Detecting Offensive Language in Bengali, Bodo, and Assamese using Word Unigrams, Char N-grams, Classical Machine Learning, and Deep Learning Methods

Avigail Stekel, Avital Prives, Yaakov HaCohen-Kerner

Computer Science Department, Jerusalem College of Technology, Jerusalem 9116001, Israel

Abstract

In this paper, we, the JCT team, describe our submissions for the HASOC 2023 track. We participated in task 4, which addresses the problem of hate speech and offensive language identification in three languages: Bengali, Bodo, and Assamese. We developed different models using five classical supervised machine learning methods: multinomial Naive Bayes (MNB), support vector classifier, random forest, logistic regression (LR), and multi-layer perceptron. Our models were applied to word unigrams and/or character n-gram features. In addition, we applied two versions of relevant deep learning models. Our best model for the Assamese language is an MNB model with 5-gram features, which achieves a macro averaged F1-score of 0.6988. Our best model for Bengali is an MNB model with 6-gram features, which achieves a macro averaged F1-score of 0.66497. Our best submission for Bodo is a LR with all word unigrams in the training set. This model obtained a macro averaged F1-score of 0.85074. It was ranked in the shared 2nd-3rd place out of 20 teams. Our result is lower by only 0.00576 than the result of the team that was ranked in the 1st place. Our GitHub repository link is avigailst/co2023 (github.com).

Keywords

Char n-grams, hate speech, offensive language, supervised machine learning, word unigrams

1. Introduction

"Offensive language" lacks a universally agreed-upon definition. In the study of Jay and Janschewitz [1], offensive language is characterized as encompassing vulgar, pornographic, and hateful expressions. Xu and Zhu [2] observed that the interpretation of offensive language is subjective, as individuals can perceive the same content differently. Xu and Zhu adopted the Internet Content Rating Association's (ICRA) description of offensive language, categorizing it as text containing profanity, sexually explicit material, racism, graphic violence, or any content that might be deemed offensive based on social, religious, cultural, or moral standards. Another widely accepted interpretation of offensive language is any explicit or implicit form of attack or insult directed at an individual or group.

The prevalent use of offensive language constitutes a significant challenge within online communities and among their users. Instances of offensive language proliferate rapidly across social networks like Twitter, Facebook, and blog posts. This trend detrimentally impacts the credibility of these online communities, hindering their expansion and causing user detachment.

Distinguishing between offensive language and hate speech in contrast to non-offensive language and non-hate speech is a complex endeavor due to several factors. First, hate speech does not always rely on offensive slurs, and offensive language does not consistently convey hatred. Second, there exists a wide array of implicit and explicit methods to verbally target individuals or groups. Third, the brevity of

Forum for Information Retrieval Evaluation, December 15-18, 2023, Goa, India
EMAIL: Stekel@g.jct.ac.il (A. Stekel); avitalprives@gmail.com (A .Prives); kerner@jct.ac.il (Y. Kerner)
ORCID: 0000-0002-4834-1272

certain tweets adds to the challenge. Finally, the presence of incoherent tweets further complicates matters.

A recent outcome arising from addressing this challenge has been the establishment of several competitions focused on identifying various forms of offensive language across diverse languages, including but not limited to English, German, Hindi, Tamil, Marathi, and Malayalam. Notable instances of these contests include HASOC 2019 [3], HASOC 2020 [4], HASOC 2021, HASOC 2022, SemEval-2019 [5], and SemEval-2020 [6]. Within these tournaments, leveraging natural language processing (NLP) and machine learning (ML) models to detect offensive language has demonstrated its effectiveness.

Particularly vulnerable user segments, such as the elderly, children, youth, women, and certain minority groups, are exposed to various risks stemming from encountering offensive content. These risks encompass emotions like fear, panic, and animosity directed at specific individuals or communities, potentially resulting in adverse effects on their mental and physical well-being.

The rationale behind researching the detection of offensive language is quite evident. A clear need exists for top-tier systems capable of identifying offensive language posts, curbing their dissemination, and alerting appropriate authorities. The implementation of such systems stands to enhance the safeguarding and security of individuals, particularly in contexts closely tied to their physical and mental health.

The structure of the rest of the paper is as follows. Section 2 introduces the general background concerning offensive language. Section 3 describes the HASOC 2023 Subtask 4. In Section 4, we present the applied models and their experimental results. Section 5 summarizes, concludes, and suggests ideas for future research.

2. Related Work

According to the United Nations (UN) definition [7], hate speech is "any type of communication in speech, writing or behavior that attacks or uses derogatory or discriminatory language in reference to a person or group on the basis of who they are, in other words, on the basis of their religion, ethnicity, nationality, race, color, origin, gender or other identity factor." Some studies [8-9] characterized hate speech as messages marked by hostility and aggression, often referred to as flames. In more recent studies [10-12], there has been a shift toward using the term "cyberbullying" to describe these harmful online behaviors. Nevertheless, within the Natural Language Processing (NLP) community, a range of terms is employed to encompass the realm of hate speech, including discrimination, flaming, abusive language, profanity, toxic discourse, or derogatory comments [13]. These various terms collectively encompass the multifaceted nature of offensive and harmful speech in the digital sphere.

Most of the studies in the field of hate and offensive speech recognition have primarily centered on widely spoken languages, such as English, while the challenges posed by less-represented languages, including Assamese, Bodo, and Bengali, have garnered increased attention Notable studies have delved into these challenges by examining the nuances of identifying hate speech and offensive content in these languages. For instance, Ishmam et al. [14] introduced a ML-based model, as well as Gated Recurrent Unit (GRU), based deep neural network model for classifying users' comments on Facebook pages in the Bengali language. Baruah et al. [15] suggested multinomial naive Bayes (MNB) and support vector machine (SVM) with various word embedding and n-gram models as classification algorithms to detect an offensive language in Assamese text. These investigations serve as pioneering efforts in developing culturally sensitive solutions for detecting hate and offensive speech across linguistically diverse landscapes.

HaCohen-Kerner and his students have experience from previous workshops that dealt with offensive language detection [16-19].

3. Task Description

The Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages (HASOC) 2023 track includes four tasks. We took part in Task 4, which aims to detect hate speech in the Bengali, Bodo, and Assamese languages. It is a binary classification task. Each dataset (for the three languages) consists of a list of sentences with their corresponding class: hate or offensive (HOF) or not hate (NOT). Data is primarily collected from Twitter, Facebook, or YouTube comments. The macro averaged F1-score is the result measure of this task.

The overview of the HASOC Sub-track at FIRE 2021 is described in [20]. Additional information about Subtask 4 in Assamese, Bengali, and Bodo is described in [21]. The HASOC 2023 train and test datasets for Bengali, Bodo, and Assamese are located at [22].

4. Applied Models and Their Experimental Results

We used the given training and test datasets (see the end of the previous section). Due to time limitations, (We joined the competition late), we did not apply any preprocessing methods. We applied five classical supervised ML methods: Multinomial Naive Bayes (MNB), Random Forest (RF), Support Vector Classifier (SVC), Multi-Layer Perceptron (MLP), and Logistic Regression (LR) using classical features such as word unigrams and char n-gram features and features.

MNB is a statistical ML algorithm based on the Bayes theorem (Kim et al., 2006). MNB assumes that the features (i.e., attributes) are conditionally independent given the target class, and ignores all dependencies among features. MNB estimates the probabilities of each class and the probabilities of each feature given the class and uses these probabilities to make predictions.

RF is an ensemble learning method for classification and regression [23]. Ensemble methods use multiple learning algorithms to obtain improved predictive performance compared to what can be obtained from any of the constituent learning algorithms. RF operates by constructing a multitude of decision trees at training time and outputting classification for the case at hand. RF combines Breiman's "bagging" (Bootstrap aggregating) idea [24] and a random selection of features introduced by Ho [25] to construct a forest of decision trees.

SVC is a variant of the support vector machine (SVM) ML method [26] implemented in SciKit-Learn. SVC uses LibSVM [27], which is a fast implementation of the SVM method. SVM is a supervised ML method that classifies vectors in a feature space into one of two sets, given training data. It operates by constructing the optimal hyperplane dividing the two sets, either in the original feature space or in higher dimensional kernel space.

MLP is a deep, artificial neural network [28]. This model is based on a network of computational units, called perceptron, interconnected in a feed-forward way. The network is composed of layers of perceptron where each one has directed connections to the neurons of the subsequent layer. Usually, these units apply a sigmoid function, called the activation function, on the input they get and feed the next layer with the output of the function. This model is very useful especially when the data is not linearly separable.

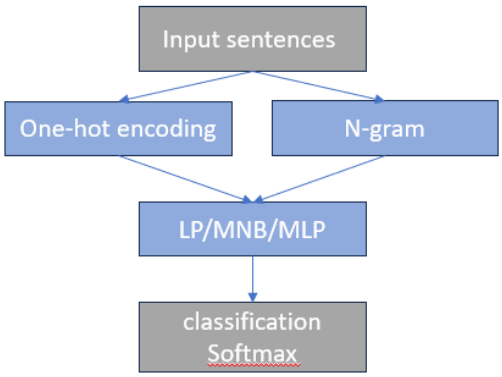
LR [29-30] is a linear classification model. It is known also as maximum entropy regression (MaxEnt), logit regression, and the log-linear classifier. In this model, the probabilities describing the possible outcome of a single trial are modeled using a logistic function. Generally, a sigmoid function is used as a predictive function. LR can be used both for binary classification and multi-class classification.

BERT [31] (Bidirectional Encoder Representations from Transformers) is a transformer-based model that was trained on a massive corpus of text data, allowing it to learn rich representations of the relationships between words and their meaning. These representations can be fine-tuned for specific NLP tasks, e.g., TC, by tokenizing the text and converting it to numerical representations using pre-trained tokenizers. These representations are fed into the pre-trained BERT model to obtain contextualized representations of the input text (Chi et al., 2019). These representations can be thought of as a fixed-length vector, which is then passed through a fully connected neural network (NN) for classification. One key advantage of using BERT for TC is that it can handle contextual information effectively.

BanglaBERT [32] is a language model designed to understand and process the Bengali language, also known as Bangla. It's part of the BERT (Bidirectional Encoder Representations from Transformers) family of models that have proven effective in various natural language processing tasks. The purpose of BanglaBERT [33] is to facilitate various language-related tasks in Bengali, even in scenarios where there's limited training data available (low-resource settings). By pre-training on a vast corpus of Bengali text, BanglaBERT learns to represent the nuances of the language and can be fine-tuned for specific tasks such as text classification, sentiment analysis, and more. This enables more effective natural language understanding and processing for the Bengali language.

The system architecture we used is described in Figure 1. This figure shows the procedure we performed on the input sentence and the use of the algorithm mentioned before.

Figure 1: System architecture description



The applied ML methods used the following tools and information sources:

- The Python 3.8 programming language [34].
- Sklearn – a Python library for ML methods [35].
- Numpy – a Python library that provides fast algebraic calculous processing [36].
- Pandas – a Python library for data analysis. It provides data structures for efficiently storing large datasets and tools for working with them [37].
- Pytorch - open-source ML framework for building, training, and deploying neural network models.

In our experiments, we test dozens of TC models for each language. We applied the models on the given training set. During the experiments, we checked what happens when we use all the existing words, and also what happens when we take only common words that appear in at least two or three documents.

Tables 1-3 present the F-Measure results of our baseline models for Bengali, Bodo, and Assamese, respectively. As mentioned above, we applied five different supervised ML methods: multinomial Naive Bayes, support vector classifier, random forest, logistic regression, and multi-layer perceptron using their default values. For these baseline models, we use only word unigrams that occur in at least 2 documents in the training set.

In our initial experiments, we randomly split each tournament train dataset into two sub-sets: train sub-set (80% of the original train sub-set) and test sub-set (20% of the original train sub-set). In the train sub-set: in the Assamese language, 27,570 words appear in three tweets or more, in the Bengali language 1,648 words appear in three tweets or more, and in the Bodo language 1,066 words appear in three tweets or more.

In Tables 1-3, we present the best baseline results in Bengali, Bodo, and Assamese respectively. The best baseline result in each table is highlighted in bold font.

Table 1
Baseline F-Measure results for word unigrams in Bengali.

Number of features	MNB	SVC	RF	LR	MLP
500	0.545991	0.480316	0.380032	0.564075	0.556609
1000	0.580773	0.467054	0.380032	0.556989	0.558711
1500	**0.599029**	0.439857	0.380032	0.564893	0.548554

Table 2
Baseline F-Measure results for word unigrams in Bodo.

Number of features	MNB	SVC	RF	LR	MLP
500	0.68221	0.64635	0.380081	0.670617	0.673706
1000	0.743217	0.74287	0.37469	**0.775797**	0.750886

Table 3
Baseline F-Measure results for word unigrams in Assamese.

Number of features	MNB	SVC	RF	LR	MLP
500	0.512793	0.481618	0.37416	0.510662	0.547265
1000	0.591038	0.525424	0.37416	0.579291	0.587754
1500	0.605971	0.56497	0.37416	0.588314	0.602868
2000	0.632991	0.563373	0.37416	0.608565	0.616363
2500	**0.656912**	0.581495	0.37416	0.620213	0.622697

We ran the baseline models on different numbers of words, and reached the results described in the above tables, some are better, and some are less. In the Bodo language, using LR with 1,000 word unigrams[2] we reached an F-Measure of 0.775795. In the other languages, the results were lower.

Later we applied character n-gram series for n values between 3 and 7. We also ran combinations of different sizes of BOWs with different character n-gram series, which caused an increase in F1 for the Assamese and Bengali languages and reached them, using a combination of BOW and character n-grams, to F-Measure of 0.6988 and 0.66497, respectively.

We also applied two types of Bert models: all-language Bert, which is a general Bert model that is not adapted to a specific language, and a Bengali Bert model, also called Bert2, which is a Bert model adapted to the Bengali language. In the Assamese language, we reached a result of 0.66967 for running Bert and MNB[3], in the Bengali language we reached a result of 0.609 when we ran the Bert2 model, and in the Bodo language, we reached a result of 0.73 when we ran Bert and MLP. We applied also MLP[4], which yielded a result of 0.7952 for the Bodo language and less good results for the other languages.

For each language, we submitted various models including the top three models according to their F-Measure results. Our best F-Measure results in the competition were as follows: Assamese (F-Measure = 0.6988, 10th place) using MNB with all word and character 5-gram features, Bengali (F-Measure = 0.66497, 12th place) using MNB with all word and 6-grams, and Bodo (F-Measure = 0.85074, 2nd place). Our best submission was the model we built for offensive language identification in Bodo using LR. This

[2] only words that appear in two or more documents in the training set.
[3] https://www.ic.unicamp.br/~rocha/teaching/2011s1/mc906/aulas/naive-bayes.pdf
[4]https://www.researchgate.net/profile/Francisco-Escobar/publication/320692297_Geomatic_Approaches_for_Modeling_Land_Change_Scenarios_An_Introduction/links/5e0da50a92851c8364ab9b63/Geomatic-Approaches-for-Modeling-Land-Change-Scenarios-An-Introduction.pdf#page=458

model was ranked in 2nd place out of 20 teams. Our result is lower by only 0.00576 than the result (0.8565) of the team that was placed in the 1st place.

Table 4 describes the best F-score we got in the three languages. The best result for each language is bold.

Table 4
Our best submitted models and their F-Measure results in Assamese, Bengali, and Bodo.

language	method	result
Assamese	MNB using all character 5-gram features and all word unigrams in the training set	**0.6988**
Assamese	MNB using all character 5-gram features and only words that were in two or more documents	0.6946
Assamese	MNB using all character 4-gram features and all word unigrams in the training set	0.6941
Assamese	MNB using only character 5-gram features that were in two or more documents and all word unigrams in the training set	0.69213
Bengali	MNB using all character 6-gram features and all word unigrams in the training set	**0.66497**
Bengali	MNB using only character 6-gram features that were in two or more documents and only words that were in two or more documents	0.66032
Bengali	MNB using only character 5-gram features that were in two or more documents and only words that were in two or more documents	0.65691
Bengali	MNB using all character 5-gram features and all word unigrams in the training set	0.65215
Bodo	LR using all word unigrams in the training set	**0.85074**
Bodo	LR using only words that were in two or more documents	0.84607
Bodo	LR using all character 4-gram features and all word unigrams in the training set.	0.8399
Bodo	MNB using only character 4-gram features that were in two or more documents and only words that were in two or more documents	0.83703

An interesting phenomenon is that in two languages (Assamese and Bengali), the MNB method was found to be the best among five classical learning methods and two variants of BERT. In the third language (Bodo), LR was found as the best ML method. However, in Bodo, a number of good models using MNB were discovered. MNB is a popular classifier for many text classification tasks, due to its simplicity, computational efficiency, relatively good predictive performance, and trivial scaling to large-scale tasks [38]

5. Summary, Conclusions, and Future Work

In this paper, we, the JCT team, described our submitted models for subtask 4 of the HASOC 2021 competition, which addresses the problem of hate speech and offensive language identification in three languages: Bengali, Bodo, and Assamese. We applied classical ML methods and deep learning methods: MNB, SVC, MLP, RF, and LR. These ML methods were applied to various combinations of character n-gram features (for n values from 1 to 7) and word unigrams.

Two interesting phenomena were discovered. First, while in Bodo the use of a classical learning method like LR was enough for a high result and **s**hared the 2nd-3rd place. Second, in the Assamese and Bengali languages, the use of classical learning methods such as RF, LR, and SVC did not yield good enough results, and precisely the naive MNB model produced the best results.

The HOF and NOT classes are unbalanced. In the Assamese language, the HOF group is about 16% larger than the NOT group, while in the Bengali language, the NOT group is about 19.5% larger than the HOF group, and in the Bodo language, the HOF group is 19% larger than the NOT group. In future

research, we can apply oversampling in order to balance the classes. Oversampling is a technique used in machine learning to balance the class distribution by increasing the frequency of the minority class in the training dataset.

Additional ideas for future research are: (1) parameter tuning, also known as hyperparameter tuning, which is the process of finding the best combination of hyperparameters for a ML model to achieve optimal performance on a specific task or dataset, (2) application of various preprocessing methods [39], and (3) definition and application of style-based and content-based features and combinations of them [40].

6. References

[1] T. Jay, K. Janschewitz, The pragmatics of swearing, Journal of Politeness Research 4 ,2008, 267-288.

[2] Z. Xu and S. Zhu, Filtering offensive language in online communities using grammatical relations. In Proceedings of the Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference, 2010, pp. 1-10.

[3] T. Mandl, S. Modha, P.,Majumder, D. Patel, M. Dave, C. Mandlia and A. Patel, Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In Proceedings of the 11th forum for information retrieval evaluation, 2019, pp. 14-17.

[4] T. Mandl, S. Modha, M, A. Kumar, B. R. Chakravarthi, Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german. In Forum for Information Retrieval Evaluation, 2020, pp. 29-32.

[5] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval), 2019, arXiv preprint arXiv:1903.08983.

[6] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak and Ç. Çöltekin, SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020), 2020, arXiv preprint arXiv:2006.07235.

[7] United Nations Office of the High Commissioner for Human Rights. (n.d.). Hate Speech.

[8] E. Spertus, Smokey: Automatic recognition of hostile messages, in: Aaai/iaai, 1997, pp.1058–1065.

[9] D. Kaufer, Flaming: A white paper, Department of English, Carnegie Mellon University, Retrieved July 20 ,2000.

[10] J.M. Xu, K.S. Jun, X. Zhu and A. Bellmore, Learning from bullying traces in social media, in: Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies, 2012, pp. 656–666.

[11] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, S. Mishra, Detection of cyberbullying incidents on the instagram social network, 2015, arXiv preprint arXiv:1503.03909.

[12] H. Zhong, H. Li, A. C. Squicciarini, S. M. Rajtmajer, C. Griffin, D. J. Miller, C. Caragea, Content-driven detection of cyberbullying on the instagram social network, in: IJCAI, 2016, pp. 3952–3958.

[13] A. Schmidt, M. Wiegand, A survey on hate speech detection using natural language processing, in: Proceedings of the Fifth International workshop on natural language processing for social media, 2017, pp. 1–10.

[14] A. Ishmam, and S. Sadia. "Hateful speech detection in public facebook pages for the bengali language." 2019 18th IEEE international conference on machine learning and applications (ICMLA). IEEE, 2019.

[15] N. Baruah, G. Arjunand N. Mandira,"Detection of Hate Speech in Assamese Text." International Conference on Communication and Computational Technologies. Singapore: Springer Nature Singapore, 2023.

[16] Y. HaCohen-Kerner, Z. Ben-David, G, Didi, E. Cahn, S. Rochman, and E. Shayovitz. JCTICOL at SemEval-2019 Task 6: Classifying offensive language in social media using deep learning methods, word/character n-gram features, and preprocessing methods. In Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 645-651.

[17] M. Uzan and Y. HaCohen-Kerner. JCT at SemEval-2020 Task 12: Offensive language detection in tweets using preprocessing methods, character and word n-grams. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, 2020, pp. 2017-2022.
[18] M. Uzan and Y. HaCohen-Kerner. Detecting Hate Speech Spreaders on Twitter using LSTM and BERT in English and Spanish. CLEF, 2021, pp. 2178-2185.
[19] Y. HaCohen-Kerner and M. Uzan. Detecting Offensive Language in English, Hindi, and Marathi using Classical Supervised Machine Learning Methods and Word/Char N-grams. Forum for Information Retrieval Evaluation (FIRE), CEUR-WS. Org. 2021.
[20] T. Ranasinghe, K. Ghosh, A. S. Pal, A. Senapati, A. E. Dmonte, M. Zampieri, S. Modha, and S. Satapara, Overview of the HASOC Subtracks at FIRE 2023: Hate speech and offensive content identification in Assamese, Bengali, Bodo, Gujarati, and Sinhala. In Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2023, Goa, India, December 15-18, 2023, ACM.
[21] K. Ghosh, A. Senapati, and A. S. Pal, Annihilate Hates (Task 4, HASOC 2023): Hate Speech Detection in Assamese, Bengali, and Bodo Languages, In Working Notes FIRE 2023 - Forum for Information Retrieval Evaluation, CEUR, December 15-18, 2023.
[22] K. Ghosh, A. Senapati, and A. S. Pal, Annihilate Hates Datasets, URL: https://sites.google.com/view/hasoc-2023-annihilate-hates/home.
[23] L. Breiman, Random forest, Machine Learning 45(1) , 2001, 5-32.
[24] L. Breiman, Bagging predictors, Machine Learning 24(2) , 1996, 123-140.
[25] T. K. Ho, Random decision forests, In Proceedings of 3rd International Conference on Document Analysis and Recognition, 1995, Vol. 1, pp. 278-282, IEEE.
[26] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 , 1995, 273–297.
[27] C.-C., Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM transactions on intelligent systems and technology (TIST) 2 , 2011, 1–27.
[28] S. K. Pal, S. Mitra, Multilayer perceptron, fuzzy sets, classification, IEEE transactions on Neural Networks 3(5), 1992, 683-697.
[29] Cox, D. R. The regression analysis of binary sequences. Journal of the Royal Statistical Society Series B: Statistical Methodology, 20(2), 215-232. 1958.
[30] D. W. Hosmer Jr, S. Lemeshow, R. X. Sturdivant, Applied logistic regression, Vol. 398, John Wiley & Sons. Applied logistic regression (Vol. 398). John Wiley & Sons, 2013.
[31] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018.
[32] B. Abhik, H. Tahmid, A. Wasi Uddin, S. Kazi, I. Md Saiful, I. Anindya, R.M. Sohel and S. Rifat, BanglaBERT: Language Model Pretraining and Benchmarks for Low-Resource Language Understanding Evaluation in Bangla. arXiv preprint arXiv: 2101.00204. 2022.
[33] M. Kowsher, A.A. Sami, N.J. Prottasha, M.S. Arefin, P.K. Dhar, and T. Koshiba, Bangla-BERT: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*, *10*, 91855-91870. 2022.
[34] Van Rossum, Guido, and Fred L. Drake. Introduction to python 3: python documentation manual part 1. CreateSpace, 2009.
[35] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... & Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. 2013. *arXiv preprint arXiv:1309.0238*.
[36] Harris, C.R., Millman, K.J., Van Der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J. and Kern, R., Array programming with NumPy. *Nature*, *585*(7825), pp.357-362. 2020.
[37] McKinney, Wes. "Data structures for statistical computing in python. Proceedings of the 9th Python in Science Conference. Vol. 445. No. 1. 2010.
[38] E. Frank, and R.R. Bouckaert, Naive bayes for text classification with unbalanced classes. In Knowledge Discovery in Databases: PKDD 2006: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings 10 (pp. 503-510). Springer Berlin Heidelberg. 2006.

[39] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, The influence of preprocessing on text classification using a bag-of-words representation. PloS one, 15(5), e0232525. 2020.
[40] Y. HaCohen-Kerner, H. Beck, E. Yehudai, M. Rosenstein, and D. Mughaz, Cuisine: Classification using stylistic feature sets and/or name-based feature sets, Journal of the American Society for Information Science and Technology 61(8) ,2010 , 1644-1657.