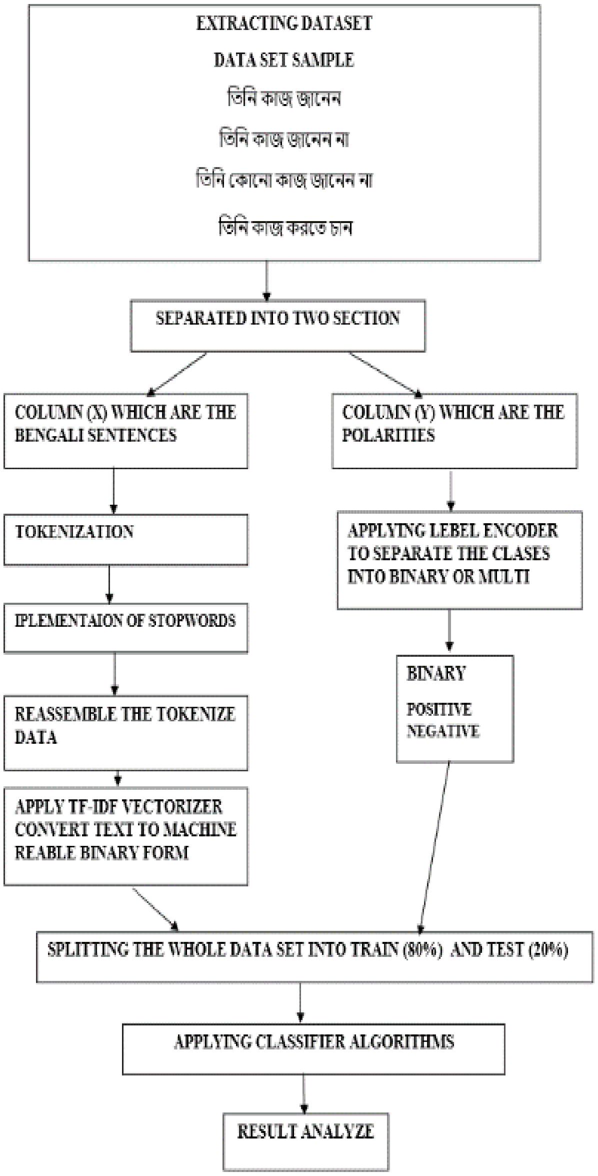
EXTRACTING DATASET
DATA SET SAMPLE
তিনি কাজ জানেন
তিনি কাজ জানেন না
তিনি কোনো কাজ জানেন না
তিনি কাজ করতে চান
SEPARATED INTO TWO SECTION
COLUMN (X) WHICH ARE THE BENGALI SENTENCES
COLUMN (Y) WHICH ARE THE POLARITIES
TOKENIZATION
APPLYING LEBEL ENCODER TO SEPARATE THE CLASES INTO BINARY OR MULTI
IPLEMENTAION OF STOPWORDS
BINARY
POSITIVE
NEGATIVE
REASSEMBLE THE TOKENIZE DATA
APPLY TF-IDF VECTORIZER CONVERT TEXT TO MACHINE REABLE BINARY FORM
SPLITTING THE WHOLE DATA SET INTO TRAIN (80%) AND TEST (20%)
APPLYING CLASSIFIER ALGORITHMS
RESULT ANALYZE

Figure 2: Working Procedure flow chart

IV. EXPERIMENTAL RESULT AND EVALUTION

In this section we have got the performance and result analysis from all the considered approaches that have been evaluated and analyzed using various performance measurements including accuracy with the help of precision, recall and F1 Score. Additionally training time as well as memory usage by the proposed approach along with all the considered approaches have been conducted. All these performance evaluation metrices have been concentrated in order to avoid overfitting problem and finding superior approach among all the approaches applied here in this work. Comparative studies of all the approaches are as follows:

After the training was completed, our proposed Bangla test dataset was executed on the trained model.

The result is shown in the following table and figure

Algorithms		Neural Network	Random Forest	Decision Tree
Before Removal Of Stop-words	Accuracy	69.97 %	70.71%	69.60%
	F1 Score	69.79%	70.71%	69.57%
	Recall Score	69.97%	70.71%	69.60%
	Precision Score	70.14%	70.64%	69.54%
After Removal Of Stop-words	Accuracy	71.91%	73.05%	70.37%
	F1 Score	71.64%	73.04%	70.36%
	Recall Score	71.91%	73.05%	70.37%
	Precision Score	72.50%	73.06%	70.50%

Table 2: Accuracy differences before and after removal of Stop-words

From the above table, it is clearly shown that, after removal of Stop-Words from the data set Accuracy, F1 Score, Recall score, Precision score are improved for all three algorithms. Almost 2% of accuracy increases after removal of Stop-Words. The highest accuracy is found using Random Forest Classifier in both section before and after using Stop-Words. Because Random Forests consists of multiple single trees each based on a Random sample of training data. Random Forest tree is fully grown and unpruned so the feature space is split into more and smaller region. In our work we mostly focus on improvement of accuracy after using Stop-Words. Below a graph describes only accuracy differences.

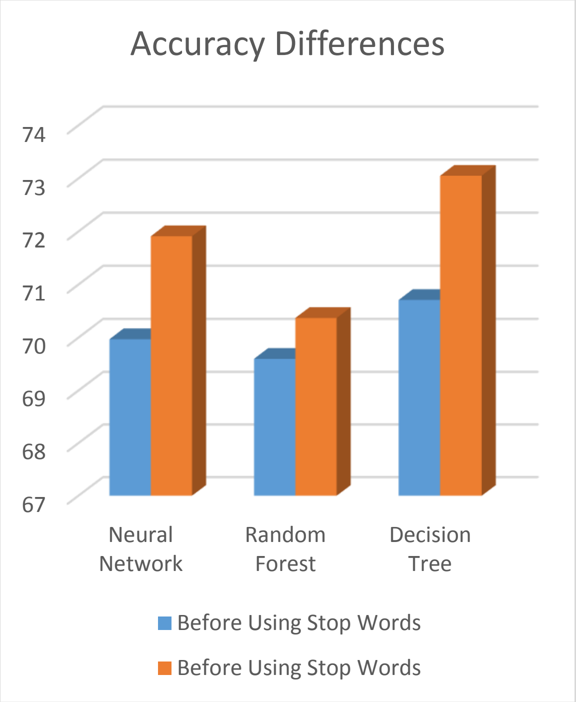

Figure 2: Accuracy difference before and after using Stop-Words

When we run built in randomly chosen three algorithms on our data set without removal of Stop-Words Neural Network gives 69.97% accuracy, Decision Tree gives 69.60 accuracy and Random Forest gives the highest accuracy which is 70.71%. After removal of Stop-Words Neural Network gives 71.91% accuracy, Decision Tree gives 70.37% accuracy and Random Forest gives the highest accuracy of 73.05%.

At the end we can ensure that using Stop-Words undoubtedly increases the accuracy rate and using this implementation, anyone can use any dataset for analyze the sentiment of Bengali dataset. But in some case exception may occur.

V. Conclusion and Future Work

It can be concluded that, Bengali language is one of the most prestigious and famous language in this world. It is the only language for which, martyrs give their valuable life. And the whole world celebrates the International Mother Language Day for this prestigious language. That’s why we choose this language and try to analysis the sentiment by using some different algorithms. In our work for binary classification we get good accuracy rate, but for six class classification we get moderate accuracy rate. In near future we will try to improve the accuracy and we will work with a large dataset than the present dataset. And we also have two different idea to work with Bengali language. We will make a module using embedded system, where a person will input his voice in Bengali language and after analyzing the Bengali sentence the module will reply automatically whether he is in happy/sad/angry/excited/scared/tender mood. As well as we have another idea to work with Bengali language, for instance we wish to work with Bengali newspaper headlines. In that work, the system will automatically detect the sarcasm of Bengali newspaper headlines using different approaches.

VI. References

[1]D. Torunoğlu, E. Çakirman, M. C. Ganiz, S. Akyokuş and M. Z. Gürbüz, "Analysis of preprocessing methods on classification of Turkish texts," *2011 International Symposium on Innovations in Intelligent Systems and Applications*, Istanbul, 2011, pp. 112-117. doi: 10.1109/INISTA.2011.5946084 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5946084&isnumber=5946042

[2]K. V. Ghag and K. Shah, "Comparative analysis of effect of stopwords removal on sentiment classification," *2015 International Conference on Computer, Communication and Control (IC4)*, Indore, 2015, pp. 1-6. doi: 10.1109/IC4.2015.7375527 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7375527&isnumber=7374772

[3] Oro.open.ac.uk. (2019). *On stopwords, filtering and data sparsity for sentiment analysis of Twitter - Open Research Online*. [online] Available at: http://oro.open.ac.uk/id/eprint/40666

[4] S. Poria, E. Cambria and A. Gelbukh, "Deep Convolutional Neural Network Textual Features and Multiple Kernrel Learning for Utterence-Level Multimodal Sentiment Analysis", *Conference on Empirical Methods in Natural Language Processing*, pp. 2539-2544, 2015.

[5] R. J, "Using TF-IDF to Determine Word Relevance in Document Queries", 2003.

[6] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment Analysis of Twitter Data."

[7] Medium. (2019). *Understanding Random Forest*. [online] Available at: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[8] Skymind. (2019). *A Beginner's Guide to Neural Networks and Deep Learning*. [online] Available at: https://skymind.ai/wiki/neural-network

[9] Available at: https://towardsdatascience.com/whats-so-naive-about-naive-bayes-58166a6a9eba

[10] Anon, (2019). [online] Available at: https://link.springer.com/chapter

[11] “Chapter 4: Decision Trees Algorithms – Deep Math Machine learning.ai – Medium.” [Online]. Available: https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1.

[12] “Decision Tree Introduction with example - GeeksforGeeks.” [Online]. Available: https://www.geeksforgeeks.org/decision-tree-introduction-example/.