Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, pages 113–119
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

© European Language Resources Association (ELRA), licensed under CC-BY-NC

AI_ML_NIT_Patna @ TRAC - 2: Deep Learning Approach for Multi-lingual Aggression Identification

Kirti Kumari and Jyoti Prakash Singh
National Institute of Technology Patna
Patna, Bihar, India
kirti.cse15@nitp.ac.in, jps@nitp.ac.in

Abstract

This paper describes the details of developed models and results of team AI_ML_NIT_Patna for the shared task of TRAC - 2. The main objective of the said task is to identify the level of aggression and whether the comment is gendered based or not. The aggression level of each comment can be marked as either Overtly aggressive or Covertly aggressive or Non-aggressive. We have proposed two deep learning systems: Convolutional Neural Network and Long Short Term Memory with two different input text representations, FastText and One-hot embeddings. We have found that the LSTM model with FastText embedding is performing better than other models for Hindi and Bangla datasets but for the English dataset, the CNN model with FastText embedding has performed better. We have also found that the performances of One-hot embedding and pre-trained FastText embedding are comparable. Our system got 11^{th} and 10^{th} positions for English Sub-task A and Sub-task B, respectively, 8^{th} and 7^{th} positions, respectively for Hindi Sub-task A and Sub-task B and 7^{th} and 6^{th} positions for Bangla Sub-task A and Sub-task B, respectively among the total submitted systems.

Keywords: Cyber-aggression, Misogyny, ComMA Project, LSTM, CNN

1. Introduction

The emergence of Internet, social networks and microblogging sites have changed our lifestyles the way we communicate, share, mingle, interact, advertise and do businesses. In India, five most popular social media platforms are Facebook, WhatsApp, YouTube, Twitter and Instagram. YouTube is the most popular social media for video sharing in which we can share educational, entertaining and informational video without paying any cost. All these changes have made our society a virtual place where most of the interactions are taking place through the electronic media. But such changes do not have only positive but also some detrimental effects such as cyber-aggression (Kumari et al., 2019a), cyberbullying (Kumari et al., 2019b; Kumari and Singh, 2020), hate speech (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018), misogynistic aggression, cyberstalking and cyber-crime. A large number of negative incidents are regularly occurring on social media creating a need for continuous monitoring of social media posts to overcome such harmful effects. The identification of cyber-aggression and misogynistic aggression can help to manage such problems. Among the most challenging issues in the identification of cyber-aggression and misogynistic aggression are multi-linguality, multi-modality and different posting styles of social media platforms. In the last few years, the research community has mainly been engaged in addressing these issues by considering these (multi-linguality, multi-modality and different posting styles of social media platforms) challenges and have provided some socio-technical solutions. Among these efforts are the works of the popular shared tasks of TRAC - 1[1] and HASOC - 2019[2] that considered the challenges of identification of cyber-aggression and hate speech on multi-lingual and multiple platforms' comments. In both of the shared tasks, the organizers were mainly focussed on English and Hindi code-mixed comments of Facebook and Twitter and at the same time HASOC - 2019 shared task also considered the English-German code-mixed comments. Similarly, in the current shared task TRAC - 2[3] which comes under Communal and Misogynistic Aggression in Hindi-English-Bangla (ComMA) Project have considered the challenge of multi-linguality for three Indian languages, Hindi-English-Bangla code-mixed comments of YouTube (Ritesh Kumar and Zampieri, 2020). In this shared task, there are two subtasks: (a) Sub-task A- Level of Aggression Identification (overtly aggressive, covertly aggressive or non-aggressive) and (b) Sub-task B- Misogyny Aggression Identification (gendered or non-gendered), for each three code-mixed (English, Hindi and Bangla) languages.

In this contribution, we analyze multi-lingual YouTube comments of three popular Indian languages (Hindi-English-Bangla) provided by TRAC - 2 organizers. We have worked for each dataset and each subtask. For this, we have implemented two popular deep learning models: Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM), and two embedding techniques: One-hot and pre-trained FastText embeddings as input representation for these deep learning models. We have found that single-layer CNN and single-layer LSTM networks have performed better than multi-layered CNN and multi-layered LSTM networks and the LSTM model is performing better than the CNN model for Hindi and Bangla datasets but for English dataset, CNN model is performing better than LSTM model. We have also found that as an input representation, pre-trained FastText embedding is better than other pre-trained embedding methods and the perfor-

[1] First Workshop on Trolling, Aggression and Cyberbullying (TRAC - 1) at COLING - 2018

[2] Hate Speech and Offensive Content Identification in Indo-European Languages at FIRE - 2019

[3] Second Workshop on Trolling, Aggression and Cyberbullying (TRAC - 2) at LREC - 2020

mance of One-hot embedding is similar to the performance of pre-trained FastText embedding.

The rest of the paper is framed as follows. The related works are briefly presented in Section 2. Our proposed framework for Cyber-aggression and Misogyny aggression detection is presented in Section 3. The finding of the proposed systems and analysis of the results are presented in Section 4. Finally, we conclude the paper in Section 5 by pointing out the future direction.

2. Related Work

Identification of aggression in social media is closely related to cyberbullying, hate speech, offensive and abusive language identification. In this section, we have briefly discussed some recently published relevant papers on aggression, cyberbullying, hate speech, offensive, and abusive language identification.

Burnap and Williams (2015) used ensemble techniques to detect racism type hate speech on Twitter and achieved a weighted F1-Score of 0.77 by voted ensemble of Random Forest, Support Vector Machine (SVM) and Logistic Regression classifiers. Malmasi and Zampieri (2017) analyzed methods for detecting hate speech of English tweets by differentiating hate speech from general profanity. They used character *n*-grams, word *n*-grams and word skip-grams features and got an accuracy of 0.78. Davidson et al. (2017) created a dataset for abusive language identification and categorizes the tweets among hate, offensive or neither (neither hate nor offensive). They have reported that the racist and the homophobic tweets generally come under hate class and the sexist tweets generally come under offensive class. They used Logistic Regression, Naive Bayes, Decision Trees, Random Forests and SVM to classify the tweets in three classes and concluded that the comment which does not have any abusive word is very difficult to detect. Malmasi and Zampieri (2018) discussed the challenges appearing in the process of identification of hate speech in social media and distinguished hate speech from profanity. Their claimed accuracy was 0.80 by using ensemble methods. Zampieri et al. (2019a) created a dataset of 14,000 English tweets for three different tasks and named the dataset as Offensive Language Identification Dataset (OLID). They also described the similarities and dissimilarities between OLID and earlier datasets for aggression detection, hate speech detection and similar tasks. These three different tasks were - Task 1: Offensive language detection (the tweet is either offensive or not-offensive), Task 2: Type of offense (Targeted insult or Un-targeted insult) and Task 3: Target of insult or threat (either individual or group or other). Zampieri et al. (2019b) analyzed all the submitted systems of the OffensEval 2019 tasks on the OLID dataset and highlighted the issues in separating the comments having profanity from those threatening comments which do not carry profane language.

Some recent works (Chatzakou et al., 2017; Chen et al., 2018; Raiyani et al., 2018; Modha et al., 2018; Samghabadi et al., 2018; Risch and Krestel, 2018) tried to solve Cyber-aggression issues. The works by Chatzakou et al. (2017) and Chen et al. (2018) are focussed on a particular platform (Twitter) and standard English text for aggression detection, which is not equally applicable to multi-lingual cases and for other social media platforms. Chatzakou et al. (2017) found improved accuracy after combining user and network-based features with text-based features. They got overall precision and recall of 0.72 and 0.73, respectively. Chen et al. (2018) used Convolutional Neural Network (CNN) and sentiment analysis method and reported an accuracy of 0.92. Some researchers of TRAC - 1 shared task (Raiyani et al., 2018; Risch and Krestel, 2018; Modha et al., 2018; Samghabadi et al., 2018) worked on the aforesaid challenges and achieved limited success due to the provided data being very noisy, unbalance and multi-lingual. Some participants (Risch and Krestel, 2018; Modha et al., 2018; Samghabadi et al., 2018) tried ensemble learning methods with various machine learning classifiers and many deep learning models and achieved better performance. The other group of researchers (Risch and Krestel, 2018; Aroyehun and Gelbukh, 2018) applied data augmentation with the help of machine translation using different languages (French, German, Spanish and Hindi) by preserving the meaning of comments with different wording and found better training result for such enlarged dataset. Raiyani et al. (2018) used three layers of dense system architecture with One-hot encoding. They found that simple three-layers of the fully connected neural network model with One-hot encoding performed better than complex deep learning models, but their system suffered from false-positive cases and they omitted the words not found in the vocabulary. Kumari and Singh (2019) proposed a four-layered CNN model with three different embedding techniques: One-hot, GloVe and FastText embeddings to detect different classes of abusive language for multi-lingual text comments of Facebook and Twitter on HASOC - 2019 shared task. They found that FastText and One-hot embeddings performed better than other pre-trained models. The work Kumari and Singh (2019) motivated us to adopt a similar approach for TRAC - 2 shared tasks because here also the tasks are multi-lingual and the data provided are noisy. In this paper, we have addressed the multi-lingual issue of social media post considering YouTube comments in Indian scenario by applying two deep learning models with different types of word embeddings.

3. Methodology and Data

This section presents the descriptions of used datasets and proposed methods. First, we discuss the three different datasets in Section 3.1 and then we explain the details of the proposed approach in Section 3.2.

3.1. Dataset Description

We have used the datasets of the shared task of TRAC - 2[4]. The provided datasets are of English, Hindi and Bangla. The shared task contains two subtasks: Sub-task A (Aggression Identification) and Sub-task B (Misogyny Aggression Identification). Sub-task A is a three-class problem, where the comments are classified into Overtly Aggressive (OAG), Covertly Aggressive (CAG) and Non-Aggressive (NAG) classes. The comment having direct aggression is

[4] https://sites.google.com/view/trac2/home

labelled as OAG comment, the comment having indirect aggression is labelled as CAG comment and the comment that does not have any type of aggression is labelled as NAG comment. Sub-task B is misogyny aggression identification, which is a binary classification task and is labelled as Gendered (GEN) and Non-gendered (NGEN). The comment in which attack is because of someone being a woman, or a man or a transgender is labelled as GEN otherwise the comment is labelled as NGEN. For the training of the proposed models for English and the Hindi Sub-task A, we have also used TRAC - 1 (Kumar et al., 2018) datasets. The organizers of TRAC - 2 shared tasks have provided three sets (Training, Validation and Test sets) of datasets for each language. The class-wise description of all the three datasets of TRAC - 2 is given in Table 1 where S refers the number of samples in each class. The class information has been given only for Training and Validation (or Dev) sets but this information has not been given for Test sets at the time of competition. The more detailed explanation of data collection and labelling is discussed in Bhattacharya et al. (2020). The comments are code-mixed Hindi-English and Bangla-English. These comments are having lots of Emojis. We have not done any pre-processing of data.

Table 1: Class-wise description of all the three datasets of TRAC - 2

Dataset	Set	Sub-task	Class	S
English	Training	A	OAG	435
			CAG	453
			NAG	3375
		B	GEN	309
			NGEN	3954
		Both A and B	Total	4263
	Dev	A	OAG	113
			CAG	117
			NAG	836
		B	GEN	73
			NGEN	993
		Both A and B	Total	1066
	Test	A	OAG	286
			CAG	224
			NAG	690
		B	GEN	175
			NGEN	1025
		Both A and B	Total	1200
Hindi	Training	A	OAG	910
			CAG	829
			NAG	2245
		B	GEN	661
			NGEN	3323
		Both A and B	Total	3984
	Dev	A	OAG	208
			CAG	211
			NAG	578
		B	GEN	152
			NGEN	845
		Both A and B	Total	997
	Test	A	OAG	684
			CAG	191
			NAG	325
		B	GEN	567
			NGEN	633
		Both A and B	Total	1200
Bangla	Training	A	OAG	850
			CAG	898
			NAG	2078
		B	GEN	712
			NGEN	3114
		Both A and B	Total	3826
	Dev	A	OAG	217
			CAG	218
			NAG	522
		B	GEN	191
			NGEN	766
		Both A and B	Total	957
	Test	A	OAG	251
			CAG	225
			NAG	712
		B	GEN	202
			NGEN	986
		Both A and B	Total	1188

3.2. Proposed Method

In this subsection, we describe our best three runs for both the subtasks (Sub-task A and Sub-task B) of each (English, Hindi and Bangla) dataset in detail. First, we have implemented Convolutional Neural Network (CNN) with pre-trained FastText (Joulin et al., 2016) embedding as input representation for the CNN model and have named the system as Run1_CNN_FastText. Then, we have tried One-hot embedding as input representation for the CNN model and have named the system as Run2_CNN_One-hot. But we have found that pre-trained FastText embedding is performing better than One-hot embedding in the validation phase. Therefore, next, we have tried Long Short Term Memory (LSTM) with pre-trained FastText (Joulin et al., 2016) embedding and have named the system as Run3_LSTM_FastText. In the following paragraphs, we discuss the systems in detail.

3.2.1. Input Representation

The deep learning model takes input as the embedding layer, which encodes each token in the dataset used by the model. We have experimented with three popular embedding techniques: pre-trained GloVe, FastText and One-hot embeddings. In One-hot embedding, we assigned each distinct word/token of the dataset with a unique index value (integer value). Then each comment is represented by a one-dimensional vector of the vocabulary size of the dataset. We have used embedding dimension 300 for both pre-trained GloVe and FastText embeddings and the embedding dimension of the size of vocabulary for One-hot embedding. Since all the comments are not of equal length so we have used padding to make them equal. We have padded each comment to the average length of the comments. We have used post padding to make comment length 26, 35 and 30 for English, Hindi and Bangla datasets, respectively. The comment having larger length is truncated up to average length and the comment having smaller than average length is appended zeros to make the length equal to average length. While experimenting, we have found that pre-trained FasText embedding is performing better than pre-trained GloVe embedding and we have also found that the performance of One-hot embedding is comparable to the performance of pre-trained FastText embedding. So, we are reporting the best three runs for the TRAC - 2 shared task obtained by pre-trained FastText and One-hot embeddings.

3.2.2. Deep Learning Models

We have done experiments with two popular deep learning models: Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). In the CNN model, we have implemented one convolutional layer with 128 filters having a filter size of 3 and Rectified Linear Unit (ReLU) as an activation function. Then we have used one max-pooling layer of size 5 followed by flatten layer. After that, we have applied two dense layers of size 256 and 2 or 3 depending upon subtask (the size of last dense layer is 3 for Sub-task A and 2 for Sub-task B) with activation function as ReLU in the first dense layer. We have used dropout of 0.5 in between two dense layers and in between max-pooling and flatten layer.

For the LSTM model, we have implemented one layer of LSTM with 192 LSTM units with both dropout and recurrent_dropout value of 0.2 followed by one dense layer of size 3 or 2 depending upon subtask as is done for CNN model. We have applied the Categorical_crossentropy and Binary_crossentropy as loss function for Sub-task A and Sub-task B, respectively, and Adam is used as optimizer function for both CNN and LSTM models. In both CNN and LSTM models, we have applied Softmax (for Sub-task A) or Sigmoid (for Sub-task B) for the last dense layer depending on the type of problem. We have trained every system for 100 epochs with a batch size of 100.

We have trained our systems with training sets and validated with validation (Dev set) sets of the datasets provided by TRAC - 2 but for Sub-task A of English and Hindi dataset, we have trained our systems with both TRAC - 1 and TRAC - 2 datasets. We have repeated the experiments with varying number of layers of CNN and LSTM networks but did not get any improvement in performance. So, we have decided to use a single-layer CNN and single-layer LSTM networks.

4. Results

This section presents the results and analysis of validation and test sets of all the three datasets provided by TRAC - 2 organizers in terms of weighted F1-Score (as a primary performance metric) and accuracy. In this section, F1-Score refers to weighted F1-Score in all the tables. The validation results for subtasks of each dataset are shown in Table 2. The best three runs for each test set on each dataset are shown in Tables 3 and 4 for Sub-task A and for Sub-task B, respectively, where Acc stands for Accuracy. Each table shows the results obtained by the best three systems for each dataset either Sub-task A or Sub-task B.

Table 2: Validation results of the best three systems for Dev sets

Dataset	Sub-task	System	F1-Score
English	A	CNN_FastText	0.74
		CNN_One-hot	0.76
		LSTM_FastText	0.73
	B	CNN_FastText	0.92
		CNN_One-hot	0.91
		LSTM_FastText	0.92
Hindi	A	CNN_FastText	0.63
		CNN_One-hot	0.63
		LSTM_FastText	0.63
	B	CNN_FastText	0.82
		CNN_One-hot	0.80
		LSTM_FastText	**0.84**
Bangla	**A**	CNN_FastText	0.64
		CNN_One-hot	0.63
		LSTM_FastText	**0.66**
	B	CNN_FastText	0.79
		CNN_One-hot	0.80
		LSTM_FastText	**0.85**

Table 3: Results of the best three systems for test sets of Sub-task A

Dataset	System	F1-Score	Acc
English	**Run1_CNN_FastText**	**0.6602**	**0.6667**
	Run2_CNN_One-hot	0.5997	0.6392
	Run3_LSTM_FastText	0.5952	0.6092
Hindi	Run1_CNN_FastText	0.5964	0.5775
	Run2_CNN_One-hot	0.6370	0.6125
	Run3_LSTM_FasText	**0.6547**	**0.6367**
Bangla	Run1_CNN_FastText	0.7037	0.7088
	Run2_CNN_One-hot	0.7002	0.6987
	Run3_LSTM_FastText	**0.7175**	**0.7306**

From the Table 3 and Table 4, it is observed that the CNN model with FastText embedding is performing better than the other two models for English dataset and the LSTM models with FastText embedding is performing better than the other two models for Hindi and Bangla datasets. Figures 1, 2, 3, 4, 5 and 6 show the confusion matrix of the best results obtained by us for the different datasets for both the subtasks.

We have found that the LSTM model is performing better than the CNN model for Hindi and Bangla datasets whereas the CNN model is performing better than the LSTM model for the English dataset. The reason behind this is that LSTM is preserving long-term dependency of comment when comments are usually longer as in the case of Hindi and Bangla dataset. Our other finding is that FastText embedding is performing better than the other embeddings especially when the data is noisy. This is because FastText embedding is capable of preserve semantics information in solving the issues related to Emoji and out of vocabulary words but One-hot embedding does not consider the semantics information.

Table 4: Results of the best three systems for test sets of Sub-task B

Dataset	System	F1-Score	Acc
English	**Run1_CNN_FastText**	**0.8227**	**0.8383**
	Run2_CNN_One-hot	0.8099	0.8158
	Run3_LSTM_FastText	0.8199	0.8450
Hindi	Run1_CNN_FastText	0.6957	0.6983
	Run2_CNN_One-hot	0.6645	0.6758
	Run3_LSTM_FastText	**0.7363**	**0.7425**
Bangla	Run1_CNN_FastText	0.7834	0.7702
	Run2_CNN_One-hot	0.8211	0.8140
	Run3_LSTM_FastText	**0.8793**	**0.8847**

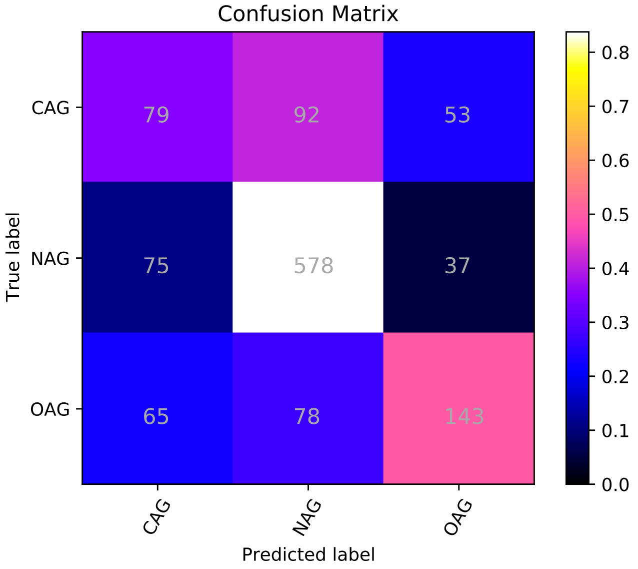

Figure 1: Confusion Matrix of the CNN_FastText model for test set of English Sub-task A

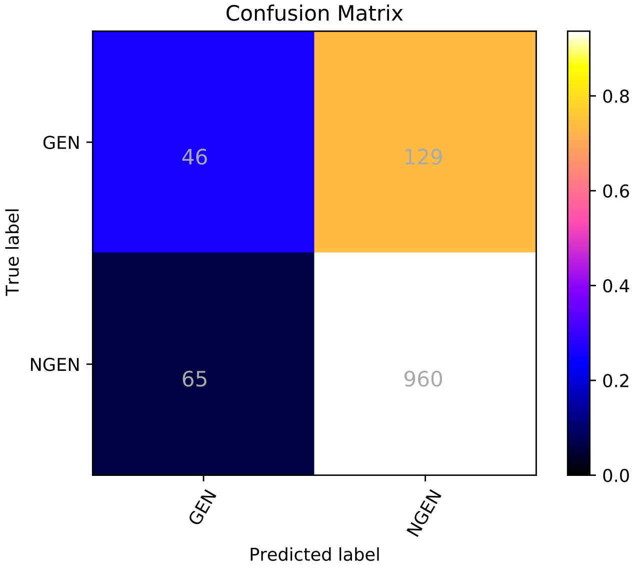

Figure 2: Confusion Matrix of the CNN_FastText model for test set of English Sub-task B

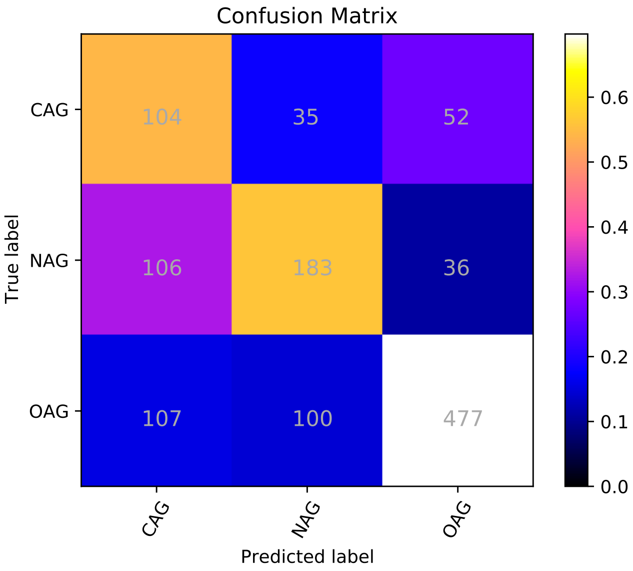

Figure 3: Confusion Matrix of the LSTM_FastText model for test set of Hindi Sub-task A

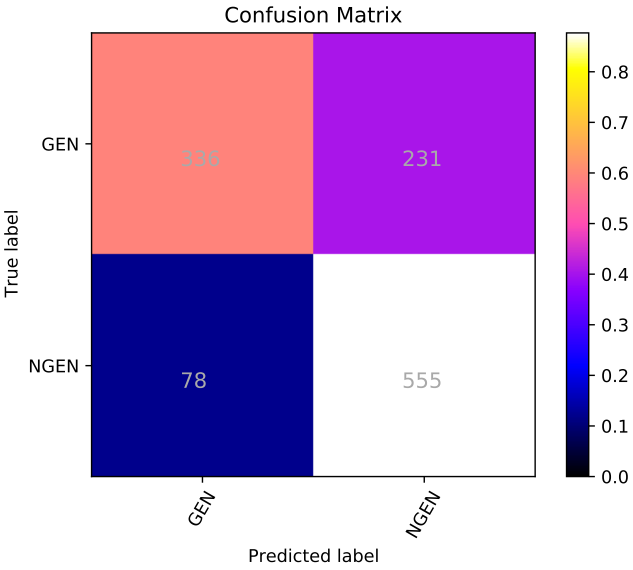

Figure 4: Confusion Matrix of the LSTM_FastText model for test set of Hindi Sub-task B

5. Conclusion

In this paper, we have established the challenges of the TRAC - 2 shared task. Then we have discussed the summary of similar works. Thereafter, we have described the proposed deep learning methods (to combat the issues) which consist of two popular deep learning models: Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM), and two embedding techniques: One-hot and pre-trained FastText embeddings. We have used two different methods for input representation: One-hot and FastText embeddings for deep learning models, and our results show that FastText embedding is performing better than other embeddings in every case. To get better results,

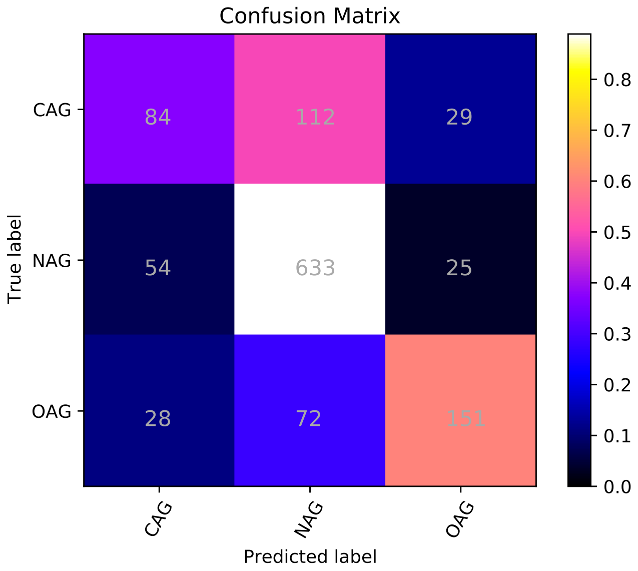

Figure 5: Confusion Matrix of the LSTM_FastText model for test set of Bangla Sub-task A

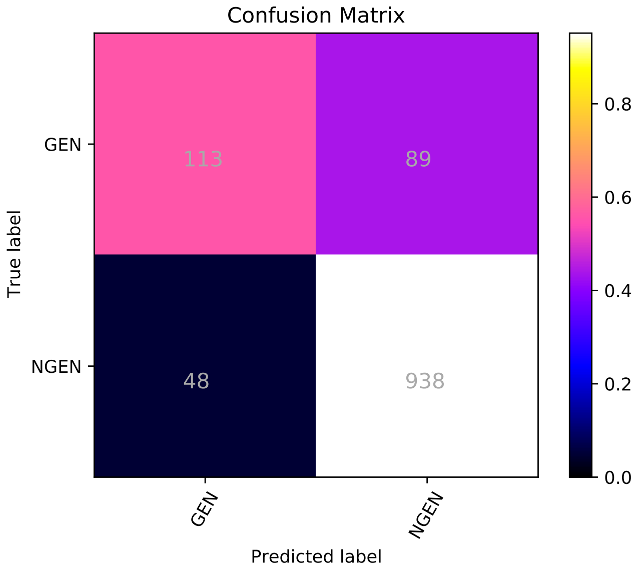

Figure 6: Confusion Matrix of the LSTM_FastText model for test set of Bangla Sub-task B

we have tried several systems and have concluded that a single layer of CNN or LSTM networks perform better for the classification of YouTube comments. We have found that the LSTM model is performing better than the CNN model except for the English dataset. We have achieved weighted F1-Score: 66% and 82% for English Sub-task A and Sub-task B, respectively, 65% and 74%, respectively for Hindi Sub-task A and Sub-task B, and 72% for Bangla Sub-task A and 88% for Sub-task B.

The future system may integrate active learning and unsupervised learning to overcome the burden of labelling efforts.

Acknowledgements

The first author would like to acknowledge the Ministry of Electronics and Information Technology (MeitY), Government of India for the financial support during the research work through Visvesvaraya Ph.D. Scheme for Electronics and IT.

Bibliographical References

Aroyehun, S. T. and Gelbukh, A. (2018). 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)*, pages 90–97.

Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., Lahiri, B., and Ojha, A. K. (2020). Developing a multilingual annotated corpus of misogyny and aggression.

Burnap, P. and Williams, M. L. (2015). Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, 7(2):223–242.

Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., and Vakali, A. (2017). Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on web science conference*, pages 13–22. ACM.

Chen, J., Yan, S., and Wong, K.-C. (2018). Verbal aggression detection on twitter comments: convolutional neural network for short-text sentiment analysis. *Neural Computing and Applications*, pages 1–10.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

Fortuna, P. and Nunes, S. (2018). A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys (CSUR)*, 51(4):85.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651:1–13.

Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. (2018). Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbulling (TRAC)*, Santa Fe, USA.

Kumari, K. and Singh, J. P. (2019). AI_ML_NIT Patna at HASOC 2019: Deep learning approach for identification of abusive content. In *Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation (FIRE 2019, December 2019)*, pages 328–335.

Kumari, K. and Singh, J. P. (2020). Identification of cyberbullying on multi-modal social media posts using genetic algorithm. *Transactions on Emerging Telecommunications Technologies*, doi:10.1002/ett.3907.

Kumari, K., Singh, J. P., Dwivedi, Y. K., and Rana, N. P. (2019a). Aggressive social media post detection system containing symbolic images. In *Conference on e-Business, e-Services and e-Society*, pages 415–424. Springer.

Kumari, K., Singh, J. P., Dwivedi, Y. K., and Rana, N. P. (2019b). Towards cyberbullying-free social media in smart cities: a unified multi-modal approach. *Soft Computing*, doi:10.1007/s00500-019-04550-x.

Malmasi, S. and Zampieri, M. (2017). Detecting Hate Speech in Social Media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 467–472.

Malmasi, S. and Zampieri, M. (2018). Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1–16.

Modha, S., Majumder, P., and Mandl, T. (2018). Filtering aggression from the multilingual social media feed. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 199–207.

Raiyani, K., Gonçalves, T., Quaresma, P., and Nogueira, V. B. (2018). Fully connected neural network with advance preprocessor to identify aggression over facebook and twitter. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 28–41.

Risch, J. and Krestel, R. (2018). Aggression identification using deep learning and data augmentation. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 150–158.

Ritesh Kumar, Atul Kr. Ojha, S. M. and Zampieri, M. (2020). Evaluating aggression identification in social media. In Ritesh Kumar, et al., editors, *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying (TRAC-2020)*, Paris, France, may. European Language Resources Association (ELRA).

Samghabadi, N. S., Mave, D., Kar, S., and Solorio, T. (2018). Ritual-uh at trac 2018 shared task: Aggression identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 12–18.

Schmidt, A. and Wiegand, M. (2017). A Survey on Hate Speech Detection Using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics*, pages 1–10, Valencia, Spain.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019a). Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019b). SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval)*.