2021 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON) | 978-1-6654-7820-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/SPICSCON54707.2021.9885420

Align and Conquer: An Ensemble Approach to Classify Aggressive Texts from Social Media

Omar Sharif
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology
Chittagong-4349, Bangladesh
omar.sharif@cuet.ac.bd

Mohammed Moshiul Hoque
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology
Chittagong-4349, Bangladesh
moshiul_240@cuet.ac.bd

***Abstract*—The phenomenal proliferation of social media platforms has facilitated the spontaneous sharing of expressions, opinions, and emotions in public spaces. Unfortunately, with this rapid rise, these mediums have been repeatedly used to spread propaganda, excite religious and political violence, jeopardize social harmony and disseminate other aggressive content. The pernicious societal effects of such undesired content have become a severe concern for tech giants and government bodies. Studies revealed that the majority of such activities are performed via texts written in regional languages. Developing an automatic system to identify and classify aggressive texts in resource constraint languages like Bengali is monumental. This work proposes an ensemble-based classifier model to compensate for this deficiency. A corpus of 10095 annotated texts (5095 for non-aggressive and 5000 for aggressive) is developed to train the system. Moreover, aggressive texts are classified into four fine-grained categories: political, gendered, verbal and religious using hierarchical annotation schema. This work also investigates 21 standard classifier models developed based on Convolutional Neural Network (CNN), Bidirectional Long Short Term Memory (BiLSTM) and Gated Recurrent Unit (GRU) with different embedding models (i.e., Word2Vec, GloVe, FastText) and ensemble strategies. All the models are trained, tuned and tested on the developed dataset (EATxtC-Extended Aggressive Text Corpus). The experimental result exhibits that the ensemble of CNN and GRU (i.e. CNN+GRU) outperformed the other baseline models with acquiring the highest weighted f_1-score of 89.55% (coarse-grained) and 83.77% (fine-grained).**

***Keywords*—Natural language processing, Aggression corpus, Ensemble technique, Deep learning, Low resource language**

I. Introduction

With the advent of the internet, social media platforms have become a powerful tool to spread information to millions within a short period. However, it is very inauspicious that a group of malign users misuses this power to publicize fake news, disseminate offensive, aggressive and abusive content in these mediums. Often the viciousness of such content is strong enough to trigger violence between the communities in a society [1]. To keep the information ecosystem clean, monitoring and flagging unlawful content in social media are essential. However, it is impossible to check and validate such a massive volume of information manually. Therefore, an automated system is required to analyze this bulk of information and detect the potential content that poses a threat to social stability. Studies have been conducted to identify unlawful contents written in English, Chinese, and other resource-rich languages [2]. Nevertheless, the perpetrators use regional language to carry out such malicious activities. A system trained in another language will not detect aggressive/abusive texts written in Bengali. However, developing an aggressive text classification system in Bengali is challenging due to the lack of benchmark corpora, critical language construct, and shortage of language processing tools. Furthermore, multilingual code-mixing, overlapping aggressive, abusive and hatred texts made the task more complex. This work attempts to bridge this gap by developing an aggressive text classification system. The significant contributions of this work are:

- Present a corpus of 10095 Bengali texts considering two coarse-grained (aggressive, non-aggressive) and four fine-grained (political, gendered, verbal, religious) aggression categories.
- Develop an ensemble model using CNN and GRU to identify and classify aggressive Bengali texts.

II. Related Work

A considerable body of research has been conducted to detect and categorize aggressive, hate, offensive, and abusive content. Davidson et al. [3] developed a hate speech corpus containing 25k English tweets and applied logistic regression to categorize tweets into hate and offensive categories. The system achieved the maximum macro f_1-score of 0.90. An offensive language identification dataset is constructed using hierarchical annotation schema by Zampieri et al. [4]. They employed CNN, SVM, and BiLSTM to detect offensive texts where CNN obtained the highest weighted f_1-score of 0.80. Pelle et al. [5] presented a dataset of 1250 offensive comments considering six fine-grained categories (*sexism, racism, xenophobia, cursing, LGBTQ+phobia, and religious violence.*) SVM with n-gram features achieved the maximum f_1-score of 0.80. Roy et al. [6] adopted a CNN, SVM based ensemble technique to identify aggression in English and Hindi texts. The model obtained 0.5099 (English) and 0.3790 (Hindi) f_1-scores respectively. As per our exploration, no significant work has been conducted to categorize aggressive texts in Bengali. However, few works have studied abusive and offensive text classification schemes. Emon et al. [7] developed a corpus comprising 4700 abusive Bengali texts. They employed the

978-1-6654-7820-5/21/$31.00 ©2021 IEEE

LSTM model for classification and obtained 82.2% accuracy. Sharif et al. [8] introduced an aggressive text corpus having 3888 aggressive and 3703 non-aggressive texts. Aggressive texts are further classified into fine-grained classes. This work employed SVM, CNN, BiLSTM, and CNN+BiLSTM models to classify aggressive texts. This work extended the dataset presented earlier and performed a wide range of experimentation with various embedding techniques.

III. Dataset

An Extended Aggressive Text Corpus ('EATxtC') is utilized for training, validating, and testing all the classifier models. A brief analysis and statistics of the dataset are illustrated in the following subsections.

A. EATxtC: Extended Aggressive Text Corpus

The primary objective of this work is to identify whether a text is aggressive or non-aggressive and subsequently categorize aggressive texts into fine-grained categories. To attain this goal, an aggressive text corpus is developed using a hierarchical annotation schema. In Level-A, there exists two coarse-grained classes: aggressive (AG) and non-aggressive (NoAG), while Level-B has four fine-grained classes: political aggression (PoAG), gendered aggression (GeAG), verbal aggression (VeAG), and religious aggression (ReAG). Standard steps are followed to create an aggression annotated dataset [9]. The definition of aggression categories provided by Sharif et al. [8] is used to ensure the consistency of the instances in the extended dataset. The past study is considered the word embedding models and ensemble strategies. This work comprehensively analyses 29 machine learning and deep learning classifier models on an extended dataset of 10095 samples and acquired superior outcomes.

B. Dataset Statistics

Table I presents a summary of EATxtC, which is partitioned into three mutually exclusive sets for model building and evaluation. In coarse-grained classes, AG has 5000 texts while NoAG contains 5095 texts. ReAG has the highest number of samples (1950) among the fine-grained categories, and GeAG consisting the least number of instances (450).

TABLE I: Number of train, validation and test set instances in coarse and fine-grained categories.

	Level-A		Level-B			
	NoAG	AG	PoAG	GeAG	VeAG	ReAG
Train	4065	4011	883	358	1198	1561
Val	585	525	106	58	171	215
Test	464	445	111	34	131	174
Total	5095	5000	1100	450	1500	1950

The training set is further analyzed to get in-depth insights, and statistics are demonstrated in table II. ReAG has the maximal number of total words (28471), and on average, it contains 18.23 words in each text. Gendered and verbally aggressive texts are shorter, having an average length of 10.57 and 9.75 words per text, respectively. Aggressive texts have fewer words than non-aggressive texts. Table III exhibits the Jaccard similarity scores between each coarse and fine-grained class pair. Approximately half of the most frequent words are familiar in AG and NoAG classes. GeAG has maximum similarity with VeAG, while the PoAG-ReAG pair acquires the highest similarity score of 0.36 amid the fine-grained classes.

TABLE II: Training set statistics

Class	Total words	Unique words	Avg. no. of words per texts
NoAG	82807	18455	20.37
AG	57372	13287	14.30
PoAG	14316	4633	16.21
GeAG	3787	1723	10.57
VeAG	11676	3956	9.75
ReAG	28471	7721	18.23

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

	Level-A		Level-B				
	c1	c2		f1	f2	f3	f4
c1	-	0.46	f1	-	0.32	0.21	0.36
c2	-	-	f2	-	-	0.34	0.33
			f3	-	-	-	0.23

IV. Methodology

This section briefly illustrates the computational methods utilized to develop the aggressive text classification system. Four machine learning models: logistic regression (LR), naive Bayes (NB), support vector machine (SVM), and random forest (RF) is used for preliminary model development. Following this, the combination of several deep learning (DL) models (i.e. CNN, BiLSTM, GRU) with three-word embedding techniques (Word2Vec, GloVe, FastText) have experimented. Finally, the ensemble technique is employed on deep learning models to improve the performance of the predictive models.

Feature Extraction: The preprocessing is carried out to remove errors, duplicates, punctuation, and other inconsistencies from the raw data. The feature extraction performs to transform processed texts into meaningful numerical vectors. A bag of words (BoW) and term frequency-inverse document frequency (TF-IDF) features are extracted with a vocabulary of 12k most frequent words to train ML models. Word2Vec, GloVe, and FastText embedding techniques are employed to extract semantic features for DL classifiers. The embedding matrix is created from pre-trained word vectors, and the optimal embedding dimension is set to 300. For both coarse-grained and fine-grained classification same feature extraction techniques are adopted.

ML Classifiers: Four ML models are employed to build the baseline classifiers. The LR model is trained with a 'lbfgs' optimizer and ‘l2’ regulizer for 400 iterations. To implement RF, 12k features are considered with a set of 200 decision trees. The quality of node partitioning in RF is measured with the ‘gini’ criterion. SVM is constructed with c=0.6, $\gamma = 0.9$ and ‘rbf’ kernel . Like RF, the ‘l2’ penalizer used in SVM and tolerance value is settled to 0.002. Finally, NB is implemented with an adaptive smoothing parameter value of 1.0.

DL Classifiers: Embedding features are propagated into the DL classifiers to train them. A two-layer CNN architecture is constructed with 128 and 64 filters in layer-1 and layer-2. Each layer has a kernel of size (3×3), and crucial features are captured using the max pooling window of size (1×3). The non-linearity ‘relu’ activation function is utilized, and the softmax layer is used for prediction.

CNN can not hold the long-range information in a text. Therefore to capture dependencies from both past and future bidirectional LSTM and GRU is employed. To layers of BiLSTM and GRU are constructed, each having 64 recurrent units. Although both models have similar architecture, GRU trains the model quickly since it has fewer trainable parameters. To avoid overfitting, an intermediate layer is introduced with a dropout rate of 0.15. Probabilities from DL models are passed to a softmax layer for prediction.

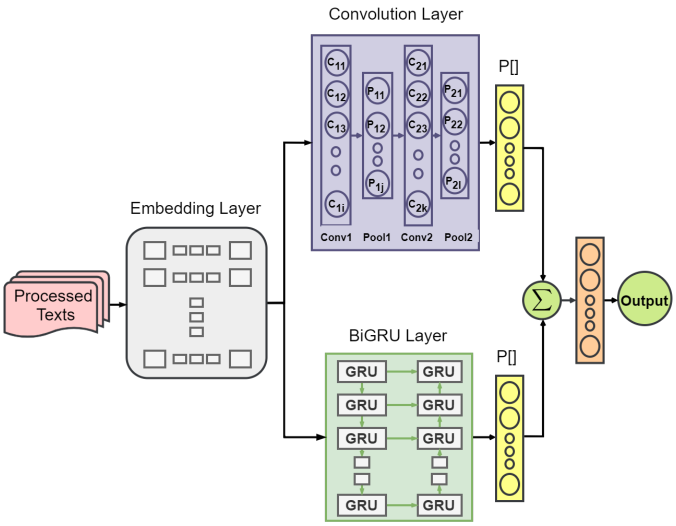

Fig. 1: Proposed ensemble-based (CNN + GRU) framework

Ensemble: Recent work exhibited that ensemble of DL models significantly enhance the system’s predictive accuracy [10]. The ensemble combines different models and exploits the strength of the individual classifiers. Figure 1 presents the proposed textual aggression classification framework. Outcomes of CNN and GRU are aggregated by using equation 1.

$$C = \arg\max\left(\frac{\sum_{i=1}^{N} P_i[]}{N}\right) \tag{1}$$

Here, $P_i[]$ denotes the probability of the classifiers, and N is the total number of base classifiers. These probabilities are summed and normalized by dividing with N. The final predictions are determined by choosing the class (C) with the maximum probability. A trial-and-error approach is used to tune hyperparameters based on the performance in a validation set. A similar architecture is utilized for both coarse-grained and fine-grained classification.

V. Experiments and Results

This section provides comprehensive performance analysis of various ML (LR, NB, SVM, RF), DL (CNN, BiLSTM, GRU), and ensemble-based models on the test set. Weighted f_1-scores are utilized to identify the superior model while precision and recall values are presented for performance comparison. All the experiments are performed on the google colab platform, where Pandas (1.1.4) and Numpy (1.18.5) are used to process and prepare data. ML models are developed by using scikit-learn (0.22.2) packages, while DL models are trained with Keras (2.4.0) and Tensorflow (2.3.0). All the instances in training and evaluation sets are randomly shuffled to eliminate unintentional biases.

Table IV exhibits the performance of various ML and DL models on the test set for coarse-grained classification. Here, the system segregates aggressive texts from non-aggressive ones. Results show that the ensemble of CNN and GRU (i.e. C+G) with FastText embedding outperformed all other models by acquiring the maximal f_1-score of 89.55. Among the ML models, NB achieved the highest f_1-score of 89.11 and 89.32 for BoW and TF-IDF features. GRU with FastText obtained

TABLE IV: Evaluation results for coarse-grained classification

	Classifier	Precision	Recall	F_1-score
BoW	LR	86.69	86.69	86.69
	RF	84.36	83.61	83.55
	SVM	84.50	84.49	84.49
	NB	89.19	89.11	89.11
TF-IDF	LR	88.47	88.01	88.02
	RF	83.19	82.51	82.45
	SVM	87.22	87.02	87.01
	NB	89.44	89.33	89.32
Word2Vec	CNN (C)	86.46	86.46	86.46
	BiLSTM (B)	88.96	88.66	88.63
	GRU (G)	87.36	87.35	87.34
	C+B	87.50	87.45	87.44
	C+G	87.13	87.12	87.13
	B+G	87.99	87.89	87.88
	C+B+G	87.71	87.68	87.68
GloVe	CNN (C)	83.56	83.49	83.50
	BiLSTM (B)	87.27	86.79	86.77
	GRU (G)	86.17	86.13	86.14
	C+B	86.98	86.68	86.69
	C+G	85.86	85.87	85.80
	B+G	86.83	86.57	86.56
	C+B+G	87.12	87.29	87.12
FastText	CNN (C)	88.11	88.12	88.11
	BiLSTM (B)	88.40	88.33	88.34
	GRU (G)	89.44	89.42	89.43
	C+B	88.45	88.44	88.44
	C+G	89.57	89.53	**89.55**
	B+G	89.02	88.97	88.99
	C+B+G	89.42	89.43	89.43

TABLE V: Evaluation results for fine-grained classification

FE	Classifier	Precision	Recall	F_1-score
BoW	LR	80.53	80.67	80.48
	RF	79.05	77.78	76.55
	SVM	81.81	81.11	79.09
	NB	74.10	76.22	73.18
TF-IDF	LR	80.38	81.56	78.95
	RF	80.13	78.67	77.61
	SVM	77.75	74.67	74.78
	NB	78.80	78.01	78.02
Word2Vec	CNN (C)	82.58	82.00	82.19
	BiLSTM (B)	78.11	78.66	78.02
	GRU (G)	78.63	79.11	78.59
	C+B	81.77	82.88	82.08
	C+G	82.98	83.78	83.24
	B+G	79.55	81.33	80.10
	C+B+G	80.40	82.00	80.89
GloVe	CNN (C)	76.56	77.55	76.90
	BiLSTM (B)	80.55	80.44	80.34
	GRU (G)	78.78	78.66	78.63
	C+B	80.93	81.33	81.01
	C+G	78.75	78.78	78.74
	B+G	80.64	80.66	80.57
	C+B+G	79.47	79.78	79.53
FastText	CNN (C)	81.85	82.44	81.85
	BiLSTM (B)	84.31	83.78	83.69
	GRU (G)	83.20	83.11	82.67
	C+B	83.41	83.56	83.11
	C+G	84.14	84.22	**83.77**
	B+G	82.02	82.89	82.15
	C+B+G	83.53	83.78	83.27

the maximum score (89.43) amid the base DL models. It is observed that after employing the ensemble technique, the overall performance is improved by $\approx 1 - 2\%$ for all the models.

Evaluation outcomes for fine-grained classification are demonstrated in table V. Unlike coarse-grained classification, LR achieved the maximum f_1-score with BoW (80.48) and TF-IDF (78.95) in fine-grained classification. BiLSTM with FastText embedding acquired the highest score of 83.69 among the different embedding and base DL models combinations. However, the proposed ensemble model (C+G) surpassed all other models and obtained the maximum f_1-score of 83.77.

A. Error Analysis

Confusion matrices are further analyzed to find out the quantitative errors of the proposed model. Figure 2 presents the values of the confusion matrices for both tasks.

In AG and NoAG classes, a total of 95 samples are misclassified. This misclassification might occur due to the high overlap of frequent words between these two classes. Amid the fine-grained classes, PoAG and VeAG mostly make confusion with ReAG class instances. Since ReAG has a higher number of training samples model might be biased to this class. The misclassification rate is much higher in GeAG compared to other classes. Among 44 test samples, 17 texts are wrongly classified as VeAG. Less number of training samples might be the reason behind this poor performance.

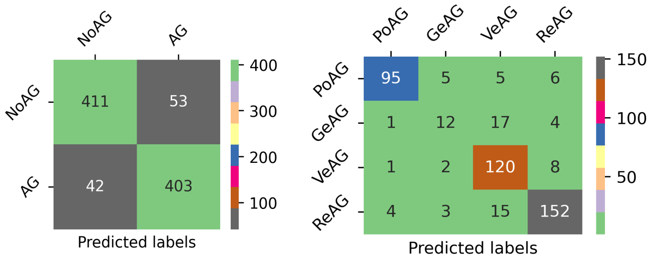

Fig. 2: Confusion matrices of the proposed model (C+G) for coarse-grained and fine-grained classification

B. Comparison with Existing Works

As per our exploration, no significant work has been carried out to classify aggressive Bengali texts. Therefore, several recent methods [8, 11, 12] that have been adopted in similar tasks on other language datasets are employed on ATxtC, and their performance is compared with the proposed technique (i.e. C+G). The comparative analysis exhibited that the proposed ensemble model outdoes past methods by obtaining the maximum weighted f_1-score of 89.55 (coarse) and 83.77 (fine-grained). Table VI shows the results of the comparison.

TABLE VI: Performance comparison between existing and the proposed methods on EATxtC

Methods	Coarse-grained	Fine-grained
Baruah et al. [11]	87.02	76.78
Sharif et al. [8]	87.56	81.72
Kumari et a. [12]	88.33	82.67
Proposed	89.55	83.77

VI. Conclusion

This paper offers an extended aggressive Bengali text corpus of 100095 samples. Comprehensive performance analysis of 8 machine learning, nine deep learning and 12 ensemble model on the developed dataset has been investigated. The experimental evaluations revealed that the ensemble of CNN and GRU model (C+G) achieved superior outcomes. The proposed model obtained the highest weighted f_1-score of 89.55 (coarse-grained) and 83.77 (fine-grained) in classification tasks. In future, it will be interesting to perform experimentation with more fine-grained aggression classes. The mixed aggression, code-switching and code-mixing in Bengali can be explored.

References

[1] 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.

[2] 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. arXiv: 2005.00295 [cs.CL].

[3] 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, no. 1, May 2017. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14955.

[4] 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*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 1415–1420.

[5] 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*, São Paulo: SBC, 2017.

[6] 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: Association for Computational Linguistics, Aug. 2018, pp. 66–73.

[7] 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)*, 2019, pp. 1–5.

[8] 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., Springer Nature Switzerland AG, 2021, pp. 1–12.

[9] B. Vidgen and L. Derczynski, "Directions in abusive language training data, a systematic review: Garbage in, garbage out," *PLOS ONE*, vol. 15, no. 12, pp. 1–32, Dec. 2021.

[10] T. Parvin, O. Sharif, and M. M. Hoque, "Multi-class textual emotion categorization using ensemble of convolutional and recurrent neural network," *SN Comput. Sci.*, vol. 3, p. 62, 2022.

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

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