Team_Syrax at BLP-2023 Task 1: Data Augmentation and Ensemble Based Approach for Violence Inciting Text Detection in Bangla

Omar Faruqe Riyad
Shahjalal University of
Science & Technology
omar42@student.sust.edu

Trina Chakraborty
Shahjalal University of
Science & Technology
trina41@student.sust.edu

Abhishek Dey
Shahjalal University of
Science & Technology
abhishek21@student.sust.edu

Abstract

This paper describes our participation in Task 1 (VITD) of BLP Workshop [1] at EMNLP 2023, focused on the detection and categorization of threats linked to violence, which could potentially encourage more violent actions. Our approach involves fine-tuning of pre-trained transformer models and employing techniques like self-training with external data, data augmentation through back-translation, and ensemble learning (bagging and majority voting). Notably, self-training improves performance when applied to data from external source but not when applied to the test-set. Our analysis highlights the effectiveness of ensemble methods and data augmentation techniques in Bangla Text Classification. Our system initially scored 0.70450 and ranked 19th among the participants but post-competition experiments boosted our score to 0.72740.

1 Introduction

In today's social media-driven world, easy self-expression has brought a downside: a surge in harmful, violent content harming people physically and mentally (Mathew et al., 2019). This critical concern needs addressing.

The EMNLP-2023 organized the BLP Shared Task 1 (VITD) (Saha et al., 2023a), addressing a vital challenge: identifying violence-inciting text in Bangla. The aim was to build models that identify violent content, especially content that might provoke more violence. Yet, in Bangla, this is tough due to limited language resources. The text from YouTube comments often lacks clear context, making it even harder to understand. Additionally, the dataset (Saha et al., 2023b) used in this task is relatively small, limiting the variety of language patterns. To overcome these issues, we used pre-trained transformer models, fine-tuning them with VITD dataset. We also applied techniques like self-training on external data, back-translation for data augmentation, and ensemble learning (Bagging and Majority Voting). These techniques, particularly when combined with self-training and back-translation, as well as Ensemble approach across multiple models, moderately improved our model's performance.

Post-competition experiments, including self-training on external data and back-translation, raised our score to 0.72740. This paper details our approach, challenges, and methods for addressing violence-inciting text in Bangla.

2 Related Work

Hate speech, cyberbullying, harassment, and incitements to violence on social platforms can harm individuals and communities in online spaces. Increasing studies have been undertaken to detect violent content on social media (Dikwatta and Fernando, 2019; Jahan and Oussalah, 2023a; Zampieri et al., 2020). People usually confront violence on social media through text, images, and videos. Researchers use natural language processing (Jahan and Oussalah, 2023b) to analyze text, visual, and audio content on social media sites. These excellent initiatives are happening worldwide in many languages. Implementing the same method in languages with low resources, like Bangla, is problematic (Das et al., 2022a). Poorly annotated Bangla-language violence detection datasets are a widespread issue (Romim et al., 2022). Bangla has a large vocabulary and several sentence forms (Das et al., 2022b). Bangla dialects vary by region, which might alter text interpretation. Although Bangla is a low-resource language (Karim et al., 2021) with its own issues, numerous studies (Emon et al., 2022) are being undertaken to identify social media breaches in this language context. Modern models, such as BERT (Mridha et al., 2021), have been substantially altered and used in these studies. These evolving investigations are encouraging us to use these modern approaches for low-resource lan-

[1] https://blp-workshop.github.io/

Proceedings of the First Workshop on Bangla Language Processing (BLP-2023), pages 247–254
December 7, 2023 ©2023 Association for Computational Linguistics

guages like Bangla (Keya et al., 2023; Kumari et al., 2023). Changing studies have provided new perspectives on violence recognition in Bangla (Jahan et al., 2022; Caselli et al., 2020) and expanded our knowledge of it. BanglaBert (Sharif et al., 2022) was a key tool in our study for addressing BLP Task 1. We wanted to get the most out of ensemble methods by using pre-trained transformers in our experiments (Risch and Krestel, 2020). This is because the field of transformer model applications is still growing and changing (Das et al., 2023). We were able to use the combined knowledge of several cutting-edge transformer models with this new method, which made our experiments more in-depth and varied. Several well-thought-out tests with multiple models have yielded key results and refined our method to make it more accurate.

3 Task and Dataset Description

The BLP Shared Task 1, known as Violence Inciting Text Detection (VITD), offers an outstanding chance to address the significant problem of detecting violence-inciting text. The dataset being analyzed consists of YouTube comments containing the most significant violent incidents. Three distinct categories are established for the purpose of classification: Direct Violence, which includes explicit threats targeted towards individuals or communities; Passive Violence, which involves the utilization of derogatory language, abusive remarks, or justification of violence; and Non-Violence, which encompasses content that is unrelated to subjects involving violence. The task has a role in the identification and mitigation of potential threats that may lead to violent situations.

The VITD dataset is divided into three subsets: the training set, development set, and testing set, all of which are formatted in CSV structure. Each entry within these CSV files consists of two key columns: “text” and “label.” The “text” column contains textual data collected from various social media sources, while the “label” column assigns a numerical value of 0, 1, or 2 to each entry, representing different categories of violence: Non-Violence, Passive Violence, and Direct Violence, respectively. In Appendix A.1, as shown in Figure 4, we tried to visualize the category distribution within each set and noticed that, the datasets are highly skewed towards Non-Violence. Occurrence of Direct Violence is very rare. The class distribution within the dataset is imbalanced, with Non-Violence being the dominant category. Detecting and classifying the less frequent instances of Passive Violence and Direct Violence poses a significant challenge. We also tried to visualize the texts associated with the labels through wordclouds in figure 5, 6, 7 in the A.1 appendix section. The distribution of words in the wordcloud provides some insights. We discovered some words that are uniquely associated with a given label. Along with that, we also noticed, the datasets contain instances of ambiguous labeling, in which the categorization of text into the correct category of violence is difficult due to the complexity and ambiguity of the language. Given the nature of text inciting violence, the dataset may contain instances of religious bias. During annotation, it is crucial to deal with this sensitivity and maintain an ethical perspective.

4 System Description

4.1 Data Pre-processing

In our data processing pipeline, cleaning and preprocessing the text data were involved as a necessary step. This was a meticulous and essential process that aimed to enhance the quality and reliability of the information we were working with. To begin with, we focused on the elimination of unwanted elements in the text. This included the removal of emojis and excess punctuation marks. Emojis, while adding expressive elements to text, are often regarded as noise in many natural language processing tasks. Removing emojis was essential to simplify the text and make it more amenable to analysis and modelling. Additionally, excess punctuation, such as multiple consecutive exclamation marks or question marks, can disrupt the flow of the text and create challenges for subsequent processing. By clearing the text of such redundancy, we aimed to make it cleaner and more straightforward. However, it’s worth noting that we made a conscious decision not to remove Bangla stop words during this pre-processing stage. Stop words are commonly occurring words in a language are often excluded from text analysis because they don’t carry substantial meaning on their own. However, when working with the Bangla language, we found that removing these stop words could sometimes alter the intended meaning of the text. To avoid such unintended alterations in meaning, we decided to retain Bangla stop words in our preprocessing steps.

4.2 Transformer based model

Transformer utilizes a mechanism called self-attention to process words in parallel, enabling it to capture intricate relationships and nuances within the text (Vaswani et al., 2017). By employing large-scale pre-training on vast text corpora, transformers gain a deep understanding of language. This general language knowledge, when fine-tuned for specific tasks, empowers them to excel in various applications including text classification. In our study of transformer based models for Bangla, we considered three main options: BanglaBert (Bhattacharjee et al., 2022), XLM-R (Conneau et al., 2019), and mBERT (Devlin et al., 2018). Both XLM-R and mBERT are pre-trained on a large amount of multilingual textual data but BanglaBert stands out due to its specific training on a large Bangla text dataset. This focused training equips it with a deep understanding of Bangla's unique language patterns, making it more effective than generic "BERT" models. It performs especially well in low-resource scenarios.

4.3 Semi-Supervised Learning: Self-Training

The VITD dataset is relatively small and has imbalanced class distribution (described in section 3). To address this, we adopted a semi-supervised learning method called self-training (Dong and de Melo, 2019). Initially, we trained our model on the train-set. Then, we used this model to label additional unlabeled data, expanding our training dataset. When we used test-set predictions as additional data, our model performed well in dev-set but not on the test-set. This happened because the test set contained some incorrect labelling from the model predictions. Additionally, we utilized self-training with external data. We selected 1500 data points from a Bangla Hate Speech dataset (Karim et al., 2020) and automatically annotated them. We filtered the newly annotated data, keeping all data points with labels 1 and 2 but only some with label 0 randomly, focusing on minority classes. Then, we combined this enriched dataset with our original training data. While this strategy resulted in a slight performance boost, it also diversified our dataset with a wider range of samples.

4.4 Data Augmentation: Back-Translation

We used back-translation technique (Sennrich et al., 2016) to increase diversity and size of data. We created a new dataset by translating Bangla sentences to English and back to Bangla using the Googletrans [2] API. We randomly combined the new dataset with the original data. This method enhances words and sentence variations by representing the words with semantic similarity in different form. Moreover, the VITD dataset, which includes YouTube comments, contains many grammatical errors and spelling mistakes. Back-translation using the Googletrans API corrects a significant portion of these errors. Combining both the back-translated data and the original data for training allows the model to recognize their semantic similarity and thus improving performance. It's essential to highlight that we conducted a manual quality check on the back-translated data to ensure its integrity and semantic similarity with the original dataset.

4.5 Ensembling

To enhance the robustness of our complex Transformer models, which tend to be sensitive to factors like initialization and data order, particularly when fine-tuned on small datasets (Dodge et al., 2020), we implement an ensemble method based on bootstrap aggregating (bagging) (Risch and Krestel, 2020) and hard majority voting. Bagging involves training multiple instances of the same model on various subsets of the training data through random re-sampling. This introduces randomness and reduces variance in the training process. In our study, we utilized seven different models for majority voting. The first model was trained on BanglaBert, while the second model was trained using a self-training approach on the first model. The remaining five models employed bagging, where we augmented the train-set with the dev-set. The final prediction was determined by taking the majority voting of individual model predictions. This ensemble strategy illustrated in Figure 2 was our best performing system during competition.

In the post-competition experiments, we implemented a majority voting system involving three top-performing models (Figure 1). The first model used a combination of the train-set and model-annotated external data. The second model combined the train-set, back-translated train-set, and back-translated dev-set. The third model was a result of a majority voting ensemble involving various experimented models. If there was a tie in the votes for two or more labels, we selected the label based on the model with the highest F1-score

[2] https://pypi.org/project/googletrans/

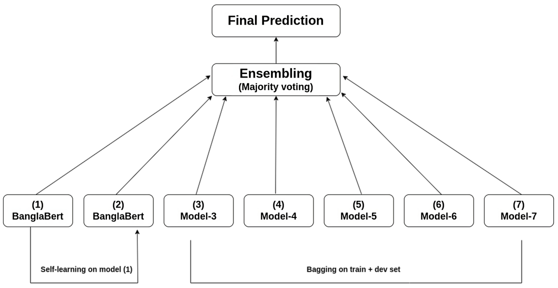


Figure 1: The overall best performing system

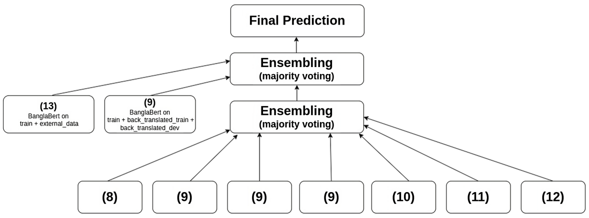


Figure 2: The best performing system during competition

(model 9). Our final macro-F1 score improved to 0.72740 compared to our competition performance, which achieved a macro-F1 score of 0.70450.

5 Experiment and Results

5.1 Experimental Setup

We utilized the Huggingface Transformers [3] library to construct our system. We employed pre-trained tokenizers and language models for fine-tuning. Training was conducted with a learning rate of 1e-5 and a batch size of 16. AdamW (Loshchilov and Hutter, 2019) optimizer is used to update the parameters. Model performance was assessed every 250 steps, with metrics including accuracy, precision, recall, and macro-F1 scores. Training lasted for 50 epochs, with early stopping implemented to select the best checkpoint based on the highest validation macro-F1 score. Our code is publicly available at Github [4].

5.2 Results

In this section, we present the performance results of our trained models [5], evaluated on the test-set released at the end of the competition.

Table 1 showcases the macro F1-scores of various models we experimented with, both during and after the competition. Notably, the dev-set scores were the main factor of our model selection during the evaluation phase. However, we observed that the best-performing model on the dev-set did not always translate to superior performance on the test-set. For instance, while model (2) outperformed model (1) on the dev-set, but this wasn't the case in the test-set. Our analysis revealed that the inclusion of back-translated data and model-annotated external data moderately improved model performance.

Model	Macro-F1	
(1)	0.70296	
(2)	0.69288	
(3)	0.70079	Bagging
(4)	0.67752	
(5)	0.67632	
(6)	0.70280	
(7)	0.70919	
(8)	0.71326	Back Translation Included
(9)	**0.71977**	
(10)	0.70519	
(11)	0.70521	
(12)	0.71136	
(13)	0.71866	Extra-Data

Table 1: Individual Model Performance Metrics

For example, model (13), which is BanglaBert trained on the train-set combined with the model-annotated external dataset, achieved a macro F1-score of 0.71866. Model (9), which is BanglaBert trained on the train-set, back-translated train-set, and back-translated dev-set, achieved the highest macro F1-score of 0.719771 among models without utilizing majority voting.

To further improve our results, we employed ensemble methods. Table 2 presents the macro F1-scores of our ensemble approach. The first model (E1) with an F1-score of 0.70450 represented our final submission during the competition. One thing to note from second ensemble method (E2) that, we incorporated 3 votes from model (9), as it consistently demonstrated the highest accuracy throughout our experiments. Our post-competition experimentation unveiled that the third model (E3) exhibited a score of 0.72740 which is the highest overall F1-score. This was attained by employ-

[3] https://huggingface.co/docs/transformers/index

[4] https://tinyurl.com/bde9cf6w

[5] The models corresponding to the numbers in the figure can be found in the Table 3 of Appendix A.2

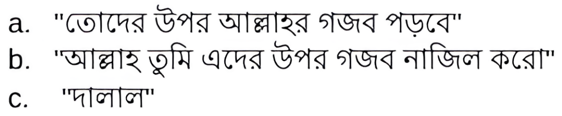
Model	Ensemble	Macro-F1
(E1)	(1)(2)(3)(4)(5)(6)(7)	**0.70450**
(E2)	(8)(10)(11)(12)(9)(9)(9)	0.71808
(E3)	(13)(9)(E2)	**0.72740**

Table 2: Ensemble Model Performance Metrics

ing a majority voting strategy among the three best-performing models. The experimental results emphasizes the significance of ensemble methods and data augmentation techniques in improving the detection of Violence Inciting Text in the Bangla language. The inclusion of back-translated data and model-annotated external data enriched our training dataset, leading to substantial performance gains.

6 Limitations and Error analysis

Error analysis is challenging in this task. A model may fail on certain datasets for many reasons. Our top performing model and the test dataset indicate the model's inaccurate classification of certain texts as direct violent or passive violent, and vice versa. Disparities in dataset labeling are a big issue. Why certain texts are labeled "2" for direct violence and others "1" for passive-violent texts is unclear. For instance, "2" is placed next to Figure 3-a and "1" is placed next to Figure 3-b. Religious biases of annotators should also be considered. This prejudice is evident when some texts are termed passive-violent and others comparable to them as non-violent or directly violent. Additionally, single-word messages like Figure 3-c are arbitrarily allocated the label "1" creating ambiguity. The inaccurate classification of shorter texts is due to lack of context. The model reveals classification accuracy of longer texts differ from shorter ones. The model's emphasis on the words of a sentence may explain this discrepancy. Longer sentences strengthen the model's contextual foundation, enabling more exact classification. After thoroughly studying the test set, we observed 472 label discrepancies between test set labels and best model predictions. Our model identified 207 of these texts as non-violent (label 0), while the test set classified them as passive-violence (label 1). The second greatest label differences was 91 instances between the test set's identification of texts as non-violent (label 0) and our model's labeling as Passive Violence (label 1). More than 50% of the mistakenly predicted classifications are Non-Violence and Passive Violence. This gap may be due to subtle distinctions between indirect Passive Violence and Non-Violence sentences. Besides, back-translation data augmentation improved model performance, but it might alter text meaning and structure, therefore NLP tasks should be used with caution. It is important to evaluate this potential impact on augmented data quality.



a. "তোদের উপর আল্লাহর গজব পড়বে"
b. "আল্লাহ্ তুমি এদের উপর গজব নাজিল করো"
c. "দালাল"

Figure 3: Examples of texts from train dataset about ambiguous labeling

7 Conclusion and Future Work

The objective of this research was to classify texts into three groups and determine whether or not they promote violence in any way. We have experimented with some prominent transformer based models for text classification before trying out other approaches to make those models perform better. After the test set was made public, we were able to strengthen the performance of our model by running further tests. In order to accurately identify violent texts in social media comments, there is still work to be done in the future. It is necessary to conduct more and more experiments with low resource languages like Bangla. We think that our efforts prepared the groundwork for this to happen.

References

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2020. Hatebert: Retraining bert for abusive language detection in english. *arXiv preprint arXiv:2010.12472*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Avishek Das, Mohammed Moshiul Hoque, Omar Sharif, M. Ali Akber Dewan, and Nazmul Siddique. 2023. Temox: Classification of textual emotion using ensemble of transformers. *IEEE Access*, 11:109803–109818.

Mithun Das, Somnath Banerjee, Punyajoy Saha, and Animesh Mukherjee. 2022a. Hate speech and offensive language detection in bengali. *arXiv preprint arXiv:2210.03479*.

Rajesh Kumar Das, Samrina Sarkar Sammi, Khadijatul Kobra, Moshfiqur Rahman Ajmain, Sharun Akter khushbu, and Sheak Rashed Haider Noori. 2022b. Analysis of bangla transformation of sentences using machine learning. In *International Conference on Deep Learning, Artificial Intelligence and Robotics*, pages 36–52. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

U Dikwatta and TGI Fernando. 2019. Violence detection in social media-review. *Vidyodaya Journal of Science*, 22(2).

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Xin Dong and Gerard de Melo. 2019. A robust self-learning framework for cross-lingual text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6306–6310, Hong Kong, China. Association for Computational Linguistics.

Md Imdadul Haque Emon, Khondoker Nazia Iqbal, Md Humaion Kabir Mehedi, Mohammed Julfikar Ali Mahbub, and Annajiat Alim Rasel. 2022. Detection of bangla hate comments and cyberbullying in social media using nlp and transformer models. In *International Conference on Advances in Computing and Data Sciences*, pages 86–96. Springer.

Md Saroar Jahan, Mainul Haque, Nabil Arhab, and Mourad Oussalah. 2022. Banglahatebert: Bert for abusive language detection in bengali. In *Proceedings of the Second International Workshop on Resources and Techniques for User Information in Abusive Language Analysis*, pages 8–15.

Md Saroar Jahan and Mourad Oussalah. 2023a. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, 546:126232.

Md Saroar Jahan and Mourad Oussalah. 2023b. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, page 126232.

Md. Rezaul Karim, Bharathi Raja Chakravarti, John P. McCrae, and Michael Cochez. 2020. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. In *7th IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA,2020)*. IEEE.

Md Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Sagor Sarker, Mehadi Hasan Menon, Kabir Hossain, Md Azam Hossain, and Stefan Decker. 2021. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE.

Ashfia Jannat Keya, Md. Mohsin Kabir, Nusrat Jahan Shammey, M. F. Mridha, Md. Rashedul Islam, and Yutaka Watanobe. 2023. G-bert: An efficient method for identifying hate speech in bengali texts on social media. *IEEE Access*, 11:79697–79709.

Versha Kumari, Khuhed Memon, Burhan Aslam, and Bhawani Shankar Chowdhry. 2023. An effective approach for violence detection using deep learning and natural language processing. In *2023 7th International Multi-Topic ICT Conference (IMTIC)*, pages 1–8.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '19, page 173–182, New York, NY, USA. Association for Computing Machinery.

M. F. Mridha, Md. Anwar Hussen Wadud, Md. Abdul Hamid, Muhammad Mostafa Monowar, M. Abdullah-Al-Wadud, and Atif Alamri. 2021. L-boost: Identifying offensive texts from social media post in bengali. *IEEE Access*, 9:164681–164699.

Julian Risch and Ralf Krestel. 2020. Bagging BERT models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 55–61, Marseille, France. European Language Resources Association (ELRA).

Nauros Romim, Mosahed Ahmed, Md Saiful Islam, Arnab Sen Sharma, Hriteshwar Talukder, and Mohammad Ruhul Amin. 2022. Bd-shs: A benchmark dataset for learning to detect online bangla hate speech in different social contexts. *arXiv preprint arXiv:2206.00372*.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Mohamed Rahouti, Nabeel Mohammed, and Mohammad Ruhul Amin. 2023a. Blp-2023 task 1: Violence inciting text detection (vitd). In *Proceedings of the 1st International Workshop on Bangla Language*

Processing (BLP-2023), Singapore. Association for Computational Linguistics.

Sourav Saha, Jahedul Alam Junaed, Maryam Saleki, Arnab Sen Sharma, Mohammad Rashidujjaman Rifat, Mohamed Rahout, Syed Ishtiaque Ahmed, Nabeel Mohammad, and Mohammad Ruhul Amin. 2023b. Vio-lens: A novel dataset of annotated social network posts leading to different forms of communal violence and its evaluation. In *Proceedings of the 1st International Workshop on Bangla Language Processing (BLP-2023)*, Singapore. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2022. M-bad: A multilabel dataset for detecting aggressive texts and their targets. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 75–85.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.

A Appendices

A.1 Dataset Description

Here, we illustrate frequency plots for the train-set, dev-set, and test-set's three different classes as well as wordclouds that indicate various texts that incite violence for the three classes.

The frequency distribution displayed in Figure 4 shows that non-violent classes are more frequently reported than passive and direct forms of violence. This illustration makes it clear that the non-violent text class dominates, skewing the dataset in that direction. The labels 0, 1, and 2 stand for the three types of violence: Direct, Passive, and Non-Violence, respectively.

Figures 5, 6, and 7 show wordcloud where we can see words that are primarily responsible for inciting violence or Non-Violence in the text.

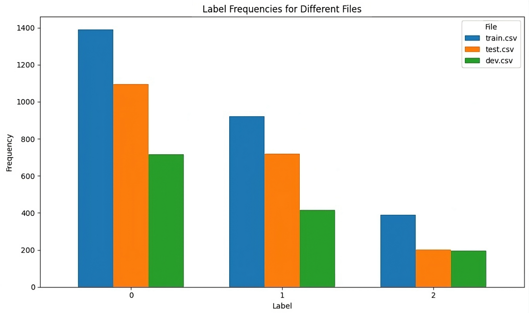


Figure 4: Label Frequency Distribution Across Different Dataset splits

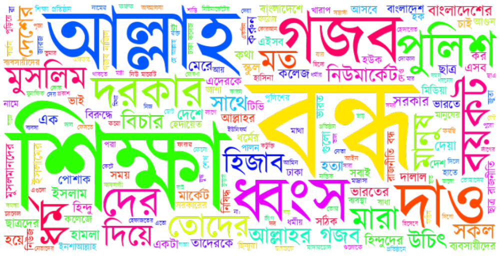


Figure 5: Distinctive Language Patterns in Direct Violence Category

A.2 Experimental Results

Table 3 describes different model names denoted as numbers from 1 to 13 with their experimental approach.

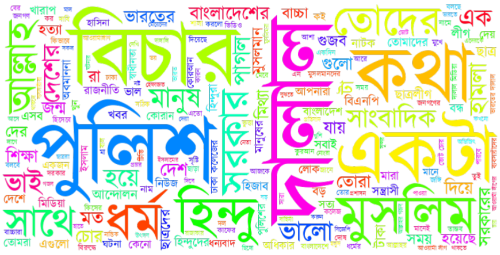


Figure 6: Distinctive Language Patterns in Passive Violence Category

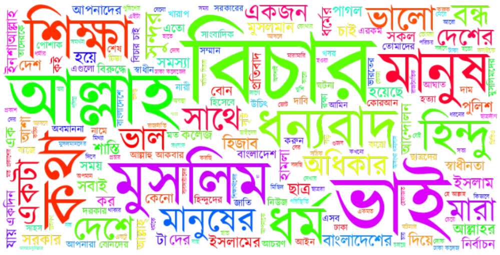


Figure 7: Distinctive Language Patterns in Non-Violence Category

Model Name	Approach
(1)	BanglaBert
(2)	BanglaBert using self learning on (1)
(3)	BanglaBert trained on subset of train + dev sets
(4)	BanglaBert trained on subset of train + dev sets
(5)	BanglaBert trained on subset of train + dev sets
(6)	BanglaBert trained on subset of train + dev sets
(7)	BanglaBert trained on subset of train + dev sets
(8)	BanglaBert trained on train + back_translated_train
(9)	BanglaBert trained on train + back_translated_train + back_translated_dev
(10)	BanglaBert trained on train + back_translated_train + predicted_test_on_best_model_during_competition
(11)	BanglaBert trained on train + back_translated_train + back_translated_dev + external_data + back_translated_external_data
(12)	BanglaBert trained on train + back_translated_train + back_translated_dev + external_data
(13)	BanglaBert trained on train + external_data

Table 3: Approaches of Different Models