NSIT & IIITDWD @ HASOC 2020: Deep learning model for hate-speech identification in Indo-European languages

Roushan Raj[a], Shivangi Srivastava[b] and Sunil Saumya[c]

[a] *Netaji Subhas Institute of Technology, Bihta, Patna, India*
[b] *Netaji Subhas Institute of Technology, Bihta, Patna, India*
[c] *Indian Institute of Information Technology Dharwad, India*

Abstract

In current times, social media is the most widely used platform, and everyone has the right to express their speculations, ideas, thoughts, etc. In such a case, it is often seen that hate speech and offensive contents are spreading like wildfire, making a detrimental impact on the world. It is important to identify and eradicate such offensive content from social media. This paper is a contribution to the *Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC) 2020 shared task*. Our target is to present deep learning models to detect hate speech and offensive content in three languages English, Hindi, and German. Our team **NSIT_ML_Geeks** has developed models using Convolutional Neural Networks (CNN), Bi-directional long short term memory (BiLSTM), and hybrid models (CNN+BiLSTM). The word-embeddings used are GloVe and fastText to convert our corpus into vectors of real numbers to train models. Our best models for Hindi sub-task A and B secured **First** and **Second positions** by outperforming other models submitted in the competition with f1 macro-avg score of 0.5337 and 0.2667 respectively.

Keywords

Hate Speech, Offensive Content, Indo-European Languages, Bi-directional Long Short-Term Memory, GloVe, fastText, CNN

1. Introduction

Social media platforms have made our life remarkably easy by instantly connecting people worldwide [1]. The content of a user can easily reach out to a massive number of people in no time [2]. Its low cost, easy accessibility, and high effectiveness have changed the way we live. But there is a darker side to this technological evolution [3, 4]. Cybercrimes have witnessed a rapid rise worldwide. Online bullying persists to occur in a variety of forms including hate speech, racial and sexual comments, offensive name-calling, trolling, and even death threats. According to a survey, 47.3% of students have been a victim of hate speech in Asian countries[1]. Victims report anxiety, depression, fear, self-harming, and mental health issues as after-effects. Cyberbullying stats 2020 show that 42% of online harassment happens on Instagram which has over a billion active users. Facebook and Snapchat follow closely, with 39% and 31% respectively[2]. Stats have shown that cyberbullying victims are 1.9 times more likely to commit suicide than those not involved in cyberbullying [5].

FIRE '20, Forum for Information Retrieval Evaluation, December 16–20, 2020, Hyderabad, India.

EMAIL: rraaj.raushan@gmail.com (R. Raj); shivangisrivastava762@gmail.com (S. Srivastava); sunil.saumya@iiitdwd.ac.in (S. Saumya)
ORCID:

CEUR Workshop Proceedings (CEUR-WS.org)

[1] https://techjury.net/blog/cyberbullying-statistics/#gref

[2] https://enough.org/stats_cyberbullying

Users across the world express their thoughts in diversified languages. Thus, there has been extensive research to create advanced automated systems using AI technologies to detect and eliminate offensive content from social media platforms in all possible languages. Several models have been proposed by various researchers in the field of hate-speech identification with feature engineering. Risch and Krestel [6] proposed a semi-automatic approach for comment moderation where moderators take notice of a potentially violating comment using a logistic regression model with features like word and character N-grams, linguistic features, etc. Shubhanshu and Sudhanshu [7] proposed fine-tuned pre-trained monolingual and multilingual BERT based approach for HASOC 2019 shared task. Waseem and Hovy [8] proposed models on Hate tweet identification linked with sexism and racism using character n-grams which outperformed word n-grams. Alfina et al. [9] presented ML models like Naïve Bayes, SVM, Bayesian Logistic Regression, and Random Forest Decision Tree to encounter hate speech on the Indonesian language. Kamble et al. [10] used domain-specific word embeddings for hate speech detection in code-mixed Hindi-English tweets. Xu et al. [11] came with the CrossNet model which consists of four layers, embedding, context encoding, attention, and finally prediction layer that learns unseen similar destination target. Research communities are increasingly taking interest to apply machine learning and natural language processing techniques. Many social media platforms monitor user's posts to identify the offensive language. The *Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC)*[3] [12] has been organized as a step towards this direction in three major languages- English, Hindi, and German.

There are 2 sub-tasks for each of the three languages [13]. Sub-task A is a binary classification problem to classify tweets into HOF (Hate and Offensive) or NOT (Non Hate-offensive). Sub-task B is a multi-class fine-grained classification of hate speech and offensive posts obtained from subtask A further into Hate, Offensive, or Profane posts. HATE class includes posts that contain hate content due to political opinion, gender, social status, race, religion, or any other equivalent reasons. OFFN (Offensive) class covers the posts containing offensive content, insulting an individual or a group, or uncomfortable content. PRFN (Profane) class counts the posts containing profane words, unacceptable languages which may be cursing or usage of swear words.

In this paper, we proposed Convolutional Neural Networks (CNN) [14] and Bi-directional long short term memory (BiLSTM) [15] deep neural networks for each language. Our model for Hindi sub-tasks A and B outshined other models, securing 1st and 2nd positions respectively.

In the forthcoming section we describe the dataset, Section 3 presents the methodology in two steps- pre-processing and model architecture, in Section 4 we analyzed the results of all the experimented models while in Section 5 we outlined the conclusion and future work.

2. Dataset Description

The given dataset for each of the three languages is a collection of tweets from Twitter [16] consisting of two sub-tasks (sub-task A and sub-task B). As shown in Table 1, each instance in the dataset consists of a tweet_id which is a unique value for the tweets, the full text of the tweet, the target variable in two separate columns for both sub-task A and sub-task B indicating whether the tweet is HOF (Hate and Offensive) or NOT (Non Hate-Offensive) for subtask A and whether it is HATE, OFFN (Offensive) or PRFN (Profane) for subtask B, and the unique HASOC IDs for each tweet. For most of the sub-tasks, the given dataset is highly imbalanced in all three languages for both training and testing as shown in Table 2 and Table 3 respectively.

[3] https://hasocfire.github.io/hasoc/2020/

Columns	Description
tweet_id	unique value for the tweets
text	full text of the tweets
task1	target value, either tweet is HOF or NOT for sub-task A
task2	target value, either tweet is HATE, OFFN or PRFN for sub-task B
ID	unique hasoc ID for each tweet

Table 1: Dataset Description

Language	Task 1		Task 2			
	HOF	NOT	HATE	OFFN	PRFN	NONE
English	1856	1852	158	321	1377	1852
German	673	1700	146	140	387	1700
Hindi	847	2116	234	465	148	2116

Table 2: Class division of both subtasks for Train Dataset

Language	Task 1		Task 2			
	HOF	NOT	HATE	OFFN	PRFN	NONE
English	423	391	25	82	233	414
German	134	392	24	36	88	378
Hindi	197	466	56	87	27	493

Table 3: Class division of both sub-tasks for Test Dataset

3. Methodology

This section describes the approach followed to bring out a model that segregates the offensive and hateful tweets from the other non-offensive tweets. The subsequent content describes the approach used for the further classification of hate speech into three different categories of hate, profane, and offensive. We begin by expounding the preprocessing steps of the dataset for each of the three languages, followed by the model architecture for each of them. We have also made our approach public[4].

3.1. Pre-processing

The preprocessing of text data for the three languages has been done in the following ways. For the Hindi language, we first converted the texts to lowercase, and removed the redundant texts such as URLs and punctuation symbols e.g. !"#$%&(́)*+,-./:;<=>?@[/]{̂|}. We removed the retweet symbol (RT) of Twitter data. Next, we removed all the Hindi stopwords using the 'Swadesh'[5] list. Further, we tokenized each word, created vocabulary for tokens followed by encoding. Lastly, we performed padding keeping a fixed length of size 100. The same steps have been applied to preprocess English and German languages with slight differences. In the English dataset, we filtered the data using regex library (re), eliminated single alphanumeric characters, and removed apostrophes by expanding the word to maintain proper structure, and to avoid any chances of word sense disambiguation. Then we removed English stopwords and performed stemming using 'PorterStemmer'. For the German dataset, stopwords were removed and stemming was performed using 'SnowballStemmer'. The further preprocessing steps of tokenization, encoding, and padding in both English and German datasets were identical to the steps mentioned above for the Hindi dataset.

[4] https://github.com/roushan-raj/HASOC-2020

[5] https://docs.cltk.org/en/latest/hindi.html

3.2. Model Architecture

The proposed model consists of two different deep neural network approaches tested for all three languages. These are the Convolutional Neural network (CNN) and Bi-directional LSTM (BiLSTM). The forthcoming section describes our best performing models. Let us understand each model one by one.

In the English language model, we used GloVe[6] embeddings [17] in both the sub-tasks. Embedding is a technique used to encode corpus into pre-trained weights. This embedding layer is fed into the input layer of deep neural networks. In the CNN model, we used two convolutional, two dropout, and two max-pooling layers accompanied by a flatten layer and a dense layer. For the German model, we used 'fastText'[7] embedding [18] for both the sub-tasks. The output of this embedding layer is fed to one convolutional layer followed by a dropout and a max-pooling layer after which flatten and dense layers are used. Also, in Hindi sub-tasks, we used 'fastText' embedding. In sub-task A, one layer of bi-directional LSTM and a dropout layer followed by a dense layer performed best. For sub-task B, one convolutional layer with dropout and max-pooling is used followed by a flatten and dense layer. For each sub-task in each language, we used an embedding dimension of 300, and applied 'Adam' optimizer to reduce the losses and to achieve the most accurate results possible. Also, we applied the 'ADASYN' [19] over-sampling technique to balance the data for sub-task B as the dataset was heavily unbalanced as shown in Table 2. 'ReLU' activation is used in the internal layers and 'sigmoid' activation at the final output dense layer. We have used Keras library[8] to build all our models.

4. Results

In this section, we describe the results obtained in each sub-task for the experimented models of all the three languages. We further analyze and compare the observations. The results for all the sub-tasks are evaluated using the f1 macro-avg score. We experimented with one-layer and two-layer CNN, one layer and two-layer BiLSTM, and Hybrid (combination of CNN and BiLSTM) models.

Table 4 shows the f1 macro-avg score of our best six models calculated by the organization with approximately 15% of the private test data. Among all the models submitted, we secured First and Second position for the Hindi sub-tasks A and B delivering the leading f1 macro-avg score of 0.5337 & 0.2667 respectively.

Languages	Sub-task	f1 macro-avg
English	Sub-task A and B	0.4879 and 0.2361
German	Sub-task A and B	0.4919 and 0.2468
Hindi	**Sub-task A and B**	**0.5337 and 0.2667**

Table 4: Results of best models on test data

[6] https://nlp.stanford.edu/projects/glove/

[7] https://fasttext.cc/docs/en/crawl-vectors.html

[8] https://keras.io/

Language	Sub-task	Model	Embedding	f1 macro-avg
English	A	CNN 1 layer	GloVe	0.84
		CNN 2 layer	**GloVe**	**0.86**
		BiLSTM 1 layer	GloVe	0.84
		BiLSTM 2 layer	GloVe	0.83
		Hybrid Model	GloVe	0.84
	B	CNN 1 Layer	GloVe, Unbalanced dataset	0.49
			GloVe, SMOTE	0.49
			GloVe, ADASYN	0.53
		CNN 2 Layer	GloVe, Unbalanced dataset	0.49
			GloVe, SMOTE	0.51
			GloVe, ADASYN	**0.54**
		BiLSTM 1 Layer	GloVe, Unbalanced dataset	0.48
			GloVe, SMOTE	0.50
			GloVe, ADASYN	0.51
		BiLSTM 2 Layer	GloVe, Unbalanced dataset	0.48
			GloVe, SMOTE	0.49
			GloVe, ADASYN	0.51
		Hybrid Model	GloVe, Adasyn	0.51
German	A	**CNN 1 layer**	**fastText**	**0.75**
		CNN 2 layer	fastText	0.73
		BiLSTM 1 layer	fastText	0.74
		BiLSTM 2 layer	fastText	0.70
		Hybrid Model	fastText	0.72
	B	**CNN 1 Layer**	fastText, Unbalanced dataset	0.39
			fastText, SMOTE	0.43
			fastText, ADASYN	**0.45**
		CNN 2 Layer	fastText, Unbalanced dataset	0.39
			fastText, SMOTE	0.40
			fastText, ADASYN	0.43
		BiLSTM 1 Layer	fastText, Unbalanced dataset	0.38
			fastText, SMOTE	0.41
			fastText, ADASYN	0.42
		BiLSTM 2 Layer	fastText, Unbalanced dataset	0.37
			fastText, SMOTE	0.33
			fastText, ADASYN	0.35
		Hybrid Model	fastText, ADASYN	0.41
Hindi	A	CNN 1 layer	fastText	0.55
		CNN 2 layer	fastText	0.57
		BiLSTM 1 layer	**fastText**	**0.67**
		BiLSTM 2 layer	fastText	0.59
		Hybrid Model	fastText	0.53
	B	**CNN 1 Layer**	fastText, Unbalanced dataset	0.23
			fastText, SMOTE	0.35
			fastText, ADASYN	**0.36**
		CNN 2 Layer	fastText, Unbalanced dataset	0.22
			fastText, SMOTE	0.37
			fastText, ADASYN	0.34
		BiLSTM 1 Layer	fastText, Unbalanced dataset	0.29
			fastText, SMOTE	0.33
			fastText, ADASYN	0.35
		BiLSTM 2 Layer	fastText, Unbalanced dataset	0.28
			fastText, SMOTE	0.32
			fastText, ADASYN	0.32
		Hybrid Model	fastText, ADASYN	0.34

Table 5: Results obtained for all experimented models on development data

Table 5 shows the f1 macro-avg score evaluated on development dataset, which is the testing data released by the HASOC organizers. We analyzed the models on the original unbalanced dataset as well as using 'SMOTE' and 'ADASYN' to overcome the effects of the imbalanced class distribution. For English sub-task A, two-layer CNN model with GloVe embedding performed the best and for sub-task B, the same model with ADASYN gave the best result bringing out the f1 macro-avg score of 0.86 and 0.54 respectively. In case of German sub-task A and B, one layer CNN model with fastText embedding and the same model with ADASYN gave the finest f1 macro-avg score among all the other models of 0.75 and 0.45 respectively. In the Hindi language, one layer BiLSTM model with fastText embedding outperformed for sub-task A with f1 macro-avg of 0.67, and for sub-task B, one layer CNN model with fastText embedding and ADASYN, outperformed with 0.36 f1 macro-avg score. We could see that sub-task B achieved a lower f1 macro-avg score than sub-task A irrespective of the language. This could be mainly due to the heavily unbalanced dataset as well as very miniature differences in the three classes due to which the model predicted a lot more false-positive classes. The Hindi dataset was mixed with a lot of English words, while the embedding used was only for the Hindi language, which could probably be a reason for the poor performance in sub-task B. From Table 5, we infer that we got better results after applying oversampling techniques such as ADASYN or SMOTE in almost all the categories of sub-task B.

5. Conclusion and Future works

This paper puts forward a deep neural network model to identify hate speech, offensive content, and profane tweets. We proposed different CNN and BilSTM architecture developed using word vectors of the relevant pre-trained corpus. Many types of researches have been carried out for the English language but languages such as Hindi, German, etc, and multilingual data are now also being focused upon. We saw that the dataset for sub-task B was majorly unbalanced and gave a lower f1 macro-avg score even after applying SMOTE and ADASYN over-sampling techniques. Future work could be improving dataset balancing. Further improvisation could be to tackle the identification of hate speech in multilingual tweets and posts on social media and presumably by adding other features that may not have been included in the specified models.

References

[1] A. Kumar, J. P. Singh, S. Saumya, A comparative analysis of machine learning techniques for disaster-related tweet classification, in: 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129), IEEE, 2019, pp. 222–227.

[2] S. Saumya, J. P. Singh, P. Kumar, Predicting stock movements using social network, in: Conference on e-Business, e-Services and e-Society, Springer, 2016, pp. 567–572.

[3] S. Saumya, J. P. Singh, et al., Spam review detection using lstm autoencoder: an unsupervised approach, Electronic Commerce Research (2020) 1–21, https://doi.org/10.1007/s10660–020–09413–4.

[4] S. Saumya, J. P. Singh, Detection of spam reviews: A sentiment analysis approach, Csi Transactions on ICT 6 (2018) 137–148.

[5] S. Hinduja, J. W. Patchin, Connecting adolescent suicide to the severity of bullying and cyberbullying, Journal of school violence 18 (2019) 333–346.

[6] J. Risch, R. Krestel, Delete or not delete? semi-automatic comment moderation for the newsroom,

in: Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018), 2018, pp. 166–176.

[7] S. Mishra, S. Mishra, 3idiots at hasoc 2019: Fine-tuning transformer neural networks for hate speech identification in indo-european languages., in: FIRE (Working Notes), 2019, pp. 208–213.

[8] 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, 2016, pp. 138–142.

[9] I. Alfina, R. Mulia, M. I. Fanany, Y. Ekanata, Hate speech detection in the indonesian language: A dataset and preliminary study, in: 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS), IEEE, 2017, pp. 233–238.

[10] S. Kamble, A. Joshi, Hate speech detection from code-mixed hindi-english tweets using deep learning models, arXiv preprint arXiv:1811.05145 (2018).

[11] C. Xu, C. Paris, S. Nepal, R. Sparks, Cross-target stance classification with self-attention networks, arXiv preprint arXiv:1805.06593 (2018).

[12] T. Mandl, S. Modha, G. K. Shahi, A. K. Jaiswal, D. Nandini, D. Patel, P. Majumder, J. Schäfer, Overview of the HASOC track at FIRE 2020: Hate Speech and Offensive Content Identification in Indo-European Languages), in: Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, CEUR, 2020.

[13] Hate speech and offensive content identification in indo-european languages competition overview and details, 2020. URL: https://competitions.codalab.org/competitions/26027.

[14] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882 (2014).

[15] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, B. Xu, Text classification improved by integrating bidirectional lstm with two-dimensional max pooling, arXiv preprint arXiv:1611.06639 (2016).

[16] Hate speech and offensive content dataset, 2020. URL: https://competitions.codalab.org/competitions/26027#participate-get-data.

[17] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[18] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146.

[19] H. He, Y. Bai, E. A. Garcia, S. Li, Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), IEEE, 2008, pp. 1322–1328.