Final Report

ডীপ লার্নিং এর সাহায্যে সামাজিক যোগাযোগ মাধ্যমে আক্রমণাত্মক বাংলা টেক্সট শনাক্তকরন ও শ্রেনীবিন্যাসকরন

ConvLSTM: A Deep Learning Approach for Classifying Aggressive Bengali Texts in Social Media

Submitted by

Professor Dr. M. Moshiul Hoque
Project Leader

Natural Language Processing Lab
Department of Computer Science and Engineering (CSE)
Chittagong University of Engineering & Technology (CUET)
Chattogram- 4349, Bangladesh

Acknowledgment

This work was supported by ICT Innovation Fund, ICT Division, Ministry of Posts, Telecommunications and Information Technology, Bangladesh. I am also thankful to all the members of CUET NLP Lab, faculty members and staff of the department of CSE, CUET for their valuable support.

Abstract

The pervasiveness of aggressive content in social media has become a serious concern for government organizations and tech companies because of its pernicious societal effects. In recent years, social media has been repeatedly used as a tool to incite communal aggression, spread distorted propaganda, damage social harmony and demean the identity of individuals or a community in the public spaces. Therefore, restraining the proliferation of aggressive content and detecting them has become an urgent duty. Studies of the identification of aggressive content have mostly been done for English and other resource-high languages. Automatic systems developed for those languages can not accurately identify detrimental contents written in regional languages like Bengali. To compensate this insufficiency, this work presents a novel Bengali aggressive text dataset (called 'BAD') with two-level annotation. In level-A, 14158 texts are labeled as either aggressive or non-aggressive. While in level-B, 6807 aggressive texts are categorized into religious, political, verbal and gendered aggression classes each having 2217, 2085, 2043 and 462 texts respectively. This thesis proposes a weighted ensemble technique including m-BERT, distil-BERT, Bangla-BERT and XLM-R as the base classifiers to identify and classify the aggressive texts in Bengali. The proposed model can readdress the softmax probabilities of the participating classifiers depending on their primary outcomes. This weighting technique has enabled the model to outdoes the simple average ensemble and all other machine learning (ML), deep learning (DL) baselines. It has acquired the highest weighted f_1-score of 93.43% in the identification task and 93.11% in the categorization task.

Keywords: Natural language processing, Aggressive text classification, Low resource language, Bengali aggressive text corpus, Deep learning, Transformer based models, Ensemble classifiers

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The phenomenal proliferation of social media platforms (i.e. Facebook, Twitter, YouTube) has dramatically transformed people's communication mode. These platforms have become the potential medium to express people's opinions on various topics such as politics, religion, finance, sports and other societal events. Information shared on social media platforms has the power to reach millions within a short period. This rapid growth of information has not only resulted in a positive exchange of information, but also allow a group of malign users to disseminate aggressive, offensive, hatred, and other illegal contents. In this work, our primary goal is to develop a computational system that can identify aggression and also detect the fine-grained domain of aggression written in Bengali. This work employed automatic text classification system to attain this goal.

1.1 Text Classification

Digitization has changed the way we process and analyze information. There is an exponential increase in online availability of information. From web pages to emails, science journals, e-books, learning content, news and social media are all full of textual data. Text classification performs an essential role in various applications that deals with organizing, classifying, searching and concisely representing a significant amount of information. Different approaches are used for classifying text documents, such as manual categorization and automatic categorization. Because of the rapid growth of online information, text classification has become more challenging. Hence manual categorization is not feasible because it requires more efforts and time. A feasible solution of this problem is automatic categorization where a machine learning based model is developed to automatically categorize the text documents. This automatic text classification method can also be applied to detect potential aggressive texts and classify them.

1.1.1 Aggressive Text Classification

Aggressive text classification is the task of assigning potential texts into predefined aggression categories such as religious, political, verbal and gendered aggression etc. Over the last few years, several studies have been carried out to develop an automatic and semi-automatic system to tackle the spread of undesired (aggressive, abusive, offensive) contents on online platforms [1, 2]. Most of the previous researches can be described with two categories.

- **Supervised Aggressive Text Classification:** in supervised classification categories are predefined. It works on training and testing principle. During training phase, the learning algorithm works on the labeled data. The algorithms are trained on labeled data set and gives the desired output. During testing phase, unobserved data are fed into algorithm and classifier classifies them based on the knowledge of training phase.

- **Unsupervised Aggressive Text Classification:** in unsupervised classification categories are not defined. Here learning algorithms try to discover natural structure in data. The algorithm looks for similar patterns and structures in the data points and groups them into clusters. The classification of the data is done based on the clusters formed. Since, aggression is a very subjective phenomena this work primarily explore supervised classification technique to accomplish the task.

1.2 Motivation

Past surveys reported that social media platforms had been utilized to publicize aggression, incite political and religious violence that jeopardize communal harmony and social stability [3]. The viciousness of aggressive and offensive texts is strong enough to trigger massive violence, create mental health problems or even instigate suicide [4, 5]. Therefore, it is monumental to develop resources and methods to flag such contents for reducing unlawful activities and keep the information ecosystem clean from polluted contents.

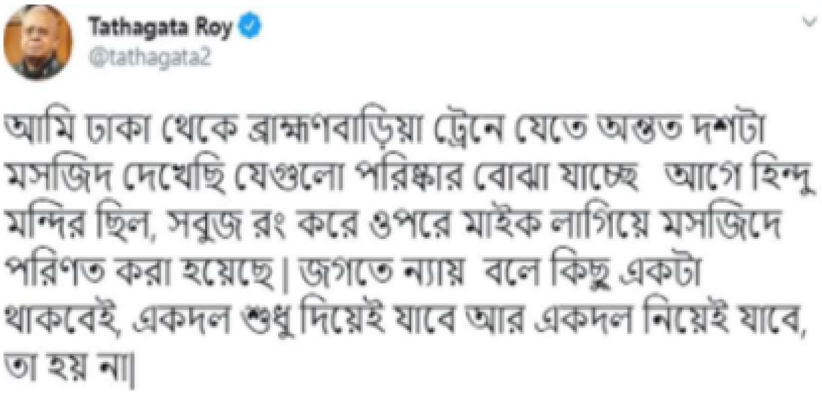
Unfortunately, despite being the seventh most widely spoken language globally, Bengali is considered one of the notable resource-constrained languages [6]. Statistics reveal that more than 45 million users on Facebook and YouTube are using Bengali daily. Most of these users commonly interact on social media via the textual form. Many textual interactions contain hostile contents that cause the significant rise of hate, abuse, cyberbullying and aggression on social media.

Thus, to ensure the quality of textual conversation and reduce unlawful activities on these platforms, developing an automated Bengali language system that can identify these aggressive activities is mandatory. Our major motivation to work in this area are,

- To develop a system which will flag posts/comments that convey any aggressiveness that might threaten national security, try to break communal harmony, and publicize distorted propaganda.
- If we able to detect aggressive texts and identify fine-grained domain of aggression it will help our law enforcement agencies to find the perpetrator and stop the undesired events.
- Such a system will ensure the quality of conversations in social spaces.
- Develop models and resources for research community and language technology-related industries who work on Bangla language processing (BLP).

1.3 Importance

Due to the substantial growth of internet users, the amount of social media contents (in Bengali) is growing enormously in recent years. Unfortunately, the misapplication of technologies has boosted with this rapid growth of social media contents which lead to the rise in aggressive activities. A group of malign users are repeatedly use social medias to incite violence.



Tathagata Roy
@tathagata2

আমি ঢাকা থেকে ব্রাহ্মণবাড়িয়া ট্রেনে যেতে অন্তত দশটা মসজিদ দেখেছি যেগুলো পরিষ্কার বোঝা যাচ্ছে আগে হিন্দু মন্দির ছিল, সবুজ রং করে ওপরে মাইক লাগিয়ে মসজিদে পরিণত করা হয়েছে। জগতে ন্যায় বলে কিছু একটা থাকবেই, একদল শুধু দিয়েই যাবে আর একদল নিয়েই যাবে, তা হয় না।

Figure 1.1: Example of a religiously aggressive post

Consider the example illustrated in figure 1.1. In this post the user directly invoking the Muslim communities against Hindus. Such a post might be a potential threat which might break communal harmony since posts shared on social media has the power to reach millions within a short time. Therefore monitoring and discarding such posts is really important to ensure social harmony. Moreover, a large amount of underprivileged women and child are victimizing by this aggressive or abusive activities online. Sufferer disadvantaged women might enjoy immediate remedy from law & enforcement/security agencies if we had a system that can identify aggressiveness by analyzing the online contents. However, it is a quite impossible task to detect aggressive texts from the enormous amount of internet text contents manually; therefore, the automatic detection of aggressive text contents should be developed. Accordingly, the automatic detection of aggressive text contents should be designed. Responsible agencies are demanding some smart tool/ system that can detect aggressive text automatically. It will also be helpful to identify potential threats in cyber-world which are communicated by text contents. Automatic detection of aggressive text system can effortlessly and promptly detect the fishy or threatening texts. Law and enforcement authority can take appropriate measures immediately, which in turn helps to reduce virtual harassment, suspicious and criminal activities mediated through online. Some important real-world implications of this thesis are illustrated in the following,

- An automated aggressive text classification system will eliminate the process of manually checking the whole conversation, which is tiresome as well as time-consuming. This system will surely help our security agencies to find the perpetrator and his/her aggressive write up in social media within a short period.

- An application can be developed along with a controlled environment where we can analyse a textual communication thread. By analysing the communication, the proposed system will be able to predict whether a conversation conveys any aggressiveness which might pose a threat for our national security, tires to break communal harmony, publicize distorted propaganda and excite any specific group of people.

- The system proposed in this work can classify aggressive Bengali text which can be useful for law enforcement/security agencies to identify potential perpetrators in any violation of digital security act-2018.

1.4 Challenges

Till to date, most of the resources/tools have been developed for languages like English, Chinese, Arabic and other European languages [7, 8]. Nevertheless, people usually interact via their regional language to carry out day-to-day communication. System trained in resource-rich languages can not be directly replicated to detect aggressive/abusive texts written in the local language. Therefore, it is a prerequisite to develop resources, techniques, and regional language tools to reduce the effect of undesired texts. However, developing a system to detect aggressive textual conversation in a resource-constrained language like Bengali is challenging. The key barriers to work on this problem are illustrated in the following,

- **Unavailability of Dataset:** The scarcity of benchmark dataset and deficiency of language processing tools are the major barriers to develop such a system in Bengali.
- **Unique Characteristics of Bengali Language:** Complicated morphological structure, presence of ambiguous words, diversities in different dialects and rich variations in the constituent parts of a sentence have made the task more complicated. Bengali has a rich vocabulary and unique writing script which has no overlap with other resource-high languages.
- **Code-mixing:** Multilingual code-mixing in social media texts has added a new challenge to the existing task [9].
- **Characteristics Overlap:** The overlapping characteristics of aggression with other corelated phenomena such as hate, abuse, offense has made the task more challenging.

Finally, model on resource-rich languages can not be directly used without proper modifications. We have to change the network architecture and tune hyperparameters (learning rate, dropout rate, optimizer etc.) that best suited our dataset.

1.5 Contributions

This work develops a Bengali aggressive text dataset by analyzing aggressive and non-aggressive texts' properties to address the above research questions. Various aspects of the dataset are also explained to get better insights. Several machine learning (ML), deep learning (DL) and transformer-based techniques are investigated to build the aggressive text identification and classification system. Exploring the models' outcomes, this work proposes a weighted ensemble technique that exploits the best performing models' strength. Finally, we investigate the proposed model's results and errors and compare it with other existing techniques. Major contributions of this work can be illustrated in the following:

- **Dataset:** present a new Bengali aggressive text dataset which contains 6807 aggressive and 7351 non-aggressive texts. Furthermore, by employing a hierarchical annotation schema, aggressive texts are annotated into religious, political, verbal and gendered aggression classes.
- **Insights:** provide useful insights and detailed statistics of the data that ensure the quality of the dataset.
- **Model:** develop a weighted ensemble model using m-BERT, distil-BERT, Bangla-BERT, XLM-R to identify and categorize aggressive Bengali texts. The proposed model emphasizes the participating classifiers' softmax probabilities based on their previous performance on the dataset. This weighting technique outperforms the simple average ensemble approach and enhances the classifier performance in the developed dataset.
- **Benchmarking:** investigate and compare the performance of the proposed model with other ML, DL baselines and existing techniques, thus setting up a benchmark work to compare in the future.
- **Error analysis:** deeply analyze the results and errors of the proposed model. Presents qualitative and quantitative analysis that shed light on the reasons behind some of the errors and provide a few directions that might help to mitigate the system's deficiency.

This research is one of the pioneering works that aims to identify and classify aggressive texts in Bengali as per our exploration. We expect that the resources developed in this work will pave the way for aggressive text classification researchers in Bengali.

1.6 Organization of Thesis

The remaining of the thesis is organized as follows. Chapter 2 discusses the studies related to unwanted text detection and classification on online platforms. The detailed definition of aggressive text and its categories included in Chapter 3. It also presents dataset development steps, analysis and statistics of the developed dataset. Chapter 4 illustrates the techniques adopted to develop the proposed system. Experimental findings with quantitative and qualitative error analysis are reported in Chapter 5. Chapter 6 points out the future scopes with concluding remarks.

Chapter 2

Literature Review

Over the last few years, a significant amount of work has been carried out to identify and categorize unwanted texts on various online platforms such as twitter, facebook, reddit and so on. Works included aggression classification [10, 11, 12], hate speech detection [13, 14, 15], abuse detection [16, 17, 18], toxicity classification [19, 20], misogyny [21, 22], trolling identification [23, 24], cyberbullying detection [25, 26], and offensive text classification [27, 28, 29]. Although most of the researches focused in English however a considerable body of work has been conducted for other languages too. This chapter briefly describes the researches related to aggression, hate, offense detection/classification with other co-related phenomena concerning non-Bengali and Bengali languages.

2.1 Non-Bengali Language Based Undesired Text Classification

This section gives a brief overview of the researches done in English, Hindi, Arabic and other languages. Kumar et al. [10] present an aggressive language identification dataset that has three category: *overt, covert, non-aggressive*. The Dataset contains 15k aggression annotated comments/posts written in English and Hindi. Aroyehun et al. [30] develop deep neural network-based models on English with data augmentation and pseudo labelling strategy. Their system achieved macro f_1-score of 0.64 and 0.59 using LSTM and CNN-LSTM methods, respectively. Risch et al. [31] employ bootstrap aggregating based ensemble with multiple fine-tuned BERT on TRAC-2 [32] dataset to identify aggression and misogyny. They obtain an 80.3% weighted f_1-score on the test set of English social media posts. Zampieri et al. [33] compile an offensive language identification dataset *(OLID)* of 14k English tweets. They used a three-layer hierarchical annotation schema to detect, categorize and identify the target of texts whether it attack individuals or a group of people. Baseline evaluation is performed using SVM, BiLSTM and CNN. In all three levels, CNN outperforms others by achieving macro-f_1 of 0.80, 0.69 and 0.47. Fortuna et al. [34] offer a dataset of 80000 tweets labelled with four categories: *hateful, abusive, spam and normal.* They performed a holistic approach to identify confusion among various categories. Davidson et al. [35] develop a hate speech dataset of 25k tweets with three categories: *hate, offence and neither.* Logistic regression with tf-idf and n-gram features obtains the best macro f_1-score of 0.90.

Mathur et al. [36] introduce a Hindi-English code switched dataset of 3.6k tweets split into three categories: *abuse, hate speech, and non-offensive.* They proposed a system based on CNN and transfer learning which achieves f_1-score of 71.4%. Aggression annotated Hindi-English code mixed Dataset of 21k Facebook comments and 18k tweets are developed by Kumar et al. [37]. Instances were labelled with three top-level tags and ten discursive classes. Annotation performed by four annotators where the inter-annotator agreements were 72% and 57% on the top-level and 10 class annotations. Bhardwaj et al. [38] presented a multi-label hostility detection dataset of 8.2k online posts in Hindi. Dataset divided into five dimensions: *fake, hate, offensive, defamation, and non-hostile* where SVM achieved the highest weighted f_1-score of 84.11% with m-BERT embedding.

Mulki et al. [39] built a dataset of 5.8k tweets in the Arabic Levantine dialect and manually annotated tweets into *abusive, hate or normal* classes. Their system achieved f_1-score of 89.6% in binary *(abusive or normal)* and 74.4% in ternary *(abusive, hate or normal)* classification scenario using naive Bayes (NB). Mubarak et al. [40] present a dataset to classify Arabic tweets into *offensive, obscene or clean* categories. Their system obtains a maximum of f_1-score of 0.60 with the combination of seed words and unigram features. Hassan et al. [41] employ ensemble technique over SVM, CNN-BiLSTM and m-BERT to identify offensive Arabic texts in OffensEval-2020 dataset. They used character n-grams, word n-grams, character and pre-trained word embedding features. The system acquired of 90.16% f_1-score on the Arabic test set.

Carmona et al. [42] manually annotated 11k Mexican Spanish tweets into *aggressive* and *non-aggressive* classes. They organized a task over this dataset and lexicon-based approach [43] obtained the best performamnce with a macro f_1-score of 0.62. Few tasks organized at GermEval [44, 45] aimed to classify German tweets into *offensive* and *non-offensive* classes. The top performance achieved f_1-score of 76.77% using feature ensemble method on a dataset of 8.5k German tweets.

Leite et al. [46] presented a toxic language dataset (ToLD-Br) composed of 21k tweets. They manually annotated tweets into seven classes: *LGBTQ+phobia, racism, insult, xenophobia, obscene, misogyny, and non-toxic*. Their system obtained macro-f_1 of 76% using BERT models. An offensive dataset consisting of 1250 comments is developed by Pelle et al. [47]. They split offensive texts into six fine-grained labels: *racism, sexism, xenophobia, LGBTQ+phobia, cursing, and religious intolerance*. N-gram features with SVM achieved the best f_1-score ranging from 77% to 82%. Fortuna et al. [48] presented a Portuguese hate speech dataset consisting of 5668 tweets. The Dataset was labelled into binary *(hate or non-hate)* and 81 hierarchical categories. LSTM with pre-trained word embedding acquires 78% f_1-score on the binary labels.

In recent years, a series of the shared task and academic events have organized, focusing on multilingual identification and classification of aggressive, abusive, offensive and hatred contents in social media. Shared task on trolling, aggression and cyberbullying (TRAC-1 [10]) aims to classify English and Hindi texts into overtly, covertly and non-aggressive classes. In the second iteration (TRAC-2 [32]), they added Bengali texts with an additional task of identifying gendered aggression. The best outcome was achieved with variants of transformer models [49, 50]. OffensEval-2020 [51] provided manually annotated offensive texts in five different languages *(English, Arabic, Turkish, Greek and Danish)* that follow the hierarchical annotation schema of 'OLID'. The top system of all the languages have employed ensemble technique with fine-tuned transformers [52, 53]. HASOC-2020 [15] offered hate and offensive language dataset in *Tamil, Malayalam, Hindi, English and German* to perform two tasks. At first, identify hate or offensive posts and further categorize them into *hate, offense* and *profane* classes. The best system for Hindi, German and English achieved 0.53, 0.52 and 0.51 macro f_1-score, respectively. Other notable works included *Automatic Misogyny Identification* [54], *Workshop on Abusive Language* [55] and *HatEval* [56] that investigated hate speech against women and immigrants in Spanish and English.

2.2 Related Researches in Bengali

Identification and categorization of aggressive texts in Bengali is an open avenue for future research. Due to the scarcity of benchmark dataset, linguistic tools and other resources, no significant works have been carried out to date in this arena. However, with multi-lingual and cross-lingual models' arrival, few works have been conducted recently related to the detection/classification of hate, ag-

Table 2.1: Brief literature summary of the researches concerning undesired text classification in Bengali and non-Bengali languages.

Article	Language	Approach	Limitation/Gap
Zampieri et al. [33]	English tweets	CNN, SVM, BiLSTM	Model biased towards not offensive class. Performance degrades as the number of class increases
Gamback et al. [57]	English tweets	CNN with word embedding and character ngrams	Incapable of capturing sequential features as recurrent networks are not used
Tulkens et al. [58]	Dutch posts	SVM with dictionary based features	Failed to capture the context
Mihaylov et al. [23]	English trolls	Statistical features applied on SVM with RBF kernel	Content features (keywords, named entities, topics) and other ML methods are not considered
Andrew et al. [59]	Tamil, Malayalam and Kannada posts	Tf-idf features employed on set of baseline machine learning classifiers	Used only tf-idf features and no counter measures is taken to handle code-mixing of texts
Bhardwaj et al. [38]	Hindi comments	mBERT embedding employed on set of ML classifers	Ignored the sequential information and limited number of training texts in fine-grained classes
Karim et al. [60]	Bengali social media texts	ensemble of BERT based models	Did not consider the overlap among classes
Romim et al. [61]	Bengali facebook posts	SVM	Can not capture the semantic features
Sharif et al. [62]	Bengali facebook and youtube comments	Logistic regression	Limited dataset and ignored semantic features
Emon et al. [63]	Bengali	Bidirectional LSTM	Do not employ the state of the art models
Mridha et al. [64]	Bengali texts	Combination of BERT and LSTM based models	Ignored code-mixing and code-switching phenomena

gression, offence, and abuse. Ranasinghe et al. [65] developed a model to classify aggressive Bengali texts into overtly, covertly and non-aggressive classes. They used 4k texts from the Bengali dataset presented in the TRAC-2 shared task [32]. Their system achieved the highest weighted f_1-score of 84.23% by leveraging inter-language transfer strategy with XLM-R. Karim et al. [60] collected 3k Bengali text samples and categorized them into four hatred classes: *political, personal, geopolitical, religious*. They used an ensemble of BERT variants to develop their system and obtained a 0.88 f_1 score. The class definitions provided by the authors may result in contradiction. An instance may be expressed political and religious hate simultaneously. No insight on the countermeasure is provided during such situations. Romim et al. [61] presented a hate speech dataset which contains 30k with *hate* or *non-hate* comments crawled form Facebook and YouTube. The baseline system obtained 87.5% f_1-score using SVM. A recent work [62] presented a logistic regression-based model to classify *suspicious* and *non-suspicious* Bengali texts. Five different ML algorithms are applied on the extended Dataset of 7k texts [66]. SGD classifier with tf-idf features obtains the best accuracy of 84.57%. Emon et al. [63] develop an abusive Bengali dataset of 4.7k texts consisting of seven classes (*slang, religious-hatred, political-hatred, personal attack, anti-feminism, neutral, positive*). The model gained 82.2% accuracy by utilizing LSTM. An SVM based system is developed to identify the threat and abuse from Bengali texts [67], which achieved an accuracy of 78% on a dataset of 5644 texts. In our previous work, we develop an aggressive text identification and classification dataset of 7591 texts where combined CNN, BiLSTM methods obtained the highest weighted f_1-score [68]. Here, we perform experimentation with a wide range of methods on the extension of the existing dataset.

2.3 Research Gap

Availability of a standard dataset is the prerequisite to develop any classification system. Previous research in Bengali mainly focuses on classifying hatred and abusive contents using ML and other feature-based methods. None of the research has been conducted to identify aggression and categorize aggressive texts into fine-grained classes in Bengali. Therefore, to perform the aggressive text identification and classification in Bengali, we need to develop a dataset using a hierarchical annotation schema. Moreover, Computational systems developed over other languages and datasets can not be replicated directly on a new dataset. The main reason is that models available in one language would not be able to capture the features in another language without proper modifications in the model architecture (i.e., no. of layers, no. of neurons, no. of filters) and fine-tuning of hyperparameters (i.e., learning rate, batch size, dropout rate, epochs, optimizer). Therefore, we need to design a system from scratch to perform the aggressive text identification and classification tasks in Bengali. Therefore, the key research questions we are investigating in this work are-,

- "**RQ1:** How can we successfully develop an aggression annotated dataset in the Bengali language?"
- "**RQ2:** How can we effectively identify potential aggressive texts and categorize them into predefined aggression categories?"

In the subsequent subsections we comprehensively discuss the way of adressing these research questions.

Chapter 3

BAD: Bengali Aggressive Text Dataset

The detailed definition of aggressive text and its categories, dataset development steps and analysis of the dataset presented in this chapter. This work aims to develop an aggressive text identification and classification system that can detect whether a potential text $t_i \epsilon T$ is aggressive or not from a set of m texts, $T = \{t_1, t_2, ..., t_m\}$ in the first phase. In the next phase, the system categorizes the aggressive texts into one of n predefined aggression classes, $AC = \{ac_1, ac_2, ..., ac_n\}$. The task of the system is to assign at_i automatically to ac_j where at_i and ac_j represents the aggressive text and aggressive class, respectively.

3.1 Definition of the Task

In order to accomplish the task, dataset is split into two levels using hierarchical annotation schema [33]: (A) coarse-grained identification of aggressive texts (B) fine-grained categorization of aggressive texts. This section defines the aggressive texts and their fine-grained classes to perform the tasks mentioned above.

3.1.1 Level A: Aggressive Text Identification

Determining whether a text is aggressive or not aggressive is very ticklish, even for language and psychology experts due to its subjective nature. People may define aggression in different ways, which leads to the heterogeneous interpretation of aggression. One person may contemplate a piece of text as aggressive, while another may consider it as usual. Moreover, overlapping characteristics of aggression with hate speech, cyber-bullying, abusive, offence and profanity have

Table 3.1: Definitions of aggression, incitement, violent and hatred contents according to different scientific studies, human rights organizations and various social networking sites.

Source	**Definition**
Anderson et al. [69]	"Language that used toward other individuals with the intent to cause harm".
Facebook [?]	"Contents that attack or pose credible threats to personal or public safety, facilitate high severity violence, misinformation and unverifiable rumours that contribute to risk of imminent violence".
Torres et al. [70]	"Aggressive language intents to hurt or harm an individual or a group by referring to or exciting violence".
Nobata et al. [71]	"Language which attacks or demeans a group based on race, ethnic origin, religion, gender, age, disability, or sexual orientation/gender identity".
YouTube [72]	"Contents that promote violence or hatred against individual or groups, based on age, sexual orientation, religion, disability, nationality etc".
Council of Europe (COE) [73]	"Expression which spread, incite, justify or promote violence, hatred and discrimination against a person or group of persons for variety of reasons".
Paula et al. [74]	"Language that glorify violence and hate, incite people against groups based on religion, ethnic or national origin, physical appearance, gender identity or other".
Roy et al. [75]	"Aggressive language directly attack group or person using abusive words, comparing in a derogatory manner or support false attack toward others".

made this task more complicated and challenging. Understanding the phenomena of aggression in a better way requires a large amount of literature study in aggression and impoliteness from psychological and linguistic perspectives. However, this task's aim is much simpler, and this work performs a surface level classification of aggressive text on social media. Thus, it is monumental to define aggressive text first to implement the aggressive text classification system successfully. To do this, several pieces of literature have been explored to interpret the aggression, incitement, violence, suspicion, and hatred contents from different sources. Table 3.1 presents a summary of the definitions culled from various trending social networking sites, human rights organizations, psychological and scientific studies.

Baron et al. [76] defined aggression as a behaviour that expresses the desire to harm another individual verbally, physically, and psychologically. The distinction between physical, verbal, and relational aggression exhibited by Buss et al. [77]. Kumar et al. [37] discriminated overtly and covertly aggressive texts. In overtly aggressive texts, aggression expressed directly with the strong verbal attack. While covertly aggressive texts attack the victim in rhetorical queries, satire, metaphorical reference, and sarcasm. The majority of these statements provide the broader prospect of aggression from images, videos, texts and illustrations. However, this work focuses on detecting and classifying aggression from textual contents only. Analyzing the interpretation of aggression and exploration of literature lead us to distinguish between aggressive and non-aggressive texts as follows:

- Aggressive texts **(AG)**: Text contents that incite, attack, or wish to harm an individual, group or community based on some criteria such as religious belief, gender, sexual orientation, political ideology, race, nationality and ethnicity.
- Non-aggressive texts **(NoAG)**: Text contents that do not contain any statement of aggression or express hidden intention to harm an individual, group or society.

3.1.2 Level B: Fine-grained Categorization

In recent years, the thriving interest in aggression/abuse from various perspectives have created a conglomeration of typologies and terminologies. Few works attempted to provide a uniform understanding of this complex phenomenon. Waseem et al. [78] proposed two-level categorization of abusive online language: nature of the abuse (implicit or explicit) and the target of the abuse (group or individuals). However, Kumar et al. [10] pointed out that in the majority of the abusive instances, individuals and groups are targeted simultaneously. Therefore, it would not be wise to distinguish between these classes while annotating many instances. The authors suggested that the distinction between various abuse/aggression dimensions can be made considering the attack's locus such as gender, religion, specific ideology, politics, race, and ethnicity [10, 37]. Most previous works in Bengali [60, 63] illustrated that political, gendered, verbal and religious abuse/offence classes are occurred more frequently in Bengali texts than others (such as racial, geographic). Furthermore, our exploration revealed that a higher amount of Bengali texts are available in four coarse categories: political, religious, verbal, and gendered aggression. Therefore, this work also concentrated on these four aggression dimensions due to their much textual contents availability. As these classes interpretation varies considerably across individuals, it is essential to draw a fine line among these aggression categories. In order to minimize the bias as well as overlap during annotation after analyzing existing research on aggression detection [79, 37, 80, 81], toxicity classification [46, 82, 83], hate speech identification [84, 85, 86], abuse detection [87, 88, 89], cyber-bullying categorization [90, 25] and other related terminologies guided us to make a distinction between aggression classes as the following:

- Religious aggression **(ReAG)**: incite violence by attacking religion (Islam, Hindu, Catholic, and Jew), religious organizations, or religious belief of a person or a community.
- Political aggression **(PoAG)**: provoke followers of political parties, condemn political ideology, or excite people in opposition to the state, law or enforcing agencies.
- Verbal aggression **(VeAG)**: damage social identity and status, describe a wish to harm or do evil of the target by using nasty words, curse words and other outrageous languages.
- Gendered aggression **(GeAG)**: promote aggression or attack the victim based on gender, contain an aggressive reference to one's sexual orientation, body parts or sexuality, or other lewd contents.

To the best of our knowledge, no research has been conducted yet classifying aggressive Bengali texts into these fine-grained classes.

3.2 Dataset Development

As per our exploration, none of the datasets on aggressive Bengali text is available that deals with the defined fine-grained class instances. Therefore, we develop a Bengali aggressive text dataset (called 'BAD') to serve our purpose. To develop 'BAD', we have followed the directions given by Vidgen and Derczynski [91]. Figure 3.1 illustrates the data collection and annotation pipeline. The detailed discussion on the dataset development process described in the following subsections.

3.2.1 Data Accumulation

A total of **14443** aggressive and non-aggressive texts are accumulated manually from various social media platforms. Most of the dataset instances are collected

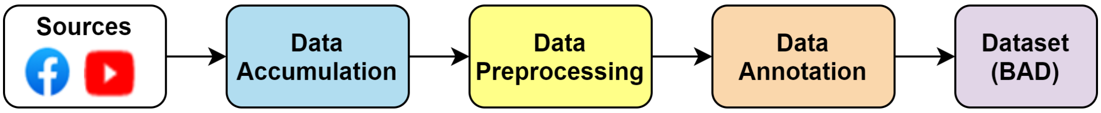


Figure 3.1: Dataset development steps. BAD stands for "Bengali Aggressive Text Dataset".

from Facebook and YouTube since majority of the Bengali social media users are active on these platforms. According to social media stats[1], 94.88% and 2.68% social media users in Bangladesh use Facebook and YouTube. Although the recent statistics exhibited a rise in Twitter users in Bangladesh, the people mostly use English for social communication. Due to the scarcity of Bengali texts related to the aggressive contents, the current work did not consider Twitter data. Dataset utilized in this work was acquired from July 1, 2020, to February 25, 2021. Within this duration, we have considered only those texts that were composed after June 30, 2019. Strategies that followed to collect aggressive and non-aggressive texts illustrated in the following subsections.

3.2.1.1 Aggressive Text Collection

For ***aggressive texts***, the general approach is to collect the posts and comments that incite violence or express aggression. Additionally, we analyze the replays of aggressive posts/comments. In a significant number of cases, we found that to counter an aggressive comment; people use another aggressive comment. Furthermore, to get additional aggressive texts, the user's timeline is scanned who like, share or comment in support of aggression-related posts.

Most religious aggressive data is collected from the comment threads of YouTube channels and Facebook pages concerning religion. The majority of the gendered aggression expressed in social media is against women compared to the male counterpart. Texts related to this category are accumulated from various domains such as fitness videos, fashion pages, and media coverage on celebrities/women. Texts that use curse/outrageous words and wish to do evil to others added into the verbal aggression category. Politically aggressive texts procured from Facebook pages of political parties, pages of their supporters and opposition parties, influential political figures, and people's reaction to the government's different policies.

[1] https://gs.statcounter.com/social-media-stats/all/bangladesh

Table 3.2: Statistics of few sources from where data were gathered. Here FP, YC indicates Facebook page and YouTube channel respectively.

Name	**Type**	**Affiliation**	**Popularity (No. of followers/ subscribers)**	**Reactions per post (in avg.)**	**Activity (frequency of posting)**
Prothom Alo	FP/YC	Newsgroup	15M	5k	200 post/day
Rafiath Mithila	FP	Artist	3M	25k	3 post/week
Mizanur Azhari	YC	Religious speaker	1.67M	30k	1 post/week
Jamuna tv	YC	Media	7.69M	4k	50 post/day
Asif Mohiuddin	FP	Public figure	118k	1.5k	1 post/day
Awami League	FP	Political org.	799k	3.5k	13 post/day
Salman BrownFish	YC/FP	Musician	2.6M	20k	2 post/week
Pinaki Bhattacharya	FP	Author	342k	9k	6 post/day
Somoynews tv	YC/FP	Media	7.8M	1K	150 post/day
Basher kella	FP	Political	42k	300	20 post/day

3.2.1.2 Non-aggressive Text Collection

Non-aggressive texts are cumulated from the news/posts related to science & technology, entertainment, sports and education. The primary sources of these data are Facebook pages and YouTube channels of popular Bangladeshi newspapers (such as Somoy-news, Prothom-Alo, Jamuna-tv). Table 3.2 illustrates the popularity and activity status of these sources. The data was collected only from the Facebook and YouTube pages of the newsgroups. None of the data is accumulated from news portals. Moreover, while procuring aggressive texts, we found plenty of non-aggressive examples and added them into this category.

The potential texts are manually accumulated from more than 100 Bengali Facebook pages and YouTube channels affiliated with media, political organizations, authors, artists, and newsgroups to develop the dataset. Table 3.2 illustrates detailed statistics to understand the quality of the data gathered from Facebook and YouTube platforms. To develop the dataset, we have considered only the public posts/comments from these sources and did not collect user's information. Data shared publicly on Facebook, YouTube can be recorded for research purposes [92] and do not subject to any copyright claim if used for academic research [93]. The source pages or channels might contain personal information; thus, we avoided disclosing the source link. Data were culled from only those threads that received at least 200 reactions (like, comment or share) in total. We did not use any list of keywords or phrases to collect data.

3.2.2 Data Preprocessing

To reduce the annotation effort and remove inconsistencies, few preprocessing filters are applied to the accumulated texts. Steps have followed in processing the texts are,

- All the flawed characters (#@!&%) dispelled from the texts.
- As concise texts do not contain any meaningful information, the text having a length of fewer than three words are discarded.
- Texts written in languages other than Bengali and duplicate texts are removed.

We eliminated **94** texts in this step, and the remaining **14349** texts are passed to the human annotators for manual annotation.

3.2.3 Data annotation

"How to achieve the correct annotation" is one of the most crucial questions to answer when labelling a training dataset [94]. Therefore, to clarify the queries regarding annotation in this part, we recapitulate the annotators' identity, annotation guidelines, and data labelling process that we pursued to develop 'BAD'.

3.2.3.1 Identity of the annotators

Bedner and Friedman [95] emphasize knowing about the identity of the annotators since their perception and experience might influence the annotations. Binns et al. [96] pointed out that in the context of online abuse, the gender of the annotators has an impact on the annotations. Moreover, a homogeneous group of annotators might not capture all the examples of aggression and abuse [97]. To mitigate these issues, we choose annotators from different racial, residential and religious backgrounds. Five annotators carry out manual annotation: two undergraduate, two graduate and one academic expert. Experience, expertise and other relevant demographic information about the annotators are presented in Table 3.3.

Table 3.3: Summary of the demographic information, field of research, research experience and personal experience of aggression in social media of the the annotators. Here AN, OA denotes annotator and online aggression respectively.

	AN-1	**AN-2**	**AN-3**	**AN-4**	**Expert**
Research-status	Undergrad	Undergrad	RA	RA	Professor
Research-field	NLP	NLP	NLP	NLP	NLP, HCI, Robotics
Experience	1 year	1 year	2 years	3 years	20 years
Age	23	23	24	26	46
Religion	Islam	Islam	Hindu	Islam	Islam
Gender	Male	Female	Male	Male	Male
Viewed OA	yes	yes	yes	yes	yes
Targeted by OA	no	yes	no	yes	yes

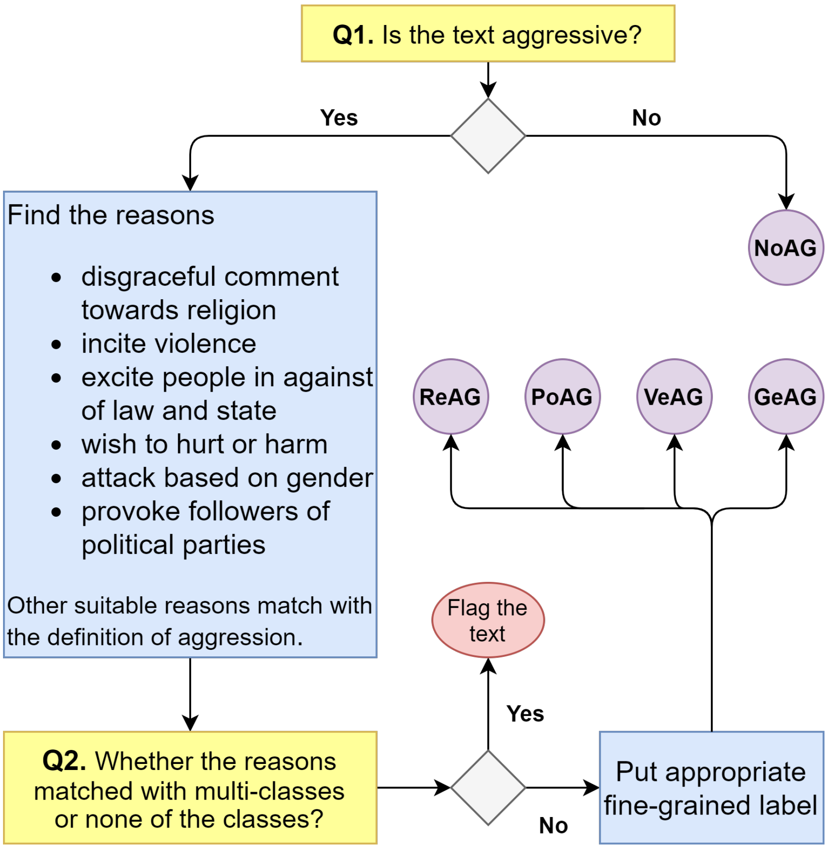

Q1. Is the text aggressive?
Yes
No
Find the reasons
• disgraceful comment towards religion
• incite violence
• excite people in against of law and state
• wish to hurt or harm
• attack based on gender
• provoke followers of political parties
Other suitable reasons match with the definition of aggression.
NoAG
ReAG
PoAG
VeAG
GeAG
Flag the text
Yes
Q2. Whether the reasons matched with multi-classes or none of the classes?
No
Put appropriate fine-grained label

Figure 3.2: Guidelines for data annotation. Reasons denotes a subset of possible reasons.

All of the annotators are native Bengali speakers. Some key characteristics of undergraduate and graduate annotators are: a) age between 22-26 years, b) field of research NLP and experience varies from 1-3 years, c) do not have extreme perspective about religion, d) not a member of any political organization e) active in social media and view aggression in these platforms. Although while selecting the annotators, we tried to keep demographic aspects balanced; however, the annotators pool is still biased in religion (Islam) and gender (Male).

Algorithm 1: Final label assigning process

```
Input: Set of texts with initial labels
Output: Aggressive text dataset with final labels
$T \leftarrow \{t_1, t_2, ..., t_m\}$ (set of accumulated texts);
$BAD \leftarrow []$ (Bengali aggressive text dataset);
$FL \leftarrow []$ (final class labels);
$IL[m][2] \leftarrow \{a_1, a_2, .., a_m\}$ (initial labels);
$D \leftarrow []$ ;
for $t_i \epsilon T$ do
    $l_1 = IL[i][1]$ (first label);
    $l_2 = IL[i][2]$ (second label);
    if ($l_1 == flag$ & $l_2 == flag$) then
        //text is discarded;
    else if ($l_1 == l_2$) then
        $BAD.append(t_i)$ ;
        $FL.append(l_1)$ ;
    else
        $D.append(t_i)$ (disagreement: put this text in separate list);
    end
    $i = i + 1$;
end
for $d_j \epsilon D$ do
    1. expert discuss with annotators;
    2. based on discussion either add $d_j$ to '$BAD = []'$ with final label or
       discard it;
    $j = j + 1$;
end
```

3.2.3.2 Annotation Guidelines

To ensure the quality of annotation and better understand the dataset, it is crucial to provide detailed guidelines for annotation [98]. In few cases, dataset creators had given the liberty to the annotators to apply their perspective [48]. However, it is risky since individual interpretation and perceptions vary considerably. We ask the annotators to follow the process depicted in Figure 3.2 during annotation to avoid such issues.

To determine the initial label at first, we have to identify whether a text is aggressive or not. If it is non-aggressive, then put the label NoAG. However, if it is aggressive, we need to ascertain the reasons. In case the reasons match with multiple or none of the defined aggression dimensions *flag* the text for further discussion. Otherwise, assign an appropriate fine-grained (ReAG, PoAG, VeAG, GeAG) label. Prior annotation, we provide few samples of each category to the annotators and explain why an example should be labelled with a specific class. Each processed texts labelled by two annotators, and in case of disagreement, the expert resolved the issue through discussion.

Table 3.4: Few examples of excluded texts. Label 'flag' indicates that the expressed aggression does not match with any predefined aggression classes and remarks provided by the expert reveals the reasons for discarding the samples.

Text	**Label**	**Remarks**
আমাদের এলাকা হলে আমরা নিজেরাই ওই হিন্দুদের মার্ডার করে দিতাম। (If it was our area, we would have killed those Hindus by ourselves)	VeAG, ReAG	describe a wish to harm Hindus
এই সরকার কয়েক বছর ক্ষমতায় থাকলে বাংলাদেশে কেও আর ধর্ম পালন করতে পারবে না (If this government stays in power for a few years, no one will be able to practice religion in Bangladesh)	ReAG, PoAG	incite people in opposition to state misusing the religion
আমরা কোন সংসদে নারি মন্ত্রী দেখতে চাইনা, হোক আওয়ামী লীগ বা বিএনপি (We do not want to see any women ministers in the parliament, be it from Awami League or BNP)	PoAG, GeAG	discrimination towards women from political perspective
ধর্ষণ কারির মৃত্যুদণ্ড চাই (I want the death penalty for the rapist)	flag	aggression against a person who commit hateful crime
বাংলাদেশ থেকে টিকটক নামে বিষধর অ্যাপটিকে চিরতরে বন্ধ করা হোক (Let ban the poisonous TikTok app forever from Bangladesh)	flag	disgust against app or media
নারী মানে কলঙ্ক। সব নষ্টেরর মূলে নারী। (Women mean stigma. Women are the root of all evil)	GeAG, VeAG	verbal attack toward women

After receiving the initial label, we follow the algorithm 1 to set the final labels. For each text t_i, we check the two initial labels l_1 and l_2. A text is discarded when both of the initial labels contain *flag*. If l_1 and l_2 match, then the text and associated label added into the final lists. When disagreement is raised expert discusses with the annotators whether to keep or remove the text. The final label of such text also decided on the discussion. For **105** texts, we observe overlap among aggression dimensions and **86** texts do not fall into any defined aggression categories. Table 3.4 shows few examples with the reasoning that have been discarded due to overlap among aggression dimensions and other disagreements. Since these numbers are deficient, such instances are not included in the current corpus. We plan to address this issue in future when we attain a significant number of such instances. Finally, we get the aggressive text ('BAD') containing **14158** processed and annotated texts.

3.3 Annotation Quality

Two annotators labelled each instance of the dataset, and an expert resolved the issue through deliberations and discussions when disagreement raised between them. To check the validity and quality of the annotations, we measured the inter-rater agreement. Cohen's kappa coefficient [99] is used to calculate the agreement between annotators (equation 3.1).

$$k = \frac{O(a) - H(ca)}{1 - H(ca)} \tag{3.1}$$

Here, $O(a)$ and $H(ca)$ denoted the observed and hypothetical chance of agreement between annotators. Table 3.5 presents the kappa score on each annotation level.

Table 3.5: Kappa score on each level of annotation.

	Class	K-score	Mean
Level-A	NoAG	0.87	0.80
	AG	0.73	
Level-B	ReAG	0.54	0.65
	PoAG	0.63	
	VeAG	0.72	
	GeAG	0.69	

The highest agreement of 0.87 is achieved for NoAG class, which exhibits that this class has a more distinctive lexicon compare to other classes. Among the fine-grained classes, the maximum and minimum k-score of 0.72, 0.54 are obtained for VeAG and ReAG classes. Investigation reveals that in many cases, the aggression was expressed covertly, which is difficult to classify. This covert form of expression may be a reason behind the low agreement in fine-grained classes. The mean k-score in coarse-grained classes is 80%, while fine-grained classes obtained the mean k-score of 65%. These scores indicate substantial agreement between the annotators. Table 3.6 shows few instances for which disagreement occurred during annotation.

Table 3.6: Few examples of annotation divergence. Label-1 and label-2 denotes the first and second annotations for each text.

Text	**Label-1**	**Label-2**
মুসলিম উম্মাহ কে ধ্বংস করার জন্য একদল নারীবাদী উঠে পড়ে লেগেছে (A group of feminists has risen up to destroy the Muslim Ummah)	GeAG	ReAG
আওয়ামীলীগের লোকেরা জাহান্নামী, কারণ কুরআন হাদিস আওয়ামী লীগের ভদ্রলোকেরা মানেনা (Ihe people of Awami League are hellish because the gentlemen of Awami League do not accept Quran and Hadith)	PoAG	ReAG
এই যুগের মেয়েরা এক একটা ডাইনি, এমন মেয়েদের পুড়িয়ে মারা দরকার (The girls of this age are witches, they need to be burnt to death)	VeAG	GeAG
চাল চুরি করা এই সরকারের ঐতিহ্য। শূধু চাল নয় ভোট ও চুরি করে তারা (Stealing rice is the tradition of this government. Not just rice, they steal vote as well)	VeAG	PoAG

3.4 Dataset Statistics

Further analysis is performed to understand the properties of the dataset. This section presents the various statistical analysis of 'BAD'[2]. This work's main objective is to detect aggressive texts and categorize them into one of the fine-grained classes. The developed (BAD) uses to build the computational models. For training and evaluation, the dataset split into three sets: train (80%), validation (10%) and test (10%). Instances of the dataset are shuffled randomly before partitioning to eliminate bias and ensure randomness. Table 3.7 illustrates a summary of the dataset. Out of 14158 texts, 7351 texts are labelled as NoAG, while the

[2] **Disclaimer:** Authors would like to state that the comments/examples referred to in this section presents as they were accumulated from the original source. Authors do not use these examples to hurt individuals or a community. Moreover, authors do not promote aggressive language usage, and this research work aims to mitigate the practice of such language.

remaining 6807 texts belong to the AG class. Aggressive texts are further categorized into fine-grained classes where religious, political, verbal and gendered aggression classes have 2217, 2085, 2043 and 462 text samples.

Table 3.7: Summary of the train, validation and test set

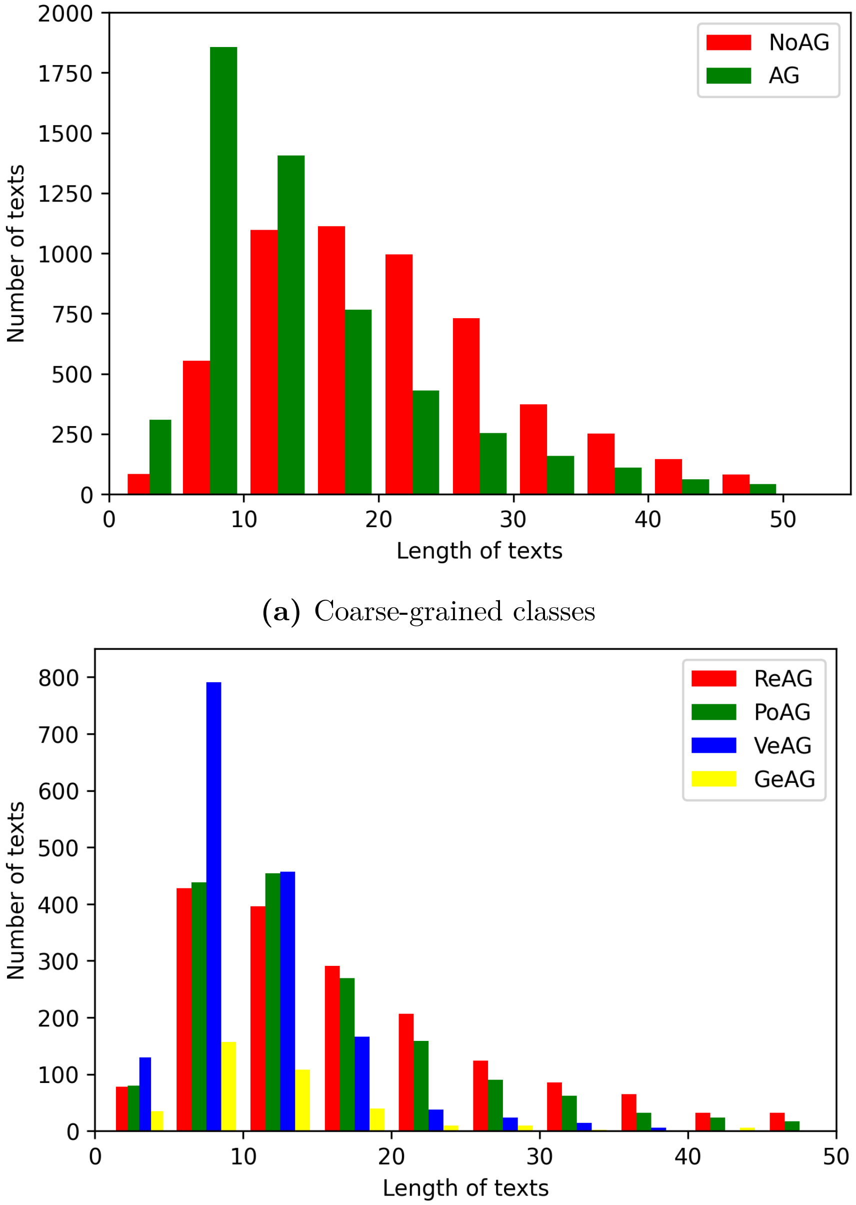
Class	Train	Valid	Test	Total
NoAG	5845	769	737	7351
AG	5481	647	679	6807
ReAG	1794	210	213	2217
PoAG	1655	229	201	2085
VeAG	1629	194	220	2043
GeAG	368	48	46	462

Table 3.8: Statistics of the training set. Here MTL, ANW, ANUW stands for maximum text length, average number of words and average number of unique words respectively.

	Level-A		Level-B			
	NoAG	**AG**	**ReAG**	**PoAG**	**VeAG**	**GeAG**
Total words	160745	78714	32282	26099	16378	3955
Unique words	26804	16155	8294	6819	5214	1738
MTL (words)	635	132	98	132	60	44
ANW (per text)	27.50	14.36	17.99	15.76	10.05	10.74
ANUW (per text)	4.59	2.95	4.62	4.12	3.20	4.72

Since the classifier models learn from the training set examples to acquire more valuable insights, we further investigated this set. Detailed statistics of the training set presented in Table 3.8. From the distribution, it notices that the training set is highly imbalanced for coarse-grained as well as fine-grained classes. There is a significant difference between aggressive and non-aggressive class in level-A in terms of the number of total words and total unique words. The NoAG class has a total of 160k words, while the AG class contained only 78k words. On average, NoAG class contained 4.6, and the AG class hold 2.9 unique words per text. In level-B, ReAG class has two and eight times as many as total words compare to VeAG and GeAG classes. ReAG consisting the maximum (18), and VeAG contained the minimum (10) number of words per text. On average, all the fine-grained classes have four unique words in each text.

In-depth investigation of the training set texts length reveals some interesting facts. Figure 3.3 depicts the number of texts vs the length of texts distribution of



(a) Coarse-grained classes

(b) Fine-grained classes

Figure 3.3: Number of text fall into various length range for different classes in training set

the training set for coarse-grained and fine-grained classes. It observed that the aggressive texts tend to be shorter than the non-aggressive ones. Approximately 4000 aggressive texts have less than 20 words among 6807 aggressive texts. On the other hand, $\approx$ 4500 non-aggressive texts have a length of higher than 20 words among 7351 non-aggressive texts. Only a fraction of texts has more than 40 words. In level-B, most of the fine-grained class texts have a length of 8 to 15 words. Several texts in PoAG and ReAG classes are approximately similar in every length range.

Table 3.9: *Jaccard* similarity between pair of coarse-grained and fine-grained classes. (c1) NoAG; (c2) AG; (f1) ReAG; (f2) PoAG; (f3) VeAG; (f4) GeAG.

Level-A			Level-B				
	c1	**c2**		**f1**	**f2**	**f3**	**f4**
c1	-	0.39	f1	-	0.40	0.24	0.35
c2	-	-	f2	-	-	0.23	0.30
			f3	-	-	-	0.33

For quantitative analysis, the *Jaccard* similarity is calculated between 200 most frequent words of each class. The similarity values between each pair exhibited in Table 3.9. ReAG-PoAG pair obtain the highest similarity score of 0.40. VeAG has maximum similarity with GeAG, while GeAG has more words in common with ReAG. Table 3.10 shows a few annotated samples of the BAD.

Table 3.10: Some examples of BAD. Level-A and level-B indicates coarse-grained and fine-grained class labels.

Text	**Level-A**	**Level-B**
ধর্ম পালন করা মানে শয়তানের উপাসনা করে। আমাদেরকে ধর্ম থেকে দূরে থাকতে হবে (Practicing religion means worshiping Satan. We have to stay away from religion)	AG	ReAG
দেশকে এই সরকারের হাত থেকে মুক্ত করতে হলে যুদ্ধ ছাড়া কোনো উপায় নেই নেই (There is no way to free the country from this government without war)	AG	PoAG
তুই দেশের বাইরে আছিস বলে এখনও বেছে আছিস।তোর সাহস থাকলে বাংলাদেশ আয় তোকে সবার সামনে হত্যা করব (You are still alive because you are out of the country. If you have the courage, come to Bangladesh. I will kill you in front of everyone)	AG	VeAG
মেয়েদের এত পড়ালেখা করে আর কি লাভ হুদাই টাকা নষ্ট (What is the benefit of educating girls so much. It is just a waste of money)	AG	GeAG
হাজারো সালাম জানাই শিক্ষকদের, যাদের অবদানে এগিয়ে যাচ্ছে বাংলাদেশ (Thousands of salutations to the teachers, who are helping Bangladesh to move forward)	NoAG	-

Chapter 4

Fine-Grained Categorization of Aggressive Texts

This work's primary concern is to identify the aggressive texts (task-A) and categorize them into four fine-grained aggression classes (task-B): ReAG, PoAG, VeAG and GeAG. To accomplish these tasks, we develop computational models using various machine learning, deep learning and transformer based methods. The dataset "BAD" is partitioned into three mutually exclusive sets: train, validation, test. Training set is used to prepare the models. Models hyperparameters are tuned based on performance on validation set. Finally models are tested on the unseen instances of the test set. This section briefly describes the methods and techniques employed to address the tasks. Figure 4.1 shows the schematic diagram of the system. Parameters and architectures of different approaches have discussed in the subsequent subsections.

4.1 Preprocessing and Feature Extraction

Raw input texts contain noises such as punctuation, digits, unwanted symbols and characters written in other languages than Bengali. All of these were removed during the preprocessing step. Various techniques such as TF-IDF and FastText word embedding are applied to extract the texts' relevant features [100].

4.1.1 TF-IDF

To train the ML-based methods, we extract the n-gram features of the texts using the term frequency-inverse document frequency technique [101] (TF-IDF). Unwanted words may get higher weights than the context-related words on the techniques like Bag of Words (BoW). The Tf-idf technique tries to mitigate this

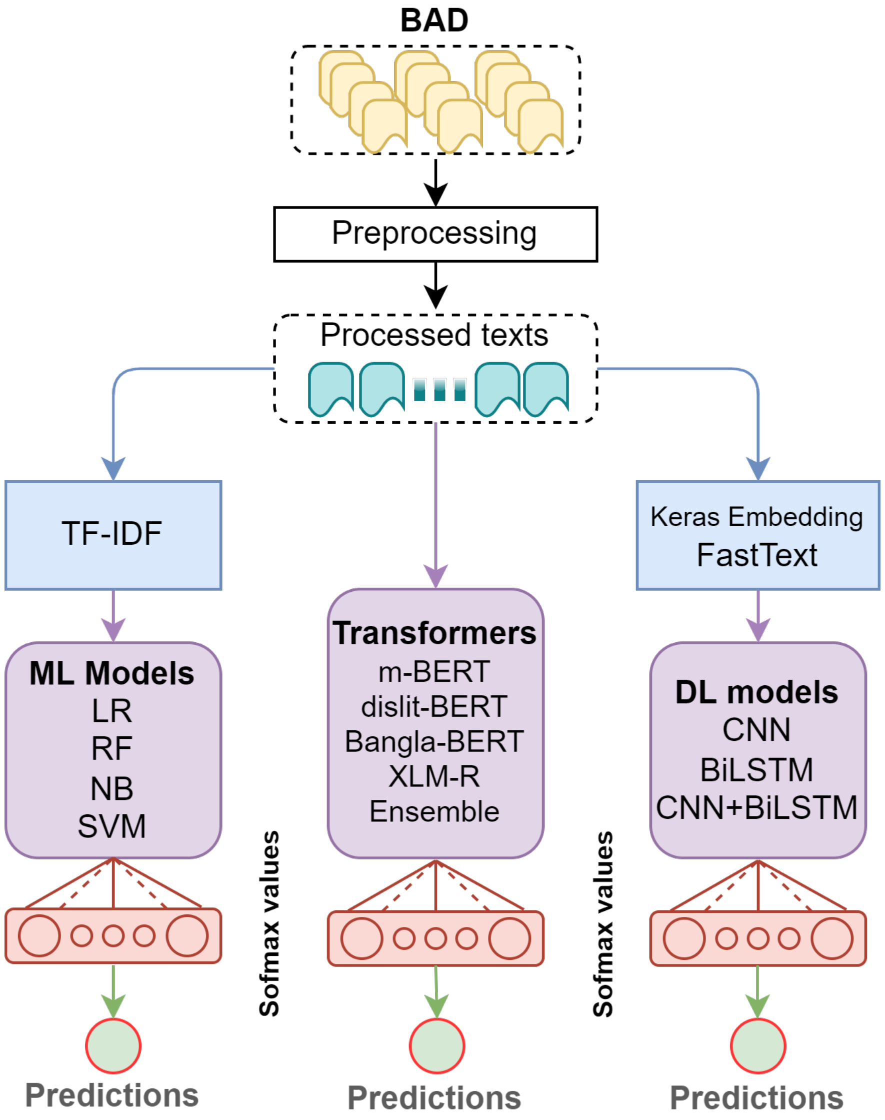


Figure 4.1: Abstract process diagram of Bengali aggressive text identification and categorization system.

weighting problem by calculating the tf-idf value according to Equation (4.1):

$$tf - idf(f_{wi}, t_i) = tf(f_{wi}, t_i) \log \frac{m}{|t\epsilon m : f_w \epsilon t|} \tag{4.1}$$

Here, $tf - idf(f_{wi}, t_i)$ indicates the tf-idf value of word f_{wi} in text document (t_i), $tf(f_{wi}, t_i)$ indicates the frequency of word f_{wi} in text document (t_i), m means total number of text documents, and $|t\epsilon m : f_w \epsilon t|$ represents the number of text document t containing word f_w.

Tf-idf value of the feature words $((f_w))$ puts more emphasis on the words related to the context than other words. To find the final weighted representation of the sentences, compute the Euclidean norm after calculating $tf - idf$ value of the feature words of a sentence. This normalization set high weight on the feature words with smaller variance. Equation (4.2) computes the norm:

$$X_{norm}(i) = X_i/\sqrt{(X_1)^2 + (X_2)^2 + ... + (X_n)^2} \tag{4.2}$$

Here, $X_{norm}(i)$ is the normalized value for the feature word f_{wi} and X_1, X_2, ..., X_n are the $tf - idf$ value of the feature word f_{w1}, f_{w2}, ..., f_{wn}, respectively. Features picked out by tf-idf technique has been applied on the classifier. Table 4.1 presents the sample feature values for first five feature words $(f_{w1}, f_{w2}, f_{w3}, f_{w4}, f_{w5})$ of the first four text samples (t_1, t_2, t_3, t_4) in our dataset.

Table 4.1: Small fragment of extracted feature values for the first four texts of the dataset.

c r	Tech-nique	f_{w1}	f_{w2}	f_{w3}	f_{w4}	f_{w5}
		Sample	**Feature**	**Values**		
t_1	tf-idf	0.35	0.03	0.42	0.59	0.23
t_2	tf-idf	0.47	0.28	0.11	0.65	0.72
t_3	tf-idf	0.04	0.11	0.22	0.75	0.44
t_4	tf-idf	0.17	0.02	0.62	0.48	0.65

Table 4.2: Representation of different N-gram features.

N-grams	“আল্লাহর নির্দেশে খুন করতে চায় আল্লাহর বান্দারা”
unigrams	‘আল্লাহর’, ‘নির্দেশে’, ‘খুন’, ‘করতে’, ‘চায়’ ‘আল্লাহর’, ‘বান্দারা'
bigrams	‘আল্লাহর নির্দেশে’, ‘নির্দেশে খুন', ‘খুন করতে’, ‘করতে চায়', 'চায় আল্লাহর’, ‘আল্লাহর বান্দারা'
trigrams	‘আল্লাহর নির্দেশে খুন’, ‘নির্দেশে খুন করতে’, ‘খুন করতে চায়’, ‘করতে চায় আল্লাহর’, ‘চায় আল্লাহর বান্দারা'

The model extracted linguistic n-gram features of the texts. The N-gram approach is used to take into account the sequence order in a sentence in order to make more sense from the sentences. Here, ‘n’ indicates the number of consecutive words that can be treated as one gram. N-gram, as well as a combination of n-gram features, have been applied in the proposed model. Table 4.2 shows the illustration of various n-gram features. A combination of unigram and bigram features are utilized for both tasks. To reduce the computation, 20k and 10k most frequent features are considered for task-1 and task-2, respectively. Inverse document reweighting technique is enabled while maximum and minimum document frequency value settled to 1.

4.1.2 Word Embedding

Although TF-IDF is an effective feature extraction (FE) technique, it could not hold the words' semantic information. Therefore, the word embedding technique is employed to capture the semantics of the words regarding the context [102]. There are many embedding techniques (e.g., GloVe, Word2Vec, FastText) and pre-trained embedding models available for English text language and their performances are outstanding for downstream tasks. Nevertheless, a limited number of embedding models that provides a set of optimized parameters are available for low resource languages including Bengali. Most of the embedding techniques have shared some standard parameters such as embedding dimension (ED), contextual window (CW), word frequency, learning rate, epoch number and context type. Among these parameters, ED, CW and minimum word frequency (min_count) are the most influential for semantic and syntactic features representation. The overall impact of three parameters (e.g., ED, CW & min_count) on classification accuracy is interrelated, and their actual combinations vary from corpus to corpus. An exponential combination of these parameters is possible and it is nearly impossible to generate these huge amount of combinations manually. Therefore, this work applied trial and error approach to the developed dataset to obtain the optimal set of hyperparameters.

Default Keras embedding layer used to obtain the embedding features. Texts are needed to be converted into fixed-length numeric sequences to acquire the features. Therefore, a vocabulary of x unique words is created where the value of x is set to 35000 and 16000 for task-1 and task-2, respectively. To achieve numeric mapping, words in a text are replaced by the word's index in the vocabulary. Since each text has a different number of words, we get a variable-length sequence which is not suitable for feature extraction. Using the Keras pad-sequences method, each sequence is converted into a fixed-length vector of size l. The value of l is set to 70 for task-1 and 50 for task-2. Extra values are removed from the long sequences, and short sequences are padded with the value 0. The Embedding layer converts a text of length l into a matrix of size $l * e$. Here, e indicates the embedding dimension that determines the word's length of the embedding vector. For both task embedding dimension value is set to 100.

The Keras embedding layer could not handle the out of vocabulary words. It set the vectors of those words to 0. The FastText embedding technique [103] is used to alleviate this problem. This technique holds the subword information since words are represented as the sum of character n-grams. This work uses the pre-trained word vectors of Bengali where the embedding dimension settled to 300 [104].

4.2 Methodology

Four ML methods (such as LR, RF, NB, SVM), three deep learning techniques (such as CNN, BiLSTM, CNN+BiLSTM) and four transformer-based models (m-BERT, distil-BERT, Bangla-BERT, XLM-R, ensemble) are implemented to investigate the Bengali aggressive text classification task performance.

4.2.1 Machine Learning Models

Features that we obtained from the previous step were used to train the machine learning models by employing different popular classification algorithms. These algorithms logistic regression (LR), random forest (RF), naïve bayes (NB) and support vector machine (SVM). We analyze these algorithms and explain their structure in our system in the following subsections.

4.2.1.1 Logistic Regression

Logistic regression is well suited for the classification problem. Equations (4.3)–(4.4) define the logistic function that determines the output of logistic regression:

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)} \tag{4.3}$$

Cost function is,

$$C(\theta) = \frac{1}{m} \sum_{i-1}^{m} c(h_\theta(x^i), y^i) \tag{4.4}$$

$$c(h_\theta(x), y) = \begin{cases} -\log(1 - h_\theta(x)) & \texttt{if} \;\; y = 0 \\ -\log(h_\theta(x)) & \texttt{if} \;\; y = 1 \end{cases}$$

Here, m indicates the number of training examples, $h_\theta(x^i)$ presents the hypothesis function of the ith training example, and y^i is the input label of ith training example.

4.2.1.2 Random Forest

Random forest is considered as the extension of decision tree. The decision tree has two types of nodes: external and internal. External nodes represent the decision class while internal nodes have the features essential for making classification. The decision tree was evaluated in the top-down approach where homogeneous data were partitioned into subsets. Its entropy determines the homogeneity of samples, which is calculated by the Equation (4.5):

$$E(S) = \sum_{l=1}^{n} p_i \log_2 p_i \tag{4.5}$$

Here, p_i is the probability of a sample in the training class, and $E(S)$ indicates entropy of the sample. The Random Forest (RF) comprises of several decision trees which operate individually. The 'Gini index' of each branch is used to find the more likely decision branch to occur. This index calculated by Equation (4.6):

$$Gini = 1 - \sum_{l=1}^{c} (p_i)^2 \tag{4.6}$$

Here, c represents the total number of class and p_i indicated the probability of the i^{th} class.

4.2.1.3 Naïve Bayes

Naïve Bayes (NB) is useful to classify discrete features such as document or text classification. NB follows multinomial distribution and uses a Bayes theorem where variables V_1, V_2, ..., V_n of class C are conditionally independent of each other given C. Equations (4.7) and (4.8) used NB for text classification in our dataset:

$$\begin{aligned} p(C|V) &= \frac{p(V|C)p(C)}{p(V)} \\ p(C|(v_1, v_2, ..., v_n) &= \frac{p(v_1|C)p(v_2|C)...p(v_n|C)p(C)}{p(v_1)p(v_2)...p(v_n)} \\ &= \frac{p(C)\prod_{i=1}^{n} p(v_i|C)}{p(v_1)p(v_2)...p(v_n)} \end{aligned} \tag{4.7}$$

Here, C is the class variable and $V = (v_1, v_2, ..., v_n)$ represents the feature vector. We assume that features are conditionally independent. The denominator remains constant for any given input; thus, it can be removed:

$$C = argmax_C \; p(C) \prod_{i=1}^{n} p(v_i|C) \tag{4.8}$$

Equation (4.8) is used to compute the probability of a given set of inputs for all possible values of class C and pick up the output with maximum probability. Laplace smoothing used and prior probabilities of a class are adjusted according to the data.

4.2.1.4 Support Vector Machine

In SVM we use, stochastic gradient descent (SGD) is technique. It is an optimization technique where a sample is selected randomly in each iteration instead of whole data samples. Equations (4.9) and (4.10) represent the weight update process for gradient descent and stochastic gradient descent at the j^{th} iteration:

$$w_j \;:=\; w_j - \alpha \, \frac{\partial J}{\partial w_j} \tag{4.9}$$

$$w_j \;:=\; w_j - \alpha \, \frac{\partial J_i}{\partial w_j} \tag{4.10}$$

Here, α indicates the learning rate, J represents the cost over all training examples, and J_i is the cost of the ith training example. It is computationally costly to calculate the sum of the gradient of the cost function of all the samples; thus, each iteration takes a lot of time to complete. To address this issue, SGD takes one sample randomly in each iteration and calculate the gradient. Although it takes more iteration to converge, it can reach the global minima with shorter training time.

The various parameters are tuned to prepare the LR, RF, NB and SVM models before performing the classification task. A summary of the parameters adopted for each model presented in Table 4.3.

Table 4.3: Parameter summary of ML models.

Model	**Parameters**
LR	optimizer=‘lbfgs’, regualizer=‘l2’, C=1.0, max_iter=400

RF	criterion=‘gini’, n_estimators=100,
	max_features=no_of_features

NB	α=1.0, class_prior=none, fit_prior=true
SVM	kernel=‘rbf’, γ=‘scale’, tol=‘0.001’, random_state=0

All four ML models utilize the combination of unigram and bigram features extracted by the TF-IDF technique. In the LR model, the 'lbfgs' optimizer is used with 'l2' regularizer. The inverse regularization strength is fixed to 1.0, and a maximum of 400 iterations are taken for solvers to converge. RF is implemented with 100 trees, and the 'gini' criterion is utilized to measure the quality of split in the tree. An internal node is partitioned if there exist at least two samples. All the system features are considered during node partitioning. The additive smoothing parameter of the NB model is set to 1. Prior class probabilities are settled based on the number of instances in the class. For SVM, the 'rbf' kernel is used with 'l2' penalizer and kernel coefficient value decided using the number of features. Tolerance of stopping criterion and random state set to 0.001 and 0, respectively.

4.2.2 Deep Learning Models

Keras and FastText embedding are used to develop deep learning models that have been applied successfully to offensive text classification [105], hostility detection [106] and aggressive texts categorization [107]. Hyperparameters and their corresponding values significantly effect DL models performance [108, 109]. Due to linguistic diversity, one model developed for a particular language can not perform similarly in another language. Thus, DL models should be prepared with their optimized hyperparameters depending on the task and language types. The preparation of DL models for Bengali aggressive text classification illustrates in the following:

4.2.2.1 CNN

Embedding features are propagated into a two-layer CNN architecture. The first and second layers contain 128 and 64 filters, respectively. Each layer consisting of kernels size (3 × 3) and features are downsampled by max-pooling technique with a (1 × 3) size window. Softmax layer take features from CNN to make the prediction. To add non-linearity *'relu'* activation function is used.

4.2.2.2 BiLSTM

Long short-term memory (LSTM) network is a commonly used variant of recurrent neural network (RNN) which used as a solution of exploding and vanishing gradient problem. Particularly, LSTM are proved [110] effective to capture the long term dependencies in a text. We applied bidirectional LSTM (BiLSTM) to keep the contextual information form previous as well as next word [111, 112]. Word embedding values of the embedding layer passed to each of the LSTM where each LSTM consists hidden units of size h. For BiLSTM, after the concatenation of each LSTM output we obtained a vector representation of length $2h$. LSTM process a input sequence of embedding vector as a pair $(e^{<i>}, y^{<i>})$. For each pair $(e^{<i>}, y^{<i>})$ and each time step t, a hidden vector $h^{<t>}$ and a remember vector $m^{<t>}$ is preserved by a LSTM. This vectors are responsible for regulating the updates and outputs of states. This helps to produce target output $y^{<i>}$ based on the past states of the $x^{<i>}$ input. The processing steps at time t executed by the Eqs. 4.11-4.16.

$$u_g = \sigma(W_u * h^{<t-1>} + I_u) \tag{4.11}$$

$$f_g = \sigma(W_f * h^{<t-1>} + I_f) \tag{4.12}$$

$$o_g = \sigma(W_o * h^{<t-1>} + I_o) \tag{4.13}$$

$$c_g = \tanh(W_c * h^{<t-1>} + I_c) \tag{4.14}$$

$$m^{<t>} = f_g \odot m^{<t-1>} + u_g \odot c_g \tag{4.15}$$

$$h^{<t>} = tanh(o_g \odot m^{<t>}) \tag{4.16}$$

Here, σ represents the sigmoid activation function, correspondingly W_u, W_f, W_o, W_c and I_u, I_f, I_o, I_c are weight matrices and projection matrices of the recurrent units. The computed gates u_g, f_g, o_g, c_g of LSTM cells play pivotal role in attaining significant attributes from the computed vector by storing in the remember vector $m^{<i>}$ as long as needed. The forget gate f_g decides the amount of information to be dumped from the previous remember vector $m^{<i-1>}$, on the contrary the update gate c_g use input gate u_g and previous remember vector $m^{<i-1>}$ to write updated information in the new remember vector $m^{<i>}$. Finally, output gate o_g monitors which information goes from new memory vector $m^{<i>}$ to the hidden vector $h^{<i>}$.

In this work we used two layers where the first layer has 128 and the second layer has 64 bidirectional LSTM cells. The dropout value settled to 0.2, and the features passed to the softmax layer for prediction.

4.2.2.3 CNN + BiLSTM

In the combined method, CNN and BiLSTM added sequentially with slight modifications in their previous architecture. One layer of CNN with 128 filters and a kernel size of (3×3) is used. Features from CNN are downsampled using a pooling layer and propagated through two layers of BiLSTM. The first layer has 64, and the second layer has 32 LSTM units. The dropout rate is unaltered, and hidden representation is passed to the softmax layer.

Table 4.4: Hyperparameter summary of DL models. C+B denotes combined CNN, BiLSTM method.

Hyperparameter	**Hyperparameter space**	**CNN**	**BiLSTM**	**C+B**
Input length	-	70(task-A), 50 (task-B)		
Embedding dimension	[32, 64, 100, 128, 200, 256, 300, 400]	100(task-A), 300 (task-B)		
Filters (layer-1)	[8, 16, 32, 64 128]	128	-	128
Kernel size	[3, 5, 7]	3	-	3
Filters (layer-2)	[16, 32, 64 128, 256]	64	-	-
Pooling type	‘max’, ‘average’	‘max’	-	‘max’
LSTM cell (layer-1)	[8, 16, 32, 64 128]	-	128	64
Dropout rate	[0.1, 0.15, 0.20, 0.25, 0.30, 0.35]	-	0.2	0.2
LSTM cell (layer-2)	[16, 32, 64 128, 256]	-	64	32
Learning rate	[0.3, 0.2, 0.1, 0.001, 0.0001, 0.00001]	0.001		
Optimizer	‘adam’, ‘Nadam’, ‘RMSprop’	‘adam’		
Batch size	[8, 16, 32, 64, 128]	16		
Epochs	-	30		

The input sequence length is set to 50 and 70 for task-A and task-B. These values are fixed based on the insights from length analysis shown in Figure 3.3. The dimension for Keras and FastText embedding settled to 100 (task-A) and 300 (task-B), respectively. The rest of the architecture is similar for both types of tasks. All the models use the ‘adam’ optimizer with a learning rate of 0.001. Models are trained with 16 samples per batch for 30 epochs. The model with the highest validation accuracy is stored using callbacks. Table 4.4 summarizes hyperparameter values used by the DL models. Experimentation was performed using the values from the hyperparameter space. Optimum hyperparameter values have been settled in a trial and error fashion depending on the validation set outcomes.

4.2.3 Transformer Models

Past studies reveal that the transformer models trained in monolingual, multilingual or cross-lingual settings are achieving the state of the art performance in categorising unwanted texts [51, 32, 15]. Thus, this work employed four pre-trained transformer models: Multilingual Bidirectional Encoder Representations from Transformers (m-BERT) [113], distilled version of BERT (distil-BERT) [114], Bangla-BERT [115] and cross-lingual version of Robustly Optimized BERT (XLM-R) [116]. By varying hyperparameters, these models are fine-tuned over the (BAD). Models are fetched from the HuggingFace[1] library and built with ktrain packages [117].

Multilingual-BERT is a large model which has been trained with 104 monolingual datasets. We use the 'bert-base-multilingual-uncased' model with 12 layers, 12 heads, and 110M parameters. The model is fine-tuned by altering the learning rate, batch size and epochs. A distilled version of m-BERT is utilised, having six layers, 768 dimensions and 12 heads. This model reduced the computational time and preserved the overall system performance up to 95%. The system also trained with the base version ('distilbert-base-multilingual-cased') fetched from the HuggingFace library. Another pre-trained model, 'bangla-bert-base', is also implemented. This model is trained with monolingual Bengali CommonCrawl corpus and utilises the BERT base model's architecture. XLM-R is a cross-lingual model which outdoes m-BERT in various benchmarks. This model is built over the of 100 languages and has 12 layers, eight heads and approximately 125M parameters. We implement the 'xlm-roberta-base' model for our purpose. Table 4.5 shows a list of parameters for BERT variants.

Table 4.5: Fine-tuned parameter values of transformers.

Hyperparameter	**Value**
Fit method	'auto_fit'
Learning rate	$2e^{-5}$
Epochs	20
Batch size	12
Max sequence length	50, 70

All the models are fine-tuned on BAD using the ktrain 'auto_fit' method. Models are trained for 20 epochs with a batch size of 12. A triangular learning rate policy is adopted with a maximum learning rate of $2e^{-5}$. Max sequence length for the texts is settled to 50 for task-1 and 70 for task-2. Model weights are stored using checkpoint, and the best model is chosen based on its efficiency in the validation set.

4.3 Proposed Ensemble Model

Recent works exhibited that the ensemble of transformers can significantly improve the efficiency of a classification task [118, 119]. Ensemble methods exploit the strength of the individual models and increase the system's predictive accuracy. Four transformer models are used (m-BERT, distil-BERT, Bangla-BERT,

[1] https://huggingface.co/models

XLM-R) that is fine-tuned on the developed dataset. Figure 4.2 shows the architecture of the proposed weighted ensemble technique. This work employs two types of ensemble techniques: average (A-ensemble) and weighted (W-ensemble). The average (A) ensemble computes the average of the softmax probabilities of the participating models. This averaging technique considers a class with the maximum probability as the output class. In this method, prior results of the base classifiers is not considered [120, 121]. On the other hand, this work proposes a weighted ensemble technique which strengthen the classifiers' performance to identify and categorize Bengali aggressive texts.

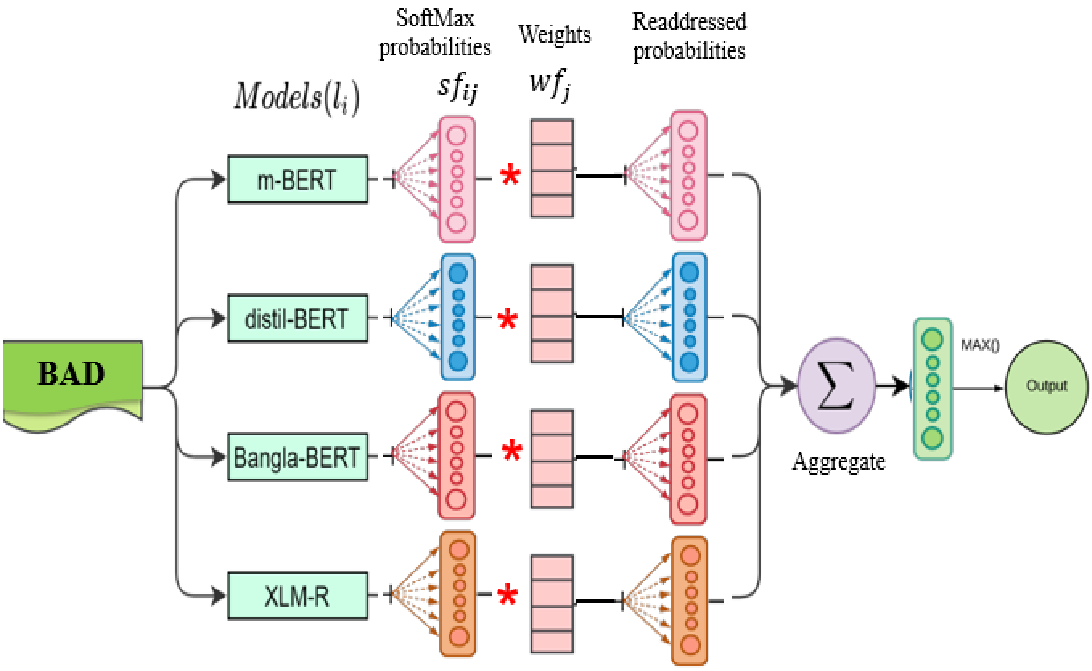


Figure 4.2: Architecture of the proposed model.

Rather than simple or traditional averaging, the proposed method offers an additional weight to the softmax probabilities of a model based on its prior results. Lets consider, we have 'l' existing models and 'm' validation/test set instances. A model classifies each instances m_i into one of n predefined classes. For each m_i, a model l_j provides a softmax probability vector of size 'n', $sp_{ij}[n]$. Thus, models output becomes: $\langle sp_{11}[], ..., sp_{m1}[] \rangle$, $\langle sp_{12}[], sp_{22}[], ..., sp_{m2}[] \rangle$,.., $\langle sp_{1l}[], sp_{2l}[], ..., sp_{ml}[] \rangle$. Prior weighted f_1-scores of 'l' models measured on the validation set are $w_{f1}, w_{f2}, ..., w_{fl}$. Utilizing these values, the proposed technique computes the output as described in Eq. (4.17).

$$O = \max\left(\frac{\forall_{i\epsilon(1,m)} \sum_{j=1}^{l} sp_{ij}[n] * w_{fj}}{\sum_{j=1}^{l} w_{fj}}\right) \tag{4.17}$$

Here, O denotes the vector of m, which contains the ensemble method's predictions.

Algorithm 2: Process of W-ensemble

```
Input: Softmax probabilities and WF score
Output: Predictions of the W-ensemble
$sp \leftarrow []$ (softmax probabilities);
$w_f \leftarrow []$ (weighted $f_1$ scores);
$sum = []$ (weighted sum);
for $i \epsilon (1, m)$ do
    for $j \epsilon (1, l)$ do
        $sum[i] = sum[i] + (sp_{ij}[] * w_{fj})$;
        $j = j + 1$;
    end
    $i = i + 1$;
end
$n\_sum = 0$;
for $j \epsilon (1, l)$ do
    $n\_sum = n\_sum + w_{fj}$;
    $j = j + 1$;
end
$P = (sum/n\_sum)$ //normalized probabilities;
$O = \max(P)$ // set of predictions;
```

Algorithm 2 describes the process of calculating ensemble weights. Softmax probabilities of the models are aggregated after multiplying with the WF scores. Probabilities are normalized by dividing with the sum of WF scores. Finally, output predictions are computed by taking the maximum from the probabilities.

4.3.1 Insights of Weight Calculation

Figure 4.3 exhibits the process of average ensemble technique. Figure 4.4 shows how weights are calculated and process of weighted ensemble technique. Let's consider, we have two classifier with 90% and 86% validation accuracy respectively. These classifiers are trying to classify a sample text into two classes $\{c_1, c_2\}$. The average ensemble technique simply take the probability score of each classes for the classifiers and aggregate them. Here, the aggregation score for c_1 and c_2 is 0.37 and 0.36 respectively. Then the class with maximum probability is considered as the output class. Hence the output class is c_1. The prior validation accuracy of the classifiers have no impact on the ensemble. Same priority is given to each of the classifiers softmax predictions.

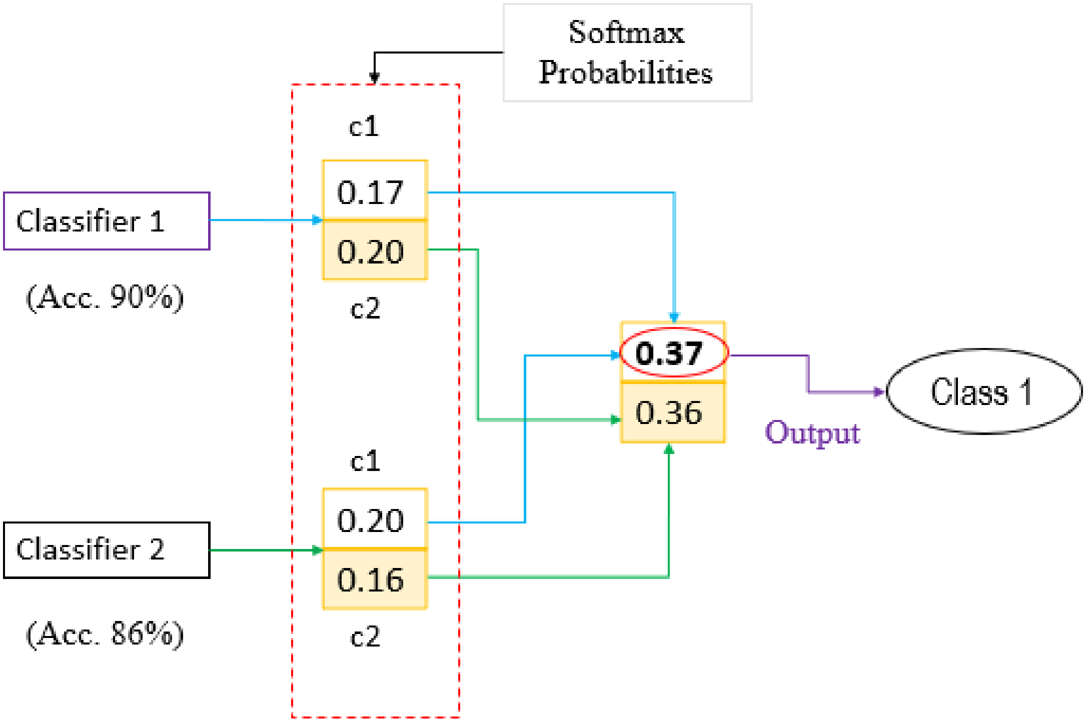


Figure 4.3: Process of average ensemble method

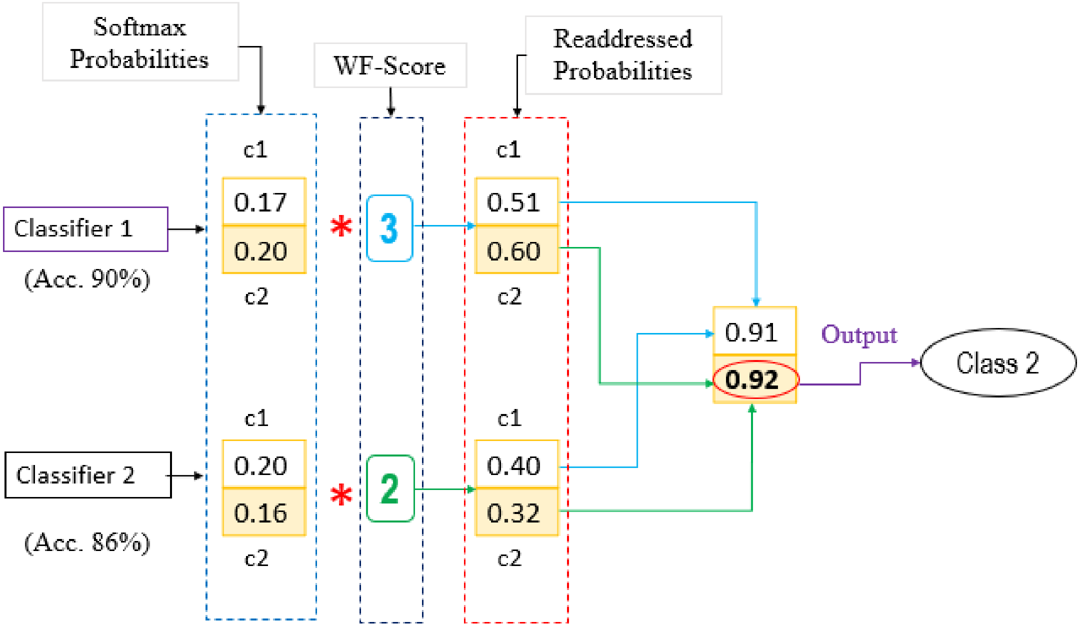


Figure 4.4: Process of proposed weighted ensemble technique

On the other hand, our proposed weighted ensemble technique does not simply take the average of the probabilities. It aggregate the probabilities of the classifiers after multiplying with weights. These weights help to put emphasis on the classifiers. The classifier with higher validation accuracy on this dataset will get higher weights. This work considers the weighted f_1-scores of the models on the validation set as the weighting factor. For calculation simplicity, the weights are set to 3 and 2 respectively for classifiers respectively. Higher weight is given to classifier 1 since it has higher validation accuracy. After multiplying the initial softmax probabilities with the assigned weights a set of readdressed probabilities are obtained. These readdressed probabilities are aggregated and the output the is the one with maximum probability score. For this example final prediction is flipped. With average ensemble technique class 1 was the output but after in weighted ensemble technique class 2 is the predicted class.

4.3.2 Complexity Analysis

The strength of the proposed model is that it offers improved accuracy without increasing complexity of the models. It utilizes the f_1-scores as weights that have already been calculated on the validation sets. Therefore no extra calculation is required to have the weights. The f_1-scores of each of the base classifiers of ensemble are simply stored during validation and used during testing. Our models used recurrent, convolutional and self-attention layers during training. The complexity [113, 122] of each of the layers are showed in table 4.6.

Table 4.6: Maximum per-layer complexity for different types of layers.

Layer Type	Complexity per layer	Weighting complexity	Final Complexity
Convolutional	$O(k \times n \times d^2)$	$O(1)$	$O(k \times n \times d^2)$
Recurrent	$O(n \times d^2)$	$O(1)$	$O(n \times d^2)$
Self-attention	$O(n^2 \times d)$	$O(1)$	$O(n^2 \times d)$

Here, n is the sequence length, d is the representation dimension and k is the kernel size of convolutions. The weighting technique simply scales up the probability of each layer. Hence, the probability of this technique is $O(1)$. Therefore the final complexity of each layer remain unchanged and our proposed model do not add any significant complexity to the models.

Chapter 5

Results and Discussions

This section presents a comprehensive performance analysis of the approaches that we employed for Bengali aggressive text classification. Various evaluation measures and the outcomes of the different models will be described here subsequently. Moreover, this section explains the proposed model's error analysis and compares its performance with other existing techniques.

5.1 Experimental Requirements

Experiments carried out on Google colaboratory platform with python 3 Google cloud engine backend (GPU). A 12.5GB RAM and 64GB disk space have been utilized to implement the models. To process and prepare the data, we used pandas (1.1.4) and numpy (1.18.5). The machine learning models are built with scikit-learn (0.22.2) packages, while the training of DL models is performed using Keras (2.4.0) and TensorFlow (2.3.0). Transformer models are developed with ktrain (0.25) packages [117].

5.2 Evaluation Measures

The train, validation and test instances are utilized to develop the models. It ensured that all the instances of these sets are mutually exclusive. Models learn from the training set instances while the hyperparameter values are settled based on the validation set. Finally, the trained models are evaluated using the unseen instances of the test set. Various statistical measures are used to calculate and compare the performance of the systems. Few measures utilized for evaluation illustrated in Eqs. (5.1)-(5.4).

- Precision: calculate the number of samples (s_i) actually belong to class (c)

among the samples (s_i) labeled as class (c).

$$P = \frac{True\,positive}{True\,positive + False\,positive} \tag{5.1}$$

- Recall: calculate how many samples (s_i) are correctly labeled as class (c) among the total number of samples (s_i) of class (c).

$$R = \frac{True\,positive}{True\,positive + False\,negative} \tag{5.2}$$

- Error: gives the value that how many samples are wrongly classified.

$$E = \frac{False\,positive + False\,negative}{Number\ of\ samples} \tag{5.3}$$

- F_1-score: calculated by simply averaging precision and recall ($F = \frac{2PR}{P+R}$). Since the dataset is imbalance we calculate the weighted f_1-score which is defined as,

$$WF = \frac{1}{N}\sum_{i=1}^{c} F_i n_i, \quad N = \sum_{i=1}^{c} n_i \tag{5.4}$$

  Here N, F_i and n_i denotes total samples in test set, f_1-score and number of samples in class (i).

The weighted f_1-score (WF) is considered to determine the superiority of the models. Other scores such as precision, recall, error rate are also reported to get an understanding of the model's performance on different classes.

5.3 Results

The current work investigated all possible combinations of the base classifiers (i.e., transformers) for both tasks (i.e., fine-grained and coarse-grained). In the subsequent subsection we analyze the obtained results briefly.

5.3.1 Coarse-grained Classification

Table 5.1 exhibits the outcomes of the developed models for coarse-grained classification. Among the ML models, LR acquired the highest weighted f_1-score (WF) of 0.8968. NB and SVM also achieved a higher than 89% WF score. In the case of DL, results revealed that models with Keras embedding achieved better scores in the classification task. Surprisingly, all the DL models' efficiency

Table 5.1: Evaluation results of different models on the test set for coarse-grained identification of aggressive texts.

		NoAG			**AG**			
	Method	**P**	**R**	**F**	**P**	**R**	**F**	**WF**
	LR	0.89	0.91	0.90	0.90	0.88	0.89	0.8968
	RF	0.80	0.91	0.85	0.89	0.76	0.82	0.8370
	NB	0.90	0.89	0.90	0.88	0.89	0.89	0.8913
	SVM	0.88	0.93	0.90	0.92	0.86	0.89	0.8953
	CNN (C)	0.92	0.90	0.91	0.90	0.92	0.91	0.9110
	BiLSTM (B)	0.92	0.90	0.91	0.89	0.92	0.90	0.9061
	C+B	0.90	0.92	0.91	0.91	0.89	0.90	0.9067
	CNN (FT)	0.91	0.91	0.91	0.90	0.90	0.90	0.9053
	BiLSTM (FT)	0.89	0.93	0.91	0.91	0.87	0.89	0.8995
	C+B (FT)	0.93	0.87	0.90	0.87	0.93	0.90	0.8997
	m-BERT (MB)	0.95	0.90	0.92	0.90	0.95	0.92	0.9223
	distil-BERT (DB)	0.91	0.93	0.92	0.92	0.90	0.90	0.9145
	Bangla-BERT (BB)	0.92	0.91	0.92	0.91	0.91	0.91	0.9124
	XLM-R (XR)	0.93	0.93	0.93	0.92	0.93	0.92	0.9272
A-ensemble	MB+DB	0.92	0.92	0.92	0.91	0.91	0.91	0.9166
	MB+BB	0.94	0.92	0.93	0.91	0.94	0.92	0.9251
	MB+XR	0.94	0.93	0.94	0.92	0.94	0.93	0.9329
	DB+BB	0.92	0.92	0.92	0.91	0.92	0.92	0.9187
	DB+XR	0.93	0.93	0.93	0.92	0.92	0.92	0.9265
	BB+XR	0.94	0.93	0.94	0.93	0.93	0.93	0.9321
	MB+DB+BB	0.94	0.92	0.93	0.91	0.94	0.93	0.9286
	MB+DB+XR	0.94	0.93	0.94	0.92	0.94	0.93	0.9323
	DB+BB+XR	0.94	0.92	0.93	0.92	0.94	0.93	0.9308
	MB+DB+BB+XR	0.94	0.93	0.94	0.92	0.94	0.93	0.9336
W-ensemble	MB+DB	0.92	0.91	0.92	0.92	0.91	0.91	0.9173
	MB+BB	0.94	0.93	0.93	0.92	0.94	0.91	0.9258
	MB+XR	0.94	0.93	0.94	0.92	0.94	0.93	0.9329
	DB+BB	0.93	0.92	0.92	0.92	0.92	0.92	0.9209
	DB+XR	0.93	0.93	0.93	0.93	0.92	0.92	0.9279
	BB+XR	0.93	0.94	0.94	0.93	0.94	0.93	0.9336
	MB+DB+BB	0.94	0.92	0.93	0.92	0.94	0.93	0.9287
	MB+DB+XR	0.94	0.93	0.94	0.93	0.94	0.93	0.9332
	DB+BB+XR	0.94	0.92	0.93	0.94	0.92	0.93	0.9315
	MB+DB+BB+XR	0.95	0.93	0.94	0.92	0.94	0.93	**0.9343**

reduced by ≈ 1% after using FastText embedding. CNN with Keras embedding gained the maximal WF score of 0.9110 amid the DL models. A significant rise is observed in the system performance with transformer models. Multilingual BERT and XLM-R models attain a higher than 92% WF score. Initially, we have evaluated a total of 14 models using several statistical measures (such as

Table 5.2: Evaluation results of various models on the test set for fine-grained classification.

	ReAG			**PoAG**			**VeAG**			**GeAG**			
Method	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**WF**
LR	0.87	0.90	0.89	0.91	0.93	0.92	0.86	0.90	0.88	0.75	0.39	0.51	0.8689
RF	0.89	0.74	0.81	0.82	0.88	0.85	0.76	0.94	0.84	0.83	0.33	0.47	0.8088
NB	0.79	0.90	0.84	0.88	0.91	0.89	0.84	0.87	0.86	0.00	0.00	0.00	0.8049
SVM	0.84	0.89	0.87	0.90	0.92	0.91	0.82	0.91	0.86	1.00	0.13	0.23	0.8342
CNN (C)	0.89	0.87	0.88	0.93	0.89	0.91	0.81	0.89	0.85	0.54	0.41	0.47	0.8504
BiLSTM (B)	0.88	0.88	0.88	0.90	0.91	0.90	0.84	0.87	0.85	0.65	0.52	0.58	0.8569
C+B	0.84	0.89	0.86	0.91	0.93	0.92	0.86	0.87	0.86	0.38	0.24	0.29	0.8412
CNN (FT)	0.89	0.85	0.87	0.94	0.87	0.90	0.83	0.89	0.86	0.50	0.59	0.54	0.8524
BiLSTM (FT)	0.86	0.89	0.87	0.89	0.92	0.91	0.90	0.85	0.87	0.61	0.59	0.60	0.8641
C+B (FT)	0.90	0.88	0.89	0.90	0.94	0.92	0.85	0.90	0.87	0.67	0.43	0.53	0.8691
m-BERT (MB)	0.92	0.92	0.92	0.97	0.96	0.96	0.91	0.88	0.90	0.60	0.74	0.66	0.9073
distil-BERT(DB)	0.90	0.92	0.91	0.93	0.92	0.93	0.85	0.92	0.88	0.72	0.39	0.51	0.8794
Bangla-BERT(BB)	0.94	0.96	0.95	0.96	0.95	0.95	0.90	0.92	0.91	0.74	0.61	0.67	0.9176
XLM-R (XR)	0.93	0.93	0.93	0.97	0.97	0.97	0.89	0.92	0.90	0.71	0.59	0.64	0.9146
A-ensemble models													
MB+DB	0.91	0.92	0.91	0.97	0.96	0.96	0.90	0.90	0.90	0.65	0.65	0.65	0.9059
MB+BB	0.93	0.96	0.95	0.98	0.97	0.97	0.91	0.92	0.91	0.73	0.65	0.69	0.9271
MB+XR	0.93	0.93	0.93	0.98	0.97	0.97	0.91	0.91	0.91	0.67	0.70	0.68	0.9210
DB+BB	0.93	0.96	0.94	0.96	0.97	0.97	0.91	0.94	0.92	0.75	0.52	0.62	0.9216
DB+XR	0.93	0.95	0.94	0.96	0.97	0.96	0.89	0.93	0.91	0.75	0.52	0.62	0.9144
BB+XR	0.94	0.96	0.95	0.97	0.97	0.97	0.90	0.92	0.91	0.71	0.63	0.67	0.9241
MB+DB+BB	0.91	0.94	0.93	0.97	0.97	0.97	0.90	0.93	0.91	0.76	0.57	0.65	0.9167
MB+DB+XR	0.93	0.92	0.93	0.98	0.97	0.97	0.90	0.93	0.91	0.70	0.67	0.69	0.9203
DB+BB+XR	0.94	0.95	0.95	0.97	0.97	0.97	0.90	0.95	0.92	0.73	0.59	0.65	0.9266
MB+DB+BB +XR	0.94	0.96	0.95	0.98	0.97	0.97	0.91	0.93	0.92	0.75	0.63	0.69	0.9275
W-ensemble models													
MB+DB	0.92	0.92	0.92	0.97	0.96	0.96	0.90	0.90	0.90	0.67	0.72	0.69	0.9123
MB+BB	0.94	0.96	0.95	0.98	0.97	0.98	0.91	0.92	0.92	0.69	0.67	0.70	0.9301
MB+XR	0.93	0.94	0.93	0.98	0.97	0.97	0.91	0.91	0.91	0.68	0.70	0.69	0.9223
DB+BB	0.93	0.95	0.94	0.96	0.97	0.97	0.91	0.94	0.92	0.75	0.52	0.62	0.9202
DB+XR	0.93	0.94	0.93	0.97	0.97	0.97	0.89	0.93	0.91	0.75	0.59	0.66	0.9172
BB+XR	0.94	0.96	0.95	0.97	0.97	0.97	0.90	0.92	0.91	0.69	0.59	0.64	0.9206
MB+DB+BB	0.91	0.94	0.93	0.97	0.97	0.97	0.90	0.92	0.91	0.74	0.57	0.64	0.9154
MB+DB+XR	0.93	0.92	0.93	0.98	0.97	0.97	0.90	0.93	0.91	0.70	0.67	0.69	0.9203
DB+BB+XR	0.94	0.95	0.95	0.97	0.97	0.97	0.90	0.94	0.92	0.78	0.63	0.69	0.9308
MB+DB+BB +XR	0.94	0.96	0.95	0.98	0.97	0.97	0.92	0.93	0.92	0.74	0.63	0.68	**0.9311**

precision, recall, f1-scores) on the dataset (BAD) and empirically observed each of them. In particular, based on the highest weighted f1-score, we selected four base models (m-BERT, distil-BERT, Bangla-BERT, XLM-R) for the ensemble. All possible combination of these base models are investigated for both average and weighted ensemble technique. As per expectation, we noticed an increase in the performance where the average ensemble method attained maximum WF score of 0.9336. Finally, by employing the proposed weighted ensemble method, the system obtains the highest WF score of **0.9343**, which outperformed all other models.

5.3.2 Fine-grained Classification

Evaluation results for fine-grained classification presented in Table 5.2. Like coarse-grained classification, LR also achieved the maximum WF score amid the ML models in fine-grained classification. Interestingly LR outdoes the CNN and BiLSTM models implemented with Keras embedding in fine-grained classification by attaining a 0.8689 WF score. Although in coarse-grained classification, DL models performed poorly with FastText embedding. However, in fine-grained classification, the DL models obtained a higher WF score with FastText. Among the ML and DL models, combined CNN+BiLSTM acquired the maximum of 0.8691 WF score. Transformer based methods also showed noteworthy performance. Bangla-BERT achieved the maximum WF score (0.9176) amid the BERT variants. However, the proposed weighted ensemble method surpasses all other models and achieves the highest WF score of **0.9311** for fine-grained classification. Thus, it is confirmed that the performance of the proposed system has significantly improved on both tasks after employing the weighted ensemble technique. This higher performance might happen because the weighting technique can adjust the softmax probabilities of the base classifiers of the ensemble depending on their prior results.

5.3.3 Effects of Cross-validation

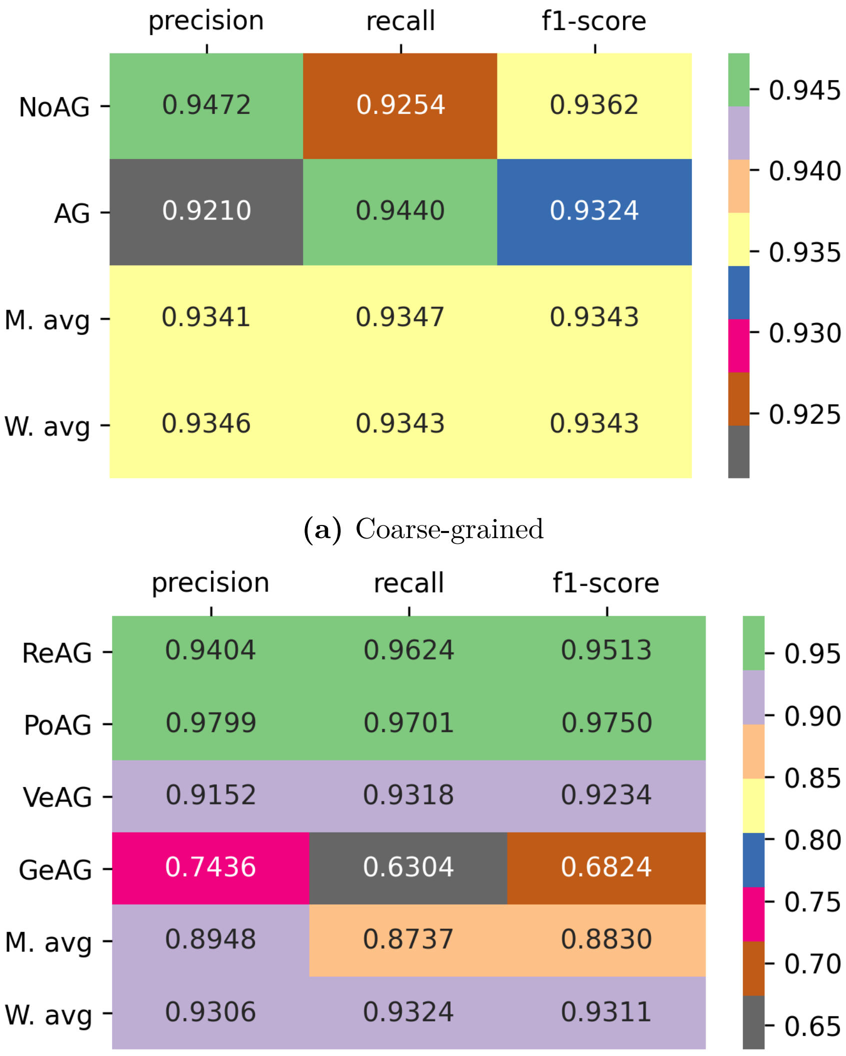
The final model is cross-validated to acquire better insight regarding the proposed model's performance. For ease of analysis, we only cross-validated the four base transformers and the proposed combination for both the A-ensemble and W-ensemble techniques. A 10-fold cross-validation technique [123] has been carried out on a combined (training + validation) set using *scikit-learn*. Table 5.3 represents the cross-validation results of the models. The W-ensemble technique has achieved the highest mean weighted f_1-scores of 92.85% (task-A) and 92.21% (task-B). The average standard deviation is approximately 3% for both tasks. The analysis of cross-validation results revealed that the model's performance had not significantly affected by the dataset split.

5.3.4 Analysis of Classification Report

To get more insights, we take a closer look at the proposed model's classification reports shown in Figure 5.1. In coarse-grained classification, NoAG class has the higher (0.9472) precision while AG has the higher (0.9440) recall value. Since both classes have approximately similar instances, no meaningful difference

Table 5.3: 10-fold cross-validation results of the transformer-based models, including the proposed technique on the combined (training + validation) set. The values in the cell represent the weighted f_1-scores, and *Std* denotes standard deviation.

	Task-A		**Task-B**	
Method	**Mean**	**Std**	**Mean**	**Std**
m-BERT(M)	0.9223	0.0291	0.9102	0.0235
dislit-BERT(D)	0.9145	0.0294	0.8777	0.0381
Bangla-BERT(B)	0.9124	0.0248	0.9167	0.0304
XLM-R(X)	0.9262	0.0279	0.9139	0.0355
A(M+D+B+X)	0.9268	0.0267	0.9205	0.0263
W(M+D+B+X)	0.9285	0.0254	0.9221	0.0266



(a) Coarse-grained

(b) Fine-grained

Figure 5.1: Classification report of the proposed (w-ensemble) model on the test set.

is observed between the macro and weighted f_1-score. Among the fined-grained classes, GeAG and PoAG obtained the minimum (0.6824) and maximum (0.9750) f_1-scores. The performance of the proposed model (W-ensemble) is lower in the GeAG than in other classes. The limited number of instances in GeAG class has resulted in a reduced performance than others. Moreover, the confusion matrix analysis and the Jaccard similarity index revealed that GeAG class mostly overlaps with the VeAG class. Therefore, the overall misclassification rate increased and hence the performance of the W-ensemble model is decreased in GeAG class. Results noticed a $\approx 5\%$ difference in macro and weighted f_1 score values as the classes are highly imbalanced.

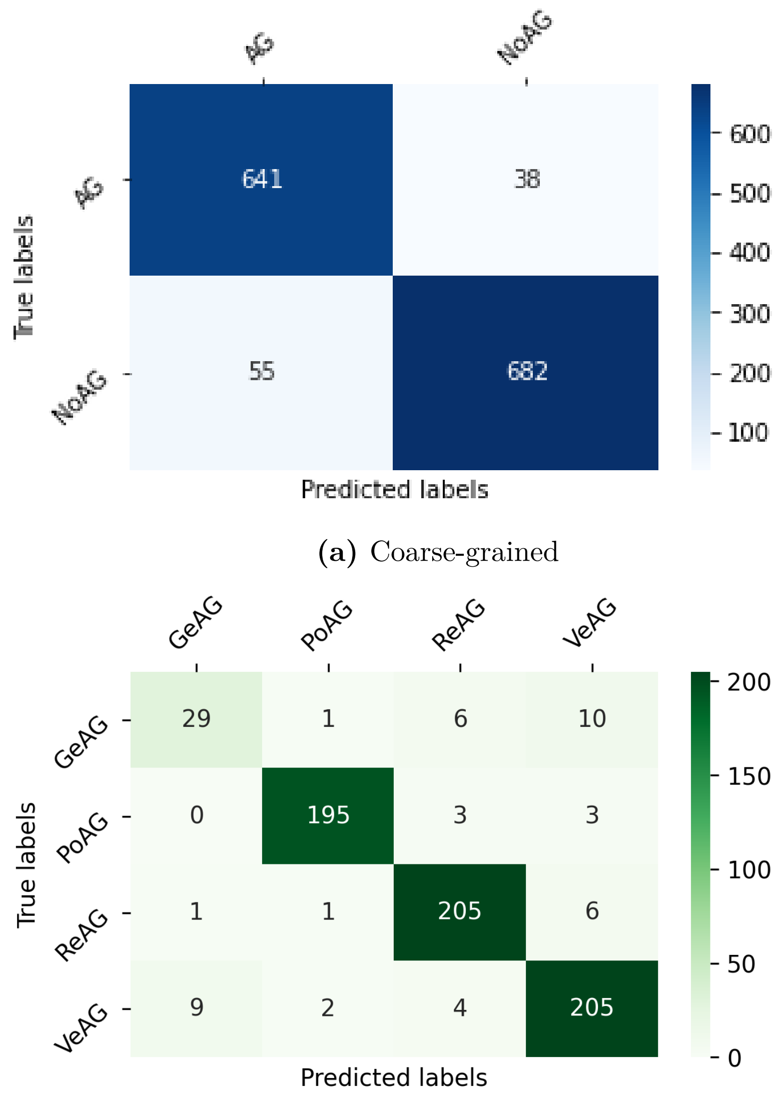
In summary, LR achieves the highest WF score among the ML models in both tasks. CNN and CNN + BiLSTM attain the maximum WF score with Keras and FastText embedding, respectively. It noticed that when the number of classes increases, the efficiency of ML and DL models decreases. However, the performance of the transformer models remains consistent. This consistency occurred due to the massive number of examples usage by pre-trained models, so their generalization capability is much higher. The results showed that the proposed weighted ensemble method outperformed all ML, DL and transformer models in coarse and fine-grained classification. The proposed architecture's ability to emphasize the models' softmax predictions based on their prior results might be the reason behind this superior performance.

5.4 Error Analysis

It is evident from Tables 5.1 and 5.2 that the weighted ensemble is the best performing model to classify aggressive texts in Bengali. Here, a detailed error analysis is carried out quantitatively and qualitatively to acquire in-depth insights into individual model's performance.

5.4.1 Quantitative analysis

Figure 5.2a shows the confusion matrix for coarse-grained classification. It indicates that 38 instances of AG class wrongly classified as NoAG whereas 55 comments of NoAG class labelled as AG by the W-ensemble model. Some aggressive texts express aggression implicitly, which is very difficult to identify. Figure 5.2b indicated that the false-negative rate in GeAG and PoAG classes are higher than the false positive rate. In contrast, false-positive values in ReAG and VeAG classes are higher. The model classifies 195, 205 and 205 instances cor-



(a) Coarse-grained

(b) Fine-grained

Figure 5.2: Confusion matrix of the proposed (w-ensemble) model on the test set

rectly among 201, 213 and 220 instances of PoAG, ReAG and VeAG classes. In the case of the GeAG class, the proposed model performed poorly. It incorrectly classifies 17 texts among 46 test texts. All the classes mostly make confusing with VeAG classes. The presence of outrageous words in all the aggressive classes may cause this confusion. The error rate for all models in coarse and fine-grained classification is presented in Table 5.4. The proposed weighted ensemble technique is achieved the lowest error rate of 6.56% (task-A) and 6.76% (task-B).

Table 5.4: Error rate of various models in Task-A (coarse-grained) and Task-B (fined-grained).

Method	**Task-A (%)**	**Task-B (%)**
LR	10.31	12.50
RF	16.17	18.24
NB	10.88	16.62
SVM	10.45	14.71
CNN (C)	8.9	14.71
BiLSTM (B)	9.39	14.12
C+B	9.32	15.15
CNN (FT)	9.46	15.00
BiLSTM (FT)	10.03	13.53
C+B (FT)	10.03	12.56
m-BERT (MB)	7.77	9.26
distil-BERT (DB)	8.55	11.46
Bangla-BERT (BB)	8.76	8.09
XLM-R (XR)	7.27	8.38
A-ensemble models		
MB+DB	8.33	9.41
MB+BB	7.48	7.21
MB+XR	6.72	7.93
DB+BB	8.12	7.5
DB+XR	7.34	8.24
BB+XR	6.71	7.5
MB+DB+BB	7.13	8.09
MB+DB+XR	6.76	7.94
DB+BB+XR	6.92	7.43
MB+DB+BB +XR	6.64	7.06
W-ensemble models		
MB+DB	8.26	8.82
MB+BB	7.42	6.98
MB+XR	6.72	7.79
DB+BB	7.91	7.64
DB+XR	7.19	8.09
BB+XR	6.64	7.78
MB+DB+BB	7.13	8.24
MB+DB+XR	6.76	7.93
DB+BB+XR	6.84	6.90
MB+DB+BB +XR	6.56	6.76

5.4.2 Qualitative analysis

Table 5.5 shows a few examples which exhibit the contrasting nature of the transformer models. Although the models quantitatively achieve similar scores, their class predictions are qualitatively different. One model can classify a test sample correctly while another can not. The proposed transformer-based weighted ensemble method can be helpful to deal with this contrasting nature. For better understanding, outputs of the ensemble models are further investigated. Table 5.6 illustrates some examples of incorrect classification on test data.

Table 5.5: Instances exhibiting the contrasting nature of the transformer models. MB, DB, BB and XR denotes predicted labels for multilingual-BERT, distil-BERT, Bangla-BERT, XLM-R models. **A** indicates the actual labels and the wrong predictions are marked in bold.

Example	**MB**	**DB**	**BB**	**XR**	**A**
এই অপরাধ এর একটাই সাজা প্রকাশে ফাঁসি দেয়া (The only punishment for this crime is hanging)	**PoAG**	VeAG	VeAG	VeAG	VeAG
মানুষ এখন হিংস্র জানোয়ার (Humans are now ferocious beasts)	**ReAG**	GeAG	**ReAG**	**VeAG**	GeAG
জয় মা কালি বলা, এটা একটা রোগ (Saying joy ma Kali is a disease)	ReAG	ReAG	ReAG	**GeAG**	ReAG
ফ্যাসিবাদের মূখে গনতন্ত্রের কথা মানায় না (In the face of fascism, democracy is not acceptable)	**GeAG**	**ReAG**	PoAG	PoAG	PoAG

Table 5.6: Few examples that are incorrectly classified by the proposed weighted ensemble model. P and A denotes predicted and actual labels respectively.

Text	**A**	**P**
নিজের মা বোনদের সম্মান কর (Respect your mother and sisters)	NoAG	GeAG
পুলিশ পাহারার বাহিরে এসে কথা বলে দেখ তখন বুঝবে আমরা কারা (Come out of the police guard and talk then you will understand who we are.)	VeAG	PoAG
ধর্ম মানুষে মানুষে বিভেদ সৃষ্টি করে সব অশান্তির পেছনে কারন ধর্ম (Religion is the cause of all the unrest that divides people)	ReAG	NoAG
এসব ফালতু মেয়েদের জন্য রাস্তাঘাটে সমস্যা হয় (These bad girls create problems in the road)	GeAG	VeAG
বাংলাদেশে নির্বাচন আর প্রহসন একই কথা। সরকার ই সব ক্ষমতার মালিক। (Election and farce are the same thing in Bangladesh. The government owns all the power)	PoAG	NoAG

Analysis of the incorrect predictions revealed that it is arduous to identify those texts that implicitly propagate or express aggression. Such instances do not contain any aggressive references or words; therefore, it is difficult to flag them. On the other hand, some texts sarcastically use aggressive words with no intention to harm or do evil, but the model wrongly classifies them as aggressive. It is challenging to identify and classify such text samples from the surface level analysis without understanding the context. Moreover, some words are frequent in both aggressive and non-aggressive classes. The presence of such words in a text creates confusion and makes the task more complicated. Contextual analysis of aggressive texts, adding their meta-information, and more training data might improve the classification performance of the proposed model.

5.5 Comparison with existing methods

As per this work exploration, no significant work has been conducted to categorize aggressive texts into fine-grained classes, including dataset development in Bengali. Therefore, this research adopted several recent techniques that have been explored on similar tasks in other language's datasets. For consistency, previous methods [124, 65, 125, 126] have implemented on the developed dataset (i.e., BAD) and compared their performance with the proposed technique. Table 5.7 shows the comparison in terms of weighted f_1-score for coarse-grained and fined-grained classification.

Table 5.7: Comparison between proposed and existing techniques in terms of weighted f_1-score of the models on BAD.

Technique	**Coarse-grained**	**Fine-grained**
Kumari et al. [124]	90.54	81.20
Ranasinghe et al. [65]	92.71	91.45
Baruah et al. [125]	89.31	84.01
Nayel et al. [126]	89.89	85.98
Proposed	93.43	93.11

Kumari et al. [124] develop a model on TRAC-2 dataset [32] using LSTM and FastText embedding to classify aggressive Bengali texts. One layer of LSTM with 192 unit is used where dropout and recurrent dropout value set to 0.2. We obtained a WF score of 90.54% (coarse-grained) and 81.20% (fine-grained) by mimicking their architecture. On the same dataset, Ranasinghe et al. [65] applied inter-language transfer strategy along with XLM-R. After employing XLM-R, the system achieved a WF score of 92.71% and 91.45%. The other two works [125, 126] used SVM with tf-idf and other parameter combination to classify aggressive and offensive languages. These methods gained lower accuracy on BAD than other methods in both classification tasks. The comparative analysis shows that the proposed technique outperformed the existing techniques by acquiring the highest weighted f_1-score of 93.43% and 93.11% in coarse and fine-grained classification, respectively.

5.6 Impact of Execution Time and Trainable Parameters

Since the models explored in this work is computationally intensive, therefore a brief analysis of their complexities presented for better understanding. Table 5.8 provides the number of trainable parameters of the deep neural networks and transformer models as well as reports their execution time on this experimental setup. As the training set of coarse-grained classification is much bigger, its complexity is also higher than the fine-grained classification task. Although the pre-trained models performed better, their execution time is 4-5 times higher than the custom deep neural networks. Among the models, XLM-R has the highest number of parameters and also requires the highest execution time.

Table 5.8: Computational complexity of deep neural networks and transformer models. Execution time reported here is for completing 30 epochs (deep neural networks) and 20 epochs (transformers) in the GPU facilitated Google colab platform.

	Task-A		**Task-B**	
Method	**Trainable Parameters**	**Execution Time**	**Trainable Parameters**	**Execution Time**
CNN (C)	3564450	13min 6s	1664196	7min 6s
BiLSTM (B)	3899106	19min 45s	1999364	5min 6s
C+B	3678690	16min 18s	1778820	4min 12s
CNN (FastText)	10641250	35min 6s	4940996	10min 6s
BiLSTM (FastText)	11103906	42min 12s	5404164	10min 1s
C+B (FastText)	10755490	40min 18s	5055620	9min 9s
m-BERT	167357954	1h 16min 29s	167359492	35min 20s
distil-BERT	135326210	46min 35s	135327748	20min 2s
Bangla-BERT	164398082	1h 15min 40s	164398082	36min 40s
XLM-R	278045186	1h 33min 1s	278046724	42min 13s

Chapter 6

Conclusion and Future Recommendations

In this chapter we briefly summarized the major outcomes of this research and points out some future directions to work on. Finally, publications which are related to this thesis are listed. This work presents a manually annotated novel Bengali aggressive text dataset ('BAD') and empirically validates it. The BAD comprises 14158 texts accumulated from various social media sources and labelled adopting a two-level hierarchical annotation schema. Level-A has two coarse-grained (AG, NoAG), and level-B has four fine-grained (ReAG, PoAG, VeAG, GeAG) classes. Various machine learning (LR, RF, NB, SVM), deep learning (CNN, BiLSTM, CNN+BiLSTM) and transformer (m-BERT, distil-BERT, Bangla-BERT, XLM-R) models are applied on BAD to examine their performance. After analyzing these models' outcomes, this work proposed a weighted ensemble architecture. The proposed technique has the ability to adjust the softmax probabilities of the participating models depending on their previous outcomes on the dataset. This technique outperformed the average ensemble and other baselines by obtaining the maximum weighted f_1-score of 0.9343 in coarse-grained classification. It also achieved the highest weighted f_1-score in fine-grained classes: ReAG (0.95), PoAG (0.97), VeAG(0.92) and GeAG (0.68). Quantitative and qualitative error analysis reveal that it is difficult to identify aggression that expressed implicitly or sarcastically.

6.1 Limitations

Research in natural language processing can have different types of core contributions. The most common are: *dataset-centric* contributions, i.e. new datasets, potentially for new tasks.; *methodology-centric* contributions which are new methods published for existing task or datasets. In this thesis we tried to contribute form both perspectives. A new dataset is presented in an unexplored research avenue concerning Bengali. Moreover, a dynamic weighting technique is proposed which helps to automatically readdress the softmax probabilities of the classifiers. Since, it is the first attempt to classify aggressive texts in Bengali, our work has some limitations. (i) A text can simultaneously express multiple types of aggression, this work did not consider the overlap of multiple aggression classes, (ii) the 'BAD' do not contain any Bangla-English code-mixed texts or Banglish (code-switched) texts although it is a prevalent phenomena in social media, (iii) limited data samples in the fine-grained classes.

6.2 Future Recommendations

The main purpose of our work was to develop a system for detecting aggressive Bengali texts using supervised learning techniques. Here we give a Bengali text document as input and it will give us a feedback whether the text is aggressive or not. Furthermore, it also detect the the fined-grained domain of the aggression. To address the limitations and improve the performance of the system, in future, we plan to work in the following areas,

- Identification of mixed aggression by adding more diverse data in fine-grained categories.
- It will be interesting to investigate how the models perform if we transfer knowledge from resource-rich language's using cross-lingual and multilingual transferring techniques.
- Other aspects to explore are code-mixing and code-switching of Bengali and English/other languages. Moreover handling texts that express aggression in sarcastic way.
- The proposed model performance can be investigated with Twitter data with more classes such as racial and geographic aggression.
- Our overarching plan is to develop a web based system which can filter different online posts, writings and alert about aggressive activities.

6.3 Remarks

The Fourth Industrial Revolution is about more than just technology-driven change; it is an opportunity to help everyone, including leaders, policy-makers and people from all income groups and nations, to harness converging technologies to create an inclusive, human-centred future. The real opportunity is to look beyond technology and find ways to give the most significant number of people the ability to impact their families, organizations and communities positively. The proposed project can be considered as the innovation of Industry 4.0. Successful use of this project will have an impact in many ways which is illustrated in the following.

- **Direct customers/beneficiaries of the project**

    1. National security agencies.
    2. Authorities that works with virtual social harassment and privacy threats in social media.
    3. Law enforcement agencies for predicting criminal activities.
    4. Research community regarding Bengali language processing.
    5. Related departments and research organization of the university.

- **Organizational outcomes:** Designing and implementing a computational model to identify suspicious Bengali texts will add a new dimension to the Bengali language processing research activities of the university. Sharing hardware and software support with other universities or research institutes, and exchanging expertise will accelerate the current research on Bengali language processing at the university. A member of this project (Omar Sharif) has completed his MSc. Engg. in Computer Science under this project. Such outcomes will help to facilitate research in the university.

- **National impacts:** As most of our communications are text-based if we can predict whether a Bangla text is either suspicious or not suspicious, it will be very helpful for our law enforcement agencies to find the perpetrators and stop terrorist events. The developed system can be used to eliminate the process of manually checking the whole conversation, which is time-consuming and costly. The developed system will reduce the labour of the people working to ensure national security. Finally, such system can also help predict criminal activities, reduce virtual social harassment in social media, mitigate national privacy threats and overall ensure our national security by detecting suspicious communications through text.

The target of this project was to the development of state-of-the-art research to solve technological problems. The developed system has made a significant contribution to the ICT and high-tech industries which will be a critical endeavor toward materialization the dreams of digital Bangladesh.

6.4 List of Publications

The following publication is a direct consequence of the research carried out during the elaboration of the thesis, and give an idea of the progression that has been achieved.

1. **Sharif, O.** & Hoque, M.M., "Tackling Cyber-Aggression: Identification and Fine-Grained Categorization of Aggressive Texts on Social Media using Weighted Ensemble of Transformers", Neurocomputing, 2021.

2. **Sharif, O.** & Hoque, M.M., "Identification and classification of textual aggressionin social media: Resource creation and evaluation", in Combating Online Hostile Posts in Regional Languages during Emergency Situation, pp. 1–12, Springer Nature Switzerland AG, 2021.

3. **Sharif, O.** & Hoque, M.M., "Align and Conquer: An Ensemble Approach to Classify Aggressive Texts from Social Media," in International Conference on Signal Processing, Information, Communication and Systems (SPICSCON), pp. 1–6, 2021.

4. **Sharif, O.,** Hossain, E., & Hoque, M.M., "Offensive language detection from multilingual code-mixed text using transformers", in Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pp. 255–261, ACL, Apr. 2021.

5. Parvin, T., **Sharif, O.**, & Hoque, M.M., "Multi-class Textual Emotion Categorization using Ensemble of Convolutional and Recurrent Neural Network", SN Computer Science 3, 62 (2022).

6. Mamun, M.M.R., **Sharif, O.** & Hoque, M.M., "Classification of Textual Sentiment Using Ensemble Technique", SN Computer Science 3, 49 (2022).

7. Hossain, E., **Sharif, O.,** & Hoque, M.M., "Investigating Visual and Textual Features to Identify Trolls from Multimodal Social Media Memes", in Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pp. 300-306, ACL, Apr. 2021.

Bibliography

[1] Z. Waseem, W. H. K. Chung, D. Hovy, and J. Tetreault, eds., *Proceedings of the First Workshop on Abusive Language Online*, (Vancouver, BC, Canada), Association for Computational Linguistics, Aug. 2017.

[2] J. Salminen, H. Almerekhi, M. Milenković, S. gyo Jung, J. An, H. Kwak, and B. Jansen, "Anatomy of online hate: Developing a taxonomy and machine learning models for identifying and classifying hate in online news media," 2018.

[3] D. U. Patton, J. S. Hong, M. Ranney, S. Patel, C. Kelley, R. Eschmann, and T. Washington, "Social media as a vector for youth violence: A review of the literature," *Computers in Human Behavior*, vol. 35, pp. 548–553, 2014.

[4] R. Bannink, S. Broeren, P. M. van de Looij – Jansen, F. G. de Waart, and H. Raat, "Cyber and traditional bullying victimization as a risk factor for mental health problems and suicidal ideation in adolescents," *PLOS ONE*, vol. 9, pp. 1–7, 04 2014.

[5] R. A. Bonanno and S. Hymel, "Cyber bullying and internalizing difficulties: Above and beyond the impact of traditional forms of bullying," *Journal of youth and adolescence*, vol. 42, no. 5, pp. 685–697, 2013.

[6] A. Bhattacharjee, T. Hasan, K. Samin, M. S. Rahman, A. Iqbal, and R. Shahriyar, "Banglabert: Combating embedding barrier for low-resource language understanding," 2021.

[7] B. Haddad, Z. Orabe, A. Al-Abood, and N. Ghneim, "Arabic offensive language detection with attention-based deep neural networks," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, (Marseille, France), pp. 76–81, European Language Resource Association, May 2020.

[8] M. Ravikiran, A. E. Muljibhai, T. Miyoshi, H. Ozaki, Y. Koreeda, and S. Masayuki, "Hitachi at semeval-2020 task 12: Offensive language identification with noisy labels using statistical sampling and post-processing," 2020.

[9] O. Sharif, E. Hossain, and M. M. Hoque, "NLP-CUET@DravidianLangTech-EACL2021: Offensive language detection from multilingual code-mixed text using transformers," in *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, (Kyiv), pp. 255–261, Association for Computational Linguistics, Apr. 2021.

[10] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Benchmarking aggression identification in social media," in *Proceedings of the First Workshop on*

Trolling, Aggression and Cyberbullying (TRAC-2018), (Santa Fe, New Mexico, USA), pp. 1–11, Association for Computational Linguistics, Aug. 2018.

[11] N. Nikhil, R. Pahwa, M. K. Nirala, and R. Khilnani, "LSTMs with attention for aggression detection," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, (Santa Fe, New Mexico, USA), pp. 52–57, Association for Computational Linguistics, Aug. 2018.

[12] N. Safi Samghabadi, P. Patwa, S. PYKL, P. Mukherjee, A. Das, and T. Solorio, "Aggression and misogyny detection using BERT: A multi-task approach," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 126–131, European Language Resources Association (ELRA), May 2020.

[13] P. Fortuna, J. Soler-Company, and L. Wanner, "How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets?," *Information Processing & Management*, vol. 58, no. 3, p. 102524, 2021.

[14] L. Gao and R. Huang, "Detecting online hate speech using context aware models," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, (Varna, Bulgaria), pp. 260–266, INCOMA Ltd., Sept. 2017.

[15] T. Mandl, S. Modha, A. Kumar M, and 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*, FIRE 2020, (New York, NY, USA), p. 29–32, Association for Computing Machinery, 2020.

[16] S. T. Roberts, J. Tetreault, V. Prabhakaran, and Z. Waseem, eds., *Proceedings of the Third Workshop on Abusive Language Online*, (Florence, Italy), Association for Computational Linguistics, Aug. 2019.

[17] E. W. Pamungkas and V. Patti, "Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, (Florence, Italy), pp. 363–370, Association for Computational Linguistics, July 2019.

[18] B. Vidgen, A. Harris, D. Nguyen, R. Tromble, S. Hale, and H. Margetts, "Challenges and frontiers in abusive content detection," in *Proceedings of the Third Workshop on Abusive Language Online*, (Florence, Italy), pp. 80–93, Association for Computational Linguistics, Aug. 2019.

[19] A. G. D'Sa, I. Illina, and D. Fohr, "Towards non-toxic landscapes: Automatic toxic comment detection using DNN," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 21–25, European Language Resources Association (ELRA), May 2020.

[20] M. Karan and J. Šnajder, "Preemptive toxic language detection in Wikipedia comments using thread-level context," in *Proceedings of the Third Workshop on Abusive Language Online*, (Florence, Italy), pp. 129–134, Association for Computational Linguistics, Aug. 2019.

[21] S. Bhattacharya, S. Singh, R. Kumar, A. Bansal, A. Bhagat, Y. Dawer, B. Lahiri, and A. K. Ojha, "Developing a multilingual annotated corpus of misogyny and aggression," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 158–168, European Language Resources Association (ELRA), May 2020.

[22] S. Sharifirad and S. Matwin, "When a tweet is actually sexist. a more comprehensive classification of different online harassment categories and the challenges in nlp," 2019.

[23] T. Mihaylov, G. Georgiev, and P. Nakov, "Finding opinion manipulation trolls in news community forums," in *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, (Beijing, China), pp. 310–314, Association for Computational Linguistics, July 2015.

[24] L. G. Mojica de la Vega and V. Ng, "Modeling trolling in social media conversations," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.

[25] M. Dadvar, D. Trieschnigg, and F. de Jong, "Experts and machines against bullies: A hybrid approach to detect cyberbullies," in *Advances in Artificial Intelligence* (M. Sokolova and P. van Beek, eds.), (Cham), pp. 275–281, Springer International Publishing, 2014.

[26] M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong, "Improving cyberbullying detection with user context," in *Proceedings of the 35th European Conference on Advances in Information Retrieval*, ECIR'13, (Berlin, Heidelberg), p. 693–696, Springer-Verlag, 2013.

[27] J. Pavlopoulos, N. Thain, L. Dixon, and I. Androutsopoulos, "ConvAI at SemEval-2019 task 6: Offensive language identification and categorization with perspective and BERT," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, (Minneapolis, Minnesota, USA), pp. 571–576, Association for Computational Linguistics, June 2019.

[28] G. Wiedemann, S. M. Yimam, and C. Biemann, "UHH-LT at SemEval-2020 task 12: Fine-tuning of pre-trained transformer networks for offensive language detection," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, (Barcelona (online)), pp. 1638–1644, International Committee for Computational Linguistics, Dec. 2020.

[29] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval)," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, (Minneapolis, Minnesota, USA), pp. 75–86, Association for Computational Linguistics, June 2019.

[30] S. T. Aroyehun and A. Gelbukh, "Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, (Santa Fe, New Mexico, USA), pp. 90–97, Association for Computational Linguistics, Aug. 2018.

[31] J. Risch and R. Krestel, “Bagging BERT models for robust aggression identification,” in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 55–61, European Language Resources Association (ELRA), May 2020.

[32] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, “Evaluating aggression identification in social media,” in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 1–5, European Language Resources Association (ELRA), May 2020.

[33] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, “Predicting the type and target of offensive posts in social media,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 1415–1420, Association for Computational Linguistics, June 2019.

[34] A. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis, “Large scale crowdsourcing and characterization of twitter abusive behavior,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, Jun. 2018.

[35] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, May 2017.

[36] P. Mathur, R. Shah, R. Sawhney, and D. Mahata, “Detecting offensive tweets in Hindi-English code-switched language,” in *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, (Melbourne, Australia), pp. 18–26, Association for Computational Linguistics, July 2018.

[37] R. Kumar, A. N. Reganti, A. Bhatia, and T. Maheshwari, “Aggression-annotated corpus of Hindi-English code-mixed data,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.

[38] M. Bhardwaj, M. S. Akhtar, A. Ekbal, A. Das, and T. Chakraborty, “Hostility detection dataset in hindi,” 2020.

[39] H. Mulki, H. Haddad, C. Bechikh Ali, and H. Alshabani, “L-HSAB: A Levantine Twitter dataset for hate speech and abusive language,” in *Proceedings of the Third Workshop on Abusive Language Online*, (Florence, Italy), pp. 111–118, Association for Computational Linguistics, Aug. 2019.

[40] H. Mubarak, K. Darwish, and W. Magdy, “Abusive language detection on Arabic social media,” in *Proceedings of the First Workshop on Abusive Language Online*, (Vancouver, BC, Canada), pp. 52–56, Association for Computational Linguistics, Aug. 2017.

[41] S. Hassan, Y. Samih, H. Mubarak, and A. Abdelali, “ALT at SemEval-2020 task 12: Arabic and English offensive language identification in social media,” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, (Barcelona (online)), pp. 1891–1897, International Committee for Computational Linguistics, Dec. 2020.

[42] M. Á. Á. Carmona, E. Guzmán-Falcón, M. Montes-y-Gómez, H. J. Escalante, L. V. Pineda, V. Reyes-Meza, and A. R. Sulayes, “Overview of MEX-A3T at ibereval 2018: Authorship and aggressiveness analysis in mexican spanish tweets,” in *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018* (P. Rosso, J. Gonzalo, R. Martínez, S. Montalvo, and J. C. de Albornoz, eds.), vol. 2150 of *CEUR Workshop Proceedings*, pp. 74–96, CEUR-WS.org, 2018.

[43] M. Graff, S. Miranda-Jiménez, E. S. Tellez, D. Moctezuma, V. Salgado, J. Ortiz-Bejar, and C. N. Sánchez, “INGEOTEC at MEX-A3T: author profiling and aggressiveness analysis in twitter using μtc and evomsa,” in *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018* (P. Rosso, J. Gonzalo, R. Martínez, S. Montalvo, and J. C. de Albornoz, eds.), vol. 2150 of *CEUR Workshop Proceedings*, pp. 128–133, CEUR-WS.org, 2018.

[44] M. Wiegand, “Overview of the germeval 2018 shared task on the identification of offensive language,” 2018. Online available: https://epub.oeaw.ac.at/?arp=0x003a10d2 - Last access:11.3.2021.

[45] J. M. Struß, M. Siegel, J. Ruppenhofer, M. Wiegand, and M. Klenner, “Overview of germeval task 2, 2019 shared task on the identification of offensive language,” Preliminary proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019), October 9 – 11, 2019 at Friedrich-Alexander-Universität Erlangen-Nürnberg, (München [u.a.]), pp. 352 – 363, German Society for Computational Linguistics & Language Technology und Friedrich-Alexander-Universität Erlangen-Nürnberg, 2019.

[46] J. A. Leite, D. Silva, K. Bontcheva, and C. Scarton, “Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis,” in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, (Suzhou, China), pp. 914–924, Association for Computational Linguistics, Dec. 2020.

[47] R. de Pelle and V. Moreira, “Offensive comments in the brazilian web: a dataset and baseline results,” in *Anais do VI Brazilian Workshop on Social Network Analysis and Mining*, (Porto Alegre, RS, Brasil), SBC, 2017.

[48] P. Fortuna, J. Rocha da Silva, J. Soler-Company, L. Wanner, and S. Nunes, “A hierarchically-labeled Portuguese hate speech dataset,” in *Proceedings of the Third Workshop on Abusive Language Online*, (Florence, Italy), pp. 94–104, Association for Computational Linguistics, Aug. 2019.

[49] S. Mishra, S. Prasad, and S. Mishra, “Multilingual joint fine-tuning of transformer models for identifying trolling, aggression and cyberbullying at TRAC 2020,” in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 120–125, European Language Resources Association (ELRA), May 2020.

[50] D. Gordeev and O. Lykova, “BERT of all trades, master of some,” in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 93–98, European Language Resources Association (ELRA), May 2020.

[51] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, and Ç. Çöltekin, “SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020),” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, (Barcelona (online)), pp. 1425–1447, International Committee for Computational Linguistics, Dec. 2020.

[52] S. Wang, J. Liu, X. Ouyang, and Y. Sun, “Galileo at SemEval-2020 task 12: Multi-lingual learning for offensive language identification using pre-trained language models,” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, (Barcelona (online)), pp. 1448–1455, International Committee for Computational Linguistics, Dec. 2020.

[53] H. Ahn, J. Sun, C. Y. Park, and J. Seo, “NLPDove at SemEval-2020 task 12: Improving offensive language detection with cross-lingual transfer,” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, (Barcelona (online)), pp. 1576–1586, International Committee for Computational Linguistics, Dec. 2020.

[54] E. Fersini, P. Rosso, and M. Anzovino, “Overview of the task on automatic misogyny identification at ibereval 2018.,” *IberEval@ SEPLN*, vol. 2150, pp. 214–228, 2018.

[55] D. Fišer, R. Huang, V. Prabhakaran, R. Voigt, Z. Waseem, and J. Wernimont, eds., *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, (Brussels, Belgium), Association for Computational Linguistics, Oct. 2018.

[56] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, and M. Sanguinetti, “SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter,” in *Proceedings of the 13th International Workshop on Semantic Evaluation*, (Minneapolis, Minnesota, USA), pp. 54–63, Association for Computational Linguistics, June 2019.

[57] B. Gambäck and U. K. Sikdar, “Using convolutional neural networks to classify hate-speech,” in *Proceedings of the First Workshop on Abusive Language Online*, (Vancouver, BC, Canada), pp. 85–90, Association for Computational Linguistics, Aug. 2017.

[58] S. Tulkens, L. Hilte, E. Lodewyckx, B. Verhoeven, and W. Daelemans, “A dictionary-based approach to racism detection in dutch social media,” 2016.

[59] J. J. Andrew, “JudithJeyafreedaAndrew@DravidianLangTech-EACL2021:offensive language detection for Dravidian code-mixed YouTube comments,” in *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, (Kyiv), pp. 169–174, Association for Computational Linguistics, Apr. 2021.

[60] M. R. Karim, S. K. Dey, and B. R. Chakravarthi, “Deephateexplainer: Explainable hate speech detection in under-resourced bengali language,” 2021.

[61] N. Romim, M. Ahmed, H. Talukder, and M. S. Islam, “Hate speech detection in the bengali language: A dataset and its baseline evaluation,” 2020.

[62] O. Sharif and M. M. Hoque, “Automatic detection of suspicious bangla text using logistic regression,” in *Intelligent Computing and Optimization* (P. Vasant, I. Zelinka, and G.-W. Weber, eds.), (Cham), pp. 581–590, Springer International Publishing, 2020.

[63] E. A. Emon, S. Rahman, J. Banarjee, A. K. Das, and T. Mittra, “A deep learning approach to detect abusive bengali text,” in *2019 7th International Conference on Smart Computing Communications (ICSCC)*, pp. 1–5, 2019.

[64] M. F. Mridha, M. A. H. Wadud, M. A. Hamid, M. M. Monowar, M. Abdullah-Al-Wadud, and A. Alamri, “L-boost: Identifying offensive texts from social media post in bengali,” *IEEE Access*, pp. 1–1, 2021.

[65] T. Ranasinghe and M. Zampieri, “Multilingual offensive language identification with cross-lingual embeddings,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 5838–5844, Association for Computational Linguistics, Nov. 2020.

[66] O. Sharif, M. M. Hoque, A. S. M. Kayes, R. Nowrozy, and I. H. Sarker, “Detecting suspicious texts using machine learning techniques,” *Applied Sciences*, vol. 10, no. 18, 2020.

[67] P. Chakraborty and M. H. Seddiqui, “Threat and abusive language detection on social media in bengali language,” in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pp. 1–6, 2019.

[68] O. Sharif and M. M. Hoque, “Identification and classification of textual aggression in social media: Resource creation and evaluation,” in *Combating Online Hostile Posts in Regional Languages during Emergency Situation* (T. Chakraborty and et al., eds.), pp. 1–12, Springer Nature Switzerland AG, 2021.

[69] C. A. Anderson and B. J. Bushman, “Human aggression,” *Annual Review of Psychology*, vol. 53, no. 1, pp. 27–51, 2002.

[70] M. J. Díaz-Torres, P. A. Morán-Méndez, L. Villasenor-Pineda, M. Montes-y Gómez, J. Aguilera, and L. Meneses-Lerín, “Automatic detection of offensive language in social media: Defining linguistic criteria to build a Mexican Spanish dataset,” in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 132–136, European Language Resources Association (ELRA), May 2020.

[71] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, “Abusive language detection in online user content,” in *Proceedings of the 25th International Conference on World Wide Web*, WWW ’16, (Republic and Canton of Geneva, CHE), p. 145–153, International World Wide Web Conferences Steering Committee, 2016.

[72] Youtube, “Harmful or dangerous content policy.” Available online: https://support.google.com/youtube/answer/2801939/ (accessed on 2 October 2020).

[73] COE, “Hate speech and violence.” Available online: https://www.coe.int/en/web/european-commission-against-racism-and-intolerance/hate-speech-and-violence/ (accessed on 3 October 2020).

[74] P. Fortuna and S. Nunes, “A survey on automatic detection of hate speech in text,” *ACM Comput. Surv.*, vol. 51, July 2018.

[75] A. Roy, P. Kapil, K. Basak, and A. Ekbal, “An ensemble approach for aggression identification in English and Hindi text,” in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, (Santa Fe, New Mexico, USA), pp. 66–73, Association for Computational Linguistics, Aug. 2018.

[76] R. A. Baron and D. R. Richardson, *Human aggression.* Springer Science & Business Media, 2004.

[77] A. H. Buss, *The psychology of aggression.* Wiley, 1961.

[78] Z. Waseem, T. Davidson, D. Warmsley, and I. Weber, “Understanding abuse: A typology of abusive language detection subtasks,” in *Proceedings of the First Workshop on Abusive Language Online*, (Vancouver, BC, Canada), pp. 78–84, Association for Computational Linguistics, Aug. 2017.

[79] R. Kumar, B. Lahiri, and A. K. Ojha, “Aggressive and offensive language identification in hindi, bangla, and english: A comparative study,” *SN Computer Science*, vol. 2, no. 1, pp. 1–20, 2021.

[80] S. Weingartner and L. Stahel, “Online aggression from a sociological perspective: An integrative view on determinants and possible countermeasures,” in *Proceedings of the Third Workshop on Abusive Language Online*, (Florence, Italy), pp. 181–187, Association for Computational Linguistics, Aug. 2019.

[81] S. Srivastava and P. Khurana, “Detecting aggression and toxicity using a multi dimension capsule network,” in *Proceedings of the Third Workshop on Abusive Language Online*, (Florence, Italy), pp. 157–162, Association for Computational Linguistics, Aug. 2019.

[82] X. Zhou, M. Sap, S. Swayamdipta, N. A. Smith, and Y. Choi, “Challenges in automated debiasing for toxic language detection,” 2021.

[83] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, “Convolutional neural networks for toxic comment classification,” 2018.

[84] P. Fortuna, J. Soler, and L. Wanner, “Toxic, hateful, offensive or abusive? what are we really classifying? an empirical analysis of hate speech datasets,” in *Proceedings of the 12th Language Resources and Evaluation Conference*, (Marseille, France), pp. 6786–6794, European Language Resources Association, May 2020.

[85] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, “Deep learning for hate speech detection in tweets,” in *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW ’17 Companion, (Republic and Canton of Geneva, CHE), p. 759–760, International World Wide Web Conferences Steering Committee, 2017.

[86] P. Kapil and A. Ekbal, “A deep neural network based multi-task learning approach to hate speech detection,” *Knowledge-Based Systems*, vol. 210, p. 106458, 2020.

[87] S. Akiwowo, B. Vidgen, V. Prabhakaran, and Z. Waseem, eds., *Proceedings of the Fourth Workshop on Online Abuse and Harms*, (Online), Association for Computational Linguistics, Nov. 2020.

[88] M. O. Ibrohim and I. Budi, “Multi-label hate speech and abusive language detection in Indonesian Twitter,” in *Proceedings of the Third Workshop on Abusive Language Online*, (Florence, Italy), pp. 46–57, Association for Computational Linguistics, Aug. 2019.

[89] N. Safi Samghabadi, A. Hatami, M. Shafaei, S. Kar, and T. Solorio, “Attending the emotions to detect online abusive language,” in *Proceedings of the Fourth Workshop on Online Abuse and Harms*, (Online), pp. 79–88, Association for Computational Linguistics, Nov. 2020.

[90] C. Van Hee, E. Lefever, B. Verhoeven, J. Mennes, B. Desmet, G. De Pauw, W. Daelemans, and V. Hoste, “Detection and fine-grained classification of cyberbullying events,” in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, (Hissar, Bulgaria), pp. 672–680, INCOMA Ltd. Shoumen, BULGARIA, Sept. 2015.

[91] B. Vidgen and L. Derczynski, “Directions in abusive language training data, a systematic review: Garbage in, garbage out,” *PLOS ONE*, vol. 15, pp. 1–32, 12 2021.

[92] Facebook, “What is public information in facebook.” Available online: https://web.facebook.com/help/203805466323736?_rdc=1&_rdr.

[93] Facebook, “Copyright.” Available online: https://web.facebook.com/help/1020633957973118?_rdc=1&_rdr.

[94] Z. Waseem, “Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter,” in *Proceedings of the First Workshop on NLP and Computational Social Science*, (Austin, Texas), pp. 138–142, Association for Computational Linguistics, Nov. 2016.

[95] E. M. Bender and B. Friedman, “Data statements for natural language processing: Toward mitigating system bias and enabling better science,” *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 587–604, 2018.

[96] R. Binns, M. Veale, M. Van Kleek, and N. Shadbolt, “Like trainer, like bot? inheritance of bias in algorithmic content moderation,” in *Social Informatics* (G. L. Ciampaglia, A. Mashhadi, and T. Yasseri, eds.), (Cham), pp. 405–415, Springer International Publishing, 2017.

[97] L. Derczynski, K. Bontcheva, and I. Roberts, “Broad Twitter corpus: A diverse named entity recognition resource,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, (Osaka, Japan), pp. 1169–1179, The COLING 2016 Organizing Committee, Dec. 2016.

[98] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki, “Measuring the reliability of hate speech annotations: The case of the european refugee crisis,” *arXiv preprint arXiv:1701.08118*, 2017.

[99] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[100] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[101] T. Tokunaga and I. Makoto, "Text categorization based on weighted inverse document frequency," in *Special Interest Groups and Information Process Society of Japan (SIG-IPSJ*, Citeseer, 1994.

[102] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," 2018.

[103] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[104] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[105] P. Kapil, A. Ekbal, and D. Das, "NLP at SemEval-2019 task 6: Detecting offensive language using neural networks," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, (Minneapolis, Minnesota, USA), pp. 587–592, Association for Computational Linguistics, June 2019.

[106] O. Sharif, E. Hossain, and M. M. Hoque, "Combating hostility: Covid-19 fake news and hostile post detection in social media," 2021.

[107] S. Madisetty and M. Sankar Desarkar, "Aggression detection in social media using deep neural networks," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, (Santa Fe, New Mexico, USA), pp. 120–127, Association for Computational Linguistics, Aug. 2018.

[108] D. J. C. MacKay, "Hyperparameters: optimize, or integrate out?," in *Maximum Entropy and Bayesian Methods: Santa Barbara, California, U.S.A., 1993*, vol. 62, pp. 43–60, Springer, Dordrecht, 1996.

[109] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[110] A. Yenter and A. Verma, "Deep cnn-lstm with combined kernels from multiple branches for imdb review sentiment analysis," in *IEEE A. Ubi. Comp., Elect. & Mob. Com. Conf. (UEMCON)*, pp. 540–546, IEEE, 2017.

[111] N. Kalchbrenner, I. Danihelka, and A. Graves, "Grid long short-term memory," *arXiv preprint arXiv:1507.01526*, 2015.

[112] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[113] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short*

Papers), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[114] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter,” 2020.

[115] S. Sarker, “Banglabert: Bengali mask language model for bengali language understading,” 2020.

[116] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 8440–8451, Association for Computational Linguistics, July 2020.

[117] A. S. Maiya, “ktrain: A low-code library for augmented machine learning,” 2020.

[118] V. Bhatnagar, P. Kumar, S. Moghili, and P. Bhattacharyya, “Divide and conquer: An ensemble approach for hostile post detection in hindi,” 2021.

[119] S. Tawalbeh, M. Hammad, and M. AL-Smadi, “KEIS@JUST at SemEval-2020 task 12: Identifying multilingual offensive tweets using weighted ensemble and fine-tuned BERT,” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, (Barcelona (online)), pp. 2035–2044, International Committee for Computational Linguistics, Dec. 2020.

[120] S. Gundapu and R. Mamidi, “Transformer based automatic covid-19 fake news detection system,” 2021.

[121] S. M. S.-U.-R. Shifath, M. F. Khan, and M. S. Islam, “A transformer based approach for fighting covid-19 fake news,” 2021.

[122] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017.

[123] Z. Waseem and D. Hovy, “Hateful symbols or hateful people? predictive features for hate speech detection on Twitter,” in *Proceedings of the NAACL Student Research Workshop*, (San Diego, California), pp. 88–93, Association for Computational Linguistics, June 2016.

[124] K. Kumari and J. P. Singh, “AI_ML_NIT_Patna @ TRAC - 2: Deep learning approach for multi-lingual aggression identification,” in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 113–119, European Language Resources Association (ELRA), May 2020.

[125] A. Baruah, K. Das, F. Barbhuiya, and K. Dey, “Aggression identification in English, Hindi and Bangla text using BERT, RoBERTa and SVM,” in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, (Marseille, France), pp. 76–82, European Language Resources Association (ELRA), May 2020.

[126] H. Nayel, “NAYEL at SemEval-2020 task 12: TF/IDF-based approach for automatic offensive language detection in Arabic tweets,” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, (Barcelona (online)), pp. 2086–2089, International Committee for Computational Linguistics, Dec. 2020.

Appendix

Table 1: Few examples that are incorrectly classified by the proposed weighted ensemble model. P and A denotes predicted and actual labels respectively.

Text	**A**	**P**
জয় মা কালি এটা একটা রোগ	ReAG	GeAG
নারীবাদী রোকেয়া বলেছে ধর্ম গ্রন্থগুলি পুরুষের বানানো অথচ এই মহিলাকে মুসলিম বলে বলে পাঠ্য বইয়ের মাধ্যমে ভবিষ্যত প্রজন্মের ব্রেইন ওয়াশ করা হচ্ছে	GeAG	ReAG
সৃজিতকে দেখলেই মনে হয় খারাপ মানুষ	VeAG	PoAG
বাংলাদেশে নির্বাচন আর প্রহসন একই কথা। সরকার ই সব ক্ষমতার মালিক।	PoAG	NoAG
ধর্ম মানুষে মানুষে বিভেদ সৃষ্টি করে সব অশান্তির পেছনে কারন ধর্ম	ReAG	NoAG
এর জন্যই তর কপালে স্বামী নাই	GeAG	VeAG
দেশকে এই সরকারের হাত থেকে মুক্ত করতে হলে যুদ্ধ ছাড়া কোনো উপায় নেই নেই	PoAG	VeAG
পুলিশ পাহারার বাহিরে এসে কথা বলে দেখ তখন বুঝবে আমরা কার	VeAG	PoAG
তসলিমা নাসরিনের যবাই করে ফেলবো	VeAG	ReAG
হাজারো সালাম জানাই শিক্ষকদের, যাদের অবদানে এগিয়ে যাচ্ছে বাংলাদেশ	NoAG	PoAG
নিজের মা বোনদের সম্মান কর	NoAG	GeAG
ধর্ম পালন করা মানে শয়তানের উপাসনা করে। আমাদেরকে ধর্ম থেকে দূরে থাকতে হবে	ReAG	NoAG
এসব ফালতু মেয়েদের জন্য রাস্তাঘাটে সমস্যা হয়	GeAG	VeAG
দেশকে এই সরকারের হাত থেকে মুক্ত করতে হলে যুদ্ধ ছাড়া কোনো উপায় নেই নেই	PoAG	VeAG
তুই দেশের বাইরে আছিস বলে এখনও বেছে আছিস।তোর সাহস থাকলে বাংলাদেশ আয় তোকে সবার সামনে হত্যা করব	VeAG	PoAG
মেয়েদের এত পড়ালেখা করে আর কি লাভ হুদাই টাকা নষ্ট	GeAG	veAG
ভারত থেকে হিন্দু ধর্মের গোড়া উপরে ফেলবো সকল হিন্দুদের হত্যা করে বস্তায় ভরে বঙ্গোপসাগরে ফেলে দেওয়া হবে	ReAG	PoAG
ধন্যবাদ আলজাজিরা চেনেলকে সত্য তুলে ধরার জন্য সুদু মহম্মদ পুরের শীর্ষ সন্ত্রাসী জোসেফই না মিরপুরের শীর্ষ সন্ত্রাসী বিকাশ কেও ছেরে দিয়েছেন বিনা ভোটের সরকার	PoAG	ReAG

Proposed models prediction on few samples are presented in table 1 and table 2[1].

Table 2: Few examples that are correctly classified by the proposed weighted ensemble model.

Text	**Prediction**
রিয়াদের আউট দেখে খুব খারাপ লাগলো।একজন সিনিয়র প্লেয়ার হ্যাট্রিক বলটা এতো বাজে ভাবে শট খেলে	NoAG
বাংলাদেশ থেকে হিন্দুদের বের করে দেয়া হবে খুবই তারাতাড়ি	ReAG
সরকারের পতনের জন্য প্রস্তুত হোন। ইসলামিক দলগুলির সাথে কারা আছেন? লাইক দেন!	PoAG
তর মরার পূজা বন্ধ রাখ তাতে কোন সমস্যা হইব না কিন্তু আমাদের আযান বন্ধ হইলে সমস্যা আছে বুঝঁলি	ReAG
মুসলিম নিজেদের দোষ অন্যের কাঁধে চাপাতে এরা সিদ্ধহস্ত। সকল জঙ্গি সংগঠন এদের সৃষ্ট হলেও এরা সকল দোষ আমেরিকা ইসরায়েল এর কাঁধে চাপিয়ে নিজেরা ধোঁয়া তুলসিপাতা হওয়ার চেষ্টা করে	ReAG
পুলিশ হচ্ছে সরকারের গোলাম বাহিনী , হায়রে ছাত্রলীগ পুলিশ আর কত !!!	PoAG
পর্দা করে সব কাজ করা যায় না। তাই নারীর বাইরে যাওয়াটাও দরকার নাই।	GeAG
মার্কিন প্রেসিডেন্ট নির্বাচন এবং বাংলাদেশী বলদগুলোর প্রতিক্রিয়া হিজরার দল বিম্পি এবং তাদের সমর্থকদের শোকের মাতম।	PoAG
৪ বছর ধরে তোর প্রতিটা পোস্টে গালি দিয়ে যাচ্ছি। তোর কি কোন প্রতিক্রিয়া নাই ভুস্কির পোলা?	VeAG
আরে আমাকেও রাজাকার বলে।কিন্তু আমার শুনতে খুবই ভালো লাগে।কারন আমি জানি বর্তমান সব রাজাকারই ভালো ভদ্র ধার্মিক শিক্ষিত হয়।তাই নিজেকে গর্বিত মনে করি।	PoAG
যে দেশের সংসদেই পতিতা ঢুকে পড়েছে! কাশিমপুর কারাগারে পতিতা ঢুকা অস্বাভাবিক কিছু	PoAG
বাংলার মাটিতে আর কেউ দিতীয় বিয়ে করার আগে একবার হলেও চিন্তা করবেন। বউ কি জিনিস	GeAG
মেয়েদের আটা,ময়দা মাখা মুখটা না দেখে,আকাশের চাদঁ দেখা অনেক তৃপ্তিদায়ক	GeAG
কি রকম হাস্যকর কথা দেশের ডাক্তারদের জন্য বাইরের রাষ্ট্র থেকে চিকিৎসা সরঞ্জাম ভিক্ষা করে আনতে হচ্ছে। আবার এরা নাকি এখান থেকে বাইরের রাষ্ট্রকে চিকিৎসা সরঞ্জাম রপ্তানি করবে।	NoAG

[1] **Disclaimer:** Authors would like to state that the comments/examples referred to the tables presented as they were accumulated from the original source. Authors do not use these examples to hurt individuals or a community. Moreover, authors do not promote aggressive language usage, and this research work aims to mitigate the practice of such language.