ORIGINAL ARTICLE

Accurate Prediction of Bangla Text Article Categorization by Utilizing Novel Bangla Stemmer

Shafayat Bin Shabbir Mugdha[1], Zahid Hossain Khan[1], Mahtab Uddin[2,*], and Ashek Ahmed[2]

[1] *Department of Computer and Engineering, United International University, Dhaka-1212, Bangladesh*

[2] *Institute of Natural Sciences, United International University, Dhaka-1212, Bangladesh*

(**Received** 29 April 2024, **Accepted** 03 July 2024, **Published Online** 06 July 2024)

*Corresponding author: mahtab@ins.uiu.ac.bd

DOI: https://doi.org/10.5875/xbzrk013

Abstract: Text categorization involves assigning predefined category labels to an unlabeled document. With the exponential growth in the accessibility and availability of digital documents over the past decade, this field significantly attracted the scientific community that immensely demands rapid and accurate categorization of these documents. Relying on experts for manual classification is time-consuming and resource-intensive. Consequently, labeling unlabeled digital documents faster more accurately, and more efficiently is inescapable. One promising approach to addressing this demand is the use of machine learning algorithms. Training these algorithms on a large dataset of labeled texts lets them learn patterns and predicted unlabeled documents. This strategy might greatly expedite the categorizing process while retaining a substantial level of accuracy through leveraging artificial intelligence. These algorithms have also enhanced natural language processing techniques, making them more accurate at classifying unlabeled digital documents. In this study, we propose a novel machine-learning computational framework to address this challenge. Our framework incorporates a novel Bangla stemmer, which reduces words to their stems. We then employed TF-IDF for document vectorization, a statistical measure assessing word relevance for categorization purposes. Experimental results reveal that our framework significantly enhances prediction performance, achieving an impressive 95.3% prediction accuracy.

Keywords: Bangla Stemmer; Natural Language Processing; Text Categorization; Vectorization.

Introduction

The advancement of technology has sparked a renewed interest in addressing text categorization challenges. With an excess of online text documents, efficiently classifying them into different semantic categories becomes both advantageous and demanding. The task of automatically assigning documents to their relevant categories based on extracted features from textual content is a crucial aspect of machine learning, known as automatic text classification. This process holds significant applications, including facilitating faster and more efficient in- formation search and retrieval. Text classification using machine learning has proven useful in several fields, including email spam filtering, sentiment analysis, data mining, contextual search, and product review analysis. Improving the information's arrangement and accessibility is the main objective. Two primary perspectives for the classification problem are single-label and multi-label text classification. A document is assigned to only one category in single-label classification; however, in multi-label classification, a document may be a part of more than one class. This paper mainly focuses on multi-label text categorization classification, contributing to the understanding and development of methodologies tailored to this certain scenario.

Automated text classification has entered an advanced phase, leveraging various sophisticated machine learning techniques. The majority of the studies that have been conducted in this area are concentrated on specific languages such as English, Arabic, and Chinese translation. On the other hand, there has been a conspicuous lack of major progress towards the

development of Indian languages, particularly Bangla. This language, which is regarded as the sixth most generally spoken language in the world and has roughly 243 million people who speak it as their first language, is of great importance because it is one of the most widely spoken languages in both India and Bangladesh [1]. The language's prevalence underscores the need for advancements in text classification specific to Bangla, offering valuable insights for a substantial user base. Moreover, a Bangla-based system can empower users who are not proficient in Western languages to effectively utilize information technology. We find inspiration to contribute to this field because of this reality and the research gap that is currently present. In real-world data scenarios, disparate variations are frequently observed. To address such variations, a diverse set of machine learning techniques and classifiers have emerged to effectively handle this diversity.

In this study, we propose a cutting-edge computational framework based on machine learning to tackle the specified problem. At the beginning of our approach is the incorporation of an inventive Bangla stemmer, a method aimed at reducing words to their basic forms by handling affixes, prefixes, or word roots, also known as lemmas. Therefore, to convert our documents into vectors, we utilized a technique known as TF-IDF, which stands for term frequency- inverse document frequency. The TF-IDF is a statistical metric that is used for categorizing documents by determining the relevance of a word within a group of documents. Our experimental results highlight the efficacy of our framework in significantly enhancing predictive performance. Notably, it attains an outstanding 95.3% prediction accuracy, demonstrating the resilience of our methodology.

Research Over the Years

Through an extensive literature review, it becomes evident that considerable research effort has been dedicated to this domain, with a primary focus on English. However, a notable observation is the limited attention given to Indian languages, particularly Bangla, where only a few studies have been conducted. Nevertheless, to classify Bangla text, Mansur et al. [2] utilized an N-Gram feature extraction strategy to classify data from the Prothom Alo newspaper that was collected over one year. SVM, NB, DT (C4.5), and KNN were the four methodologies of supervised learning that Mandal and Sen analyzed and compared in their research [3]. Using these methods, they were able to sort a total of 22,218 tokens spread across one thousand labeled news web texts into five distinct categories. The results of their experiment showed that the SVM classifier was able to achieve a maximum accuracy of 89.14%. With the help of the SGD classifier, Kabir et al. [4] were able to successfully classify documents into nine different categories, achieving an accuracy rate of 93.85%. Islam et al. [5] went through the process of analyzing and contrasting several different classification algorithms for Bangla text documents. In an independent study [6], the researchers utilized the TFIDF weighting scheme in conjunction with the SVM classifier. As a result, they were able to achieve a classification ac- curacy of 92.57% for Bangla text documents that encompassed twelve different text categories. Dhar et al. [7] proposed a similarity-measure-based approach that utilized the TF-IDF weighting scheme to classify one thousand web text documents from five different domains. They were able to achieve a maxi- mum accuracy of 95.80% for the cosine similarity measure. They chose to use a reduction technique in another investigation [8], which involved taking into consideration the top forty percent of tokens extracted through the use of the TF-IDF method. Using the LIBLINEAR classifier, this experiment was con- ducted on 1960 documents. These documents came from five different domains. The results were encouraging.

Dataset

Over the years, several databases have been published that contain experimentally verified categorized documents. For our study, we have considered the BARD [9] dataset to obtain multi-category (economy, state, international, sports, and entertainment) articles that were curated from different Bangla news portals where each article is of different lengths. The dataset contains 75,000 articles (a subset of the original dataset containing 3,76,226 articles) and each category contains 15000 articles, thus removing the imbalance factor. Additionally, to avoid ambiguity, the sub-dataset does not contain any duplicate articles. We extracted a 10% sample from the original dataset and split it into two groups through a random assignment process. This was conducted to assess the framework's ability to apply to various scenarios and to verify that the performance claims were not overstated. The remaining 90% samples were allocated for training and validation. Upon the separation, the training and validation sets together contained a total of 52,500 instances while the independent set contained 22,500 instances. Of particular importance, the independent test set was carefully preserved as unseen data and was completely excluded from both the learning and parameter tuning stages.

Data Preprocessing

As introduced in the overview section, we have employed a novel Bangla stemmer, but to serve the data to our novel Bangla stemmer we have to go through some preprocessing steps [10, 11]. Figure 1 illustrates the overall structure of the preprocessing steps, revealing a comprehensive outline of these steps before we dive into their specifics in subsequent sections.

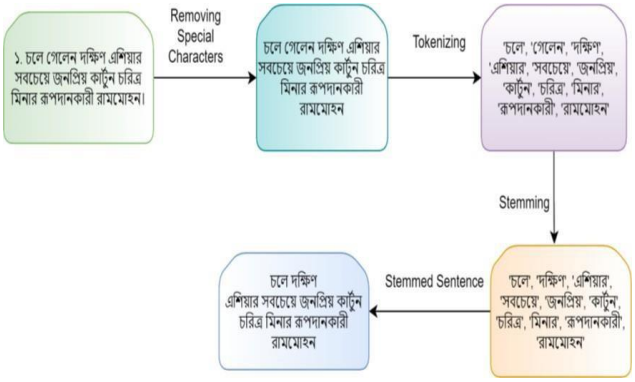

Figure 1. General architecture of preprocessing steps.

Data Cleaning

For machine learning algorithms, passing raw input data is not appropriate due to the naturalism of the format. Such as sentences, paragraphs, tweets, etc. In the raw input data, some words or characters hold minimal to no value for the algorithms to improve the prediction accuracy. Therefore, in this step, we have deducted all irrelevant words (stop-words), unique characters, and numeric values from the input data.

Tokenization

Tokenization is the fundamental step to proceed with any Natural Language. Tokenization involves breaking down a substantial amount of text into smaller units, referred to as tokens. These tokens play a crucial role in pattern identification and serve as a foundational step for both stemming and lemmatization. Additionally, tokenization proves valuable in the substitution of sensitive data elements with non-sensitive counterparts. Over the years, different tokenization techniques have been used for text classification or categorization [12]. However, in this study, we have used word tokenization [13] which converts a text into a list of words that helps us to identify and work with relevant and important words in later processes.

Removing Stopwords

Following the completion of tokenization, we eliminate Stopwords from the word list. Stopwords are ubiquitous words in the text that typically do not significantly contribute to sentence meaning. They carry minimal importance for both information retrieval and enhancing prediction accuracy. Omitting Stopwords is a safe practice that does not compromise the essence of the sentence. This removal process has the potential to enhance performance by retaining fewer yet more meaningful tokens. Consequently, it has the possibility of elevating classification accuracy.

Stemming

Stopwords are not the only words that hold little information about the text. A single word can be of multiple forms. For example, the word “eat” can be “ate”, “eaten”, or “eating”, etc. based on the sentence’s context. So, working with multiple forms of a single word is irrelevant. This is where we introduce stemming. Stemming is a rule-based approach because it slices the inflected words from prefixes or suffixes as per the need using a set of commonly underused prefixes and suffixes. This helps the framework to work with only relevant words that hold discriminative information which can improve the prediction capability of the framework [14].

After the preprocessing steps are complete, the list of words is again merged into a string for additional processing.

Vectorization

In this section, we push the string value through the vectorization process where each word or phrase from the dictionary is mapped to a corresponding vector of real numbers that accustomed information of word predictions, and word similarities.

Several types of vectorizers have surfaced over the years [15]. Some of the most popular vectorization techniques are Binary Term Frequency, Bag of Words, Normalized Term Frequency, TF-IDF, Word2Vec, Countvectorizer, Hash Vectorizer, etc. In this study, we have considered TF-IDF Vectorizer also known as Term Frequency–Inverse Document Frequency. A commonly employed technique is to convert textual data into numerical representation to assess the significance of a specific word in a document [16]. One of the most significant features of IDF is its ability to influence the frequency of terms, prioritizing the less common ones. For instance, when considering Term Frequency (TF), common words like “the” and “by then” frequently appear in the content, and TF will determine the frequency of their occurrence. Conversely, IDF quantifies the influence of these terms.

Feature Selection

When the quantity of features increases while the number of training instances remains constant, any classification model encounters the challenge of over-fitting. This phenomenon reduces the model's ability to generalize effectively. Furthermore, when features exhibit high correlation, the drawbacks associated with increased dimensionality outweigh the benefits of additional information. In our study, after vectorizing the data the number of features increases drastically. After vectorization, the total number of generated features is 3,84,620. Thus, the chance of over-fitting in the model increases exponentially. To achieve this objective, we have implemented feature selection using Random-Forest (RF), an ensemble model used extensively for classification tasks, which is capable of offering a ranking of the features based on their contribution in distinguishing instances belonging to different classes. After going through several thresholds, we have selected 300 as a threshold for the number of important features to be selected. We have selected only 0.078% of the generated features. This helps to reduce the over-fitting issue dramatically. Moreover, the framework only works with features having relevant and discriminative information.

Results and Discussion

Given that the main objective of this research is to introduce a novel stemmer mechanism and demonstrate its effectiveness in categorizing multi-class articles, we have not dedicated efforts to crafting a problem-specific model architecture. Instead, we opted for the utilization of LR [17], a commonly employed classifier. LR operates atop our optimized feature vector, learning the coefficients for a linear decision boundary within the feature space we derived. Along with LR, we added two additional machine learning classifiers to our study: Random Forest (RF) [18] and Multinomial Naive Bayes (MNB) [19]. The evaluation of our proposed framework's predictive performance involved the analysis of six distinct metrics: Accuracy, Precision, Recall (Sensitivity), Specificity, F1-score, and Matthew's correlation coefficient (MCC). Table 1 and Table 2 show the results for these metrics, which were found using 10-fold cross-validation and an independent set.

Table 1. Performance evaluation of LR, MNB, and RF using 10-fold cross-validation.

Algorithms	*Acc.*	*Pre.*	*Rec.*	*Spec.*	*F1*	*MCC*
MNB	90.8	90.9	90.8	97.7	90.8	88.5
LR	92.7	92.8	92.7	98.9	93.7	91.6
RF	91.8	90.3	91.1	97.6	92.5	89.8

Table 2. Performance evaluation of LR, MNB, and RF using independent test set.

Algorithms	*Acc.*	*Pre.*	*Rec.*	*Spec.*	*F1*	*MCC*
MNB	93.4	93.4	93.4	98.3	93.4	91.8
LR	95.3	95.3	95.3	98.8	95.3	94.1
RF	93.6	93.7	93.6	98.4	93.6	92.1

The performance evaluation in Table 1 shows that LR does a good job, with scores of 92.7% for accuracy, 92.8% for precision, 92.7% for sensitivity, 98.9% for specificity, 93.7% for accuracy, and 91.6% for MCC. LR consistently outperforms the other classifiers across all metrics during 10-fold cross-validation. In Table 2, LR exhibits superior performance compared to RF and MNB in the independent set, aligning with the trends observed in Table 1. The congruence between the results presented in Tables 1 and 2 reinforces the wide applicability of our framework. As a result, we believe our proposed framework has the potential to be an accurate and efficient approach to categorizing multi-class Bangla text articles.

All the experiments that we have conducted in this study are done using Python 3.8 and the Scikit-learn library.

Statistical Analysis

We performed a textual statistical analysis on the articles within the BARD dataset, and the findings are presented in Figure 2 and Figure 3 over the 3,76,226 data. Figure 2 showcases the frequency distributions of the top 20 most prevalent words for each of the five categories. Interestingly, the analysis reveals that the most frequent words across all categories are quite similar, indicating that these common words do not significantly aid in the categorization of the articles.

Consequently, we opted to eliminate approximately 25 of the most frequent words from all articles and conducted a subsequent statistical analysis on the filtered dataset, as presented in Figure 3. The filtered frequency distribution now highlights distinct word distributions for each category, suggesting that these unique words may play a role in contributing to the categorization of the articles. Figure 4 and Figure 5 show the visualization of both training and validation accuracy.

The model developed in this work has been deployed using the FLASK framework, and the outcome is depicted in Figure 6. From the figurative demonstration, it is observed that input texts can be efficiently recognized through the developed model.

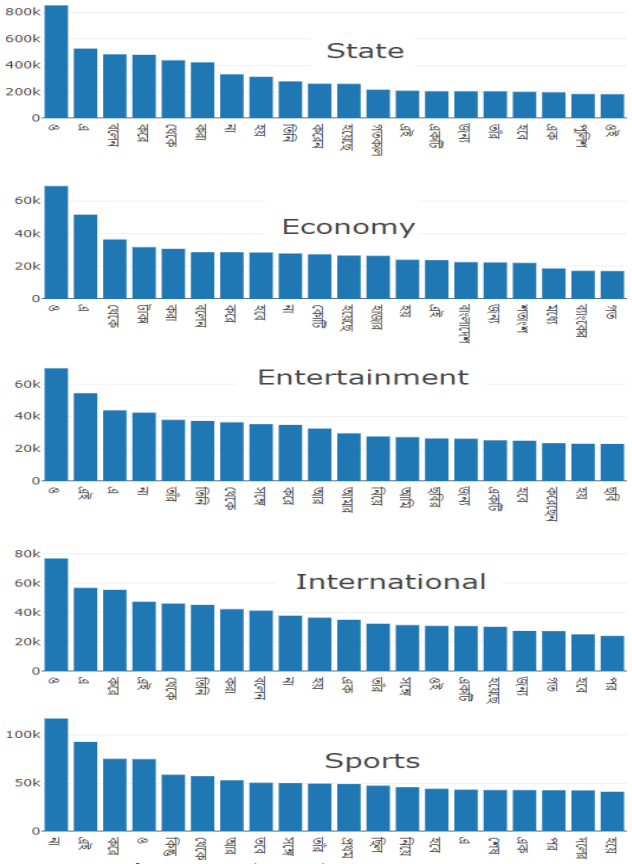

Figure 2. Most frequent words in each category.

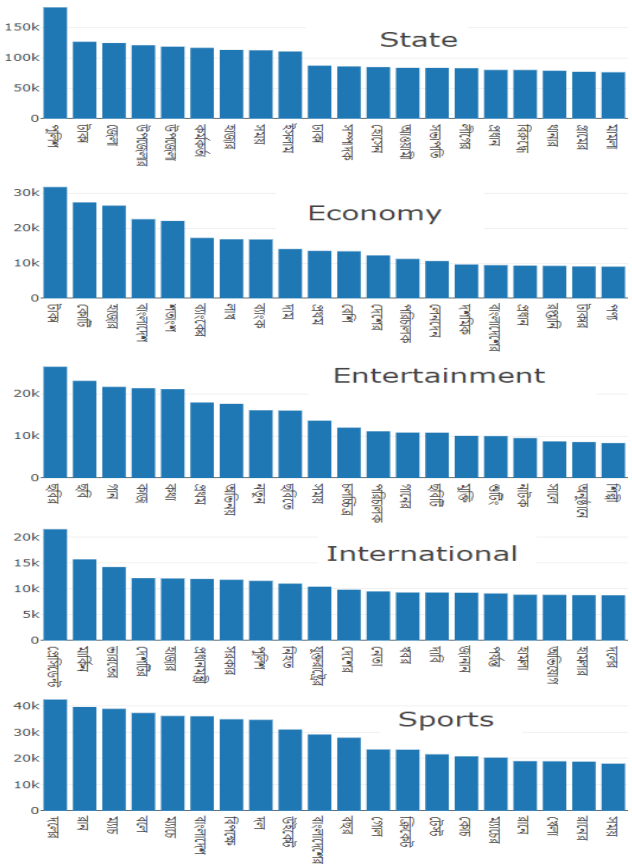

Figure 3. Most frequent words in each category after removing stop words.

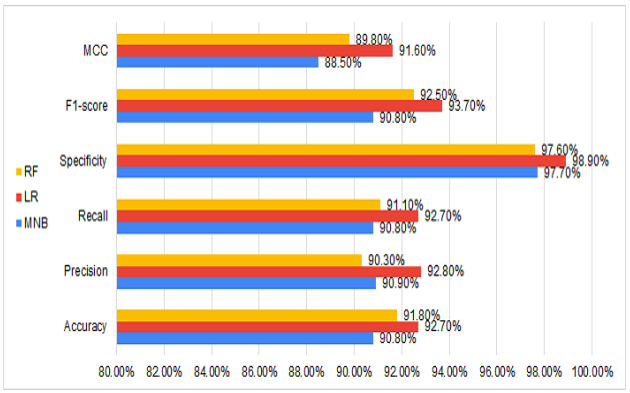

Figure 4. Train set result.

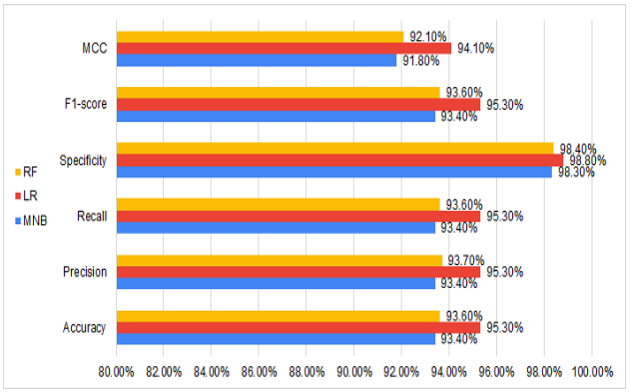

Figure 5. Prediction result.

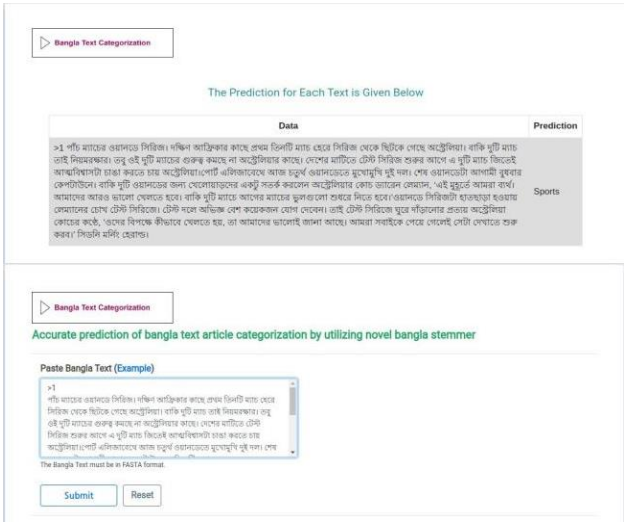

Figure 6. Model prediction: Input panel (above) and Prediction result (below).

Conclusion

This study presents a computational framework that utilizes machine learning to effectively classify Bangla text articles into multiple categories. The framework combines natural language processing techniques and algorithms to accurately analyze the linguistic features of Bangla text. It takes into account factors such as word frequency, sentence structure, and semantic meaning to

achieve high classification accuracy. Additionally, the framework can be easily adapted and applied to other languages with minimal modifications. We have shown that our framework is better by including data preprocessing steps, a novel stemmer mechanism for Bangla text articles, a vectorizer, a feature selector, and a predictor. We think that our proposed framework will be able to make a big difference in the scientific community by accurately and quickly putting multi-class Bangla text articles into the right categories. Finally, the proposed model has been deployed using the FLASK framework, and the efficiency of Bangla text recognition has been observed.

References

[1] A. Dhar, H. Mukherjee, N. S. Dash, and K. Roy, "Performance of Classifiers in Bangla Text Categorization," 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh, 2018, pp. 168–173. https://doi.org/10.1109/iciset.2018.8745621

[2] M. Mansur, "Analysis of N-Gram based text categorization for Bangla in a newspaper corpus," Bachelor thesis, BRAC University, 2006. http://hdl.handle.net/10361/61

[3] A. K. Mandal and R. Sen, "Supervised learning methods for Bangla web document categorization," International Journal of Artificial Intelligence & Applications, vol. 5, no. 5, pp. 93–105, 2014. https://doi.org/10.5121/ijaia.2014.5508

[4] F. Kabir, S. Siddique, M. R. A. Kotwal, and M. N. Huda, "Bangla text document categorization using Stochastic Gradient Descent (SGD) classifier," 2015 International Conference on Cognitive Computing and Information Processing (CCIP), Noida, India, 2015, pp. 1–4. https://doi.org/10.1109/ccip.2015.7100687

[5] M. S. Islam, F. E. M. Jubayer, and S. I. Ahmed, "A Comparative Study on Different Types of Approaches to Bengali document Categorization," 2017 International Conference on Engineering Research, Innovation and Education (ICERIE), Sylhet, Bangladesh, 2017, pp. 1-7. https://doi.org/10.48550/arXiv.1701.08694

[6] M. S. Islam, F. E. M. Jubayer, and S. I. Ahmed, "A support vector machine mixed with TF-IDF algorithm to categorize Bengali document," 2017 International conference on electrical, computer and communication engineering (ECCE), Cox's Bazar, Bangladesh, 2017, pp. 191–196. https://doi.org/10.1109/ecace.2017.7912904

[7] A. Dhar, N. Dash, and K. Roy, "Classification of text documents through distance measurement: An experiment with multi-domain Bangla text documents," 2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA), Dehradun, India, 2017, pp. 1–6. https://doi.org/10.1109/icaccaf.2017.8344721

[8] A. Dhar, N. S. Dash, and K. Roy, "Application of TF-IDF Feature for Categorizing Documents of Online Bangla Web Text Corpus," Intelligent Engineering Informatics: Proceedings of the 6th International Conference on FICTA, Springer, 2018, pp. 51–59. https://doi.org/10.1007/978-981-10-7566-7_6

[9] M. T. Alam and M. M. Islam, "Bard: Bangla article classification using a new comprehensive dataset," 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, Bangladesh, 2018, pp. 1–5. https://doi.org/10.1109/icbslp.2018.8554382

[10] S. B. S. Mugdha, S. M. Ferdous, and A. Fahmin, "Evaluating Machine Learning Algorithms for Bengali Fake News Detection," 2020 23rd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2020, pp. 1–6. https://doi.org/10.1109/iccit51783.2020.9392662

[11] S. B. S. Mugdha, M. B. M. M. Kuddus, L. Salsabil, A. Anika, P. P. Marma, Z. Hossain, and S. Shatabda, "A Gaussian Naive Bayesian Classifier for Fake News Detection in Bengali," Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 2. Springer, 2021, pp. 283–291. https://doi.org/10.1007/978-981-33-4367-2_28

[12] Grefenstette, G. "Tokenization. In: van Halteren, H. (eds) Syntactic Wordclass Tagging. Text, Speech and Language Technology", vol 9. Springer, Dordrecht, 1999. https://doi.org/10.1007/978-94-015-9273-4_9

[13] G. Grefenstette and P. Tapanainen, "What is a word, what is a sentence? Problems of Tokenization," 1994. https://api.semanticscholar.org/CorpusID:22436434

[14] J. B. Lovins, “Development of a stemming algorithm.” Mechanical Translation and Computational Linguistics, vol. 11, no. 1-2, pp. 22–31, 1968. http://chuvyr.ru/MT-1968-Lovins.pdf

[15] T. Pavlidis, “A vectorizer and feature extractor for document recognition,” Computer Vision, Graphics, and Image Processing, vol. 35, no. 1, pp. 111– 127, 1986. https://doi.org/10.1016/0734-189x(86)90128-3

[16] T. Joachims et al., “A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization,” 1997: Proceedings of the Fourteenth International Conference on Machine Learning, vol. 97. Citeseer, 1997, pp. 143–151. https://dl.acm.org/doi/abs/10.5555/645526.65727

[17] E. Vittinghoff, D. V. Glidden, S. C. Shiboski, and C. E. McCulloch, “Logistic Regression,” Regression methods in biostatistics: linear, logistic, survival, and repeated measures models, pp. 139–202, 2012. https://doi.org/10.1007/978-1-4614-1353-0_4

[18] B. Xu, X. Guo, Y. Ye, and J. Cheng, “An improved random forest classifier for text categorization.” Journal of Computers, vol. 7, no. 12, pp. 2913–2920, 2012. http://www.jcomputers.us/vol7/jcp0712-09.pdf

[19] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, “Multinomial Naive Bayes for Text Categorization Revisited,” AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17, Springer, 2005, pp. 488–499. https://doi.org/10.1007/978-3-540-30549-1_43