IIIT_DWD@HASOC 2020: Identifying offensive content in Indo-European languages

Ankit Kumar Mishra[a], Sunil Saumya[b] and Abhinav Kumar[a]

[a]Magadh University, Bodh Gaya, India
[b]Indian Institute of Information Technology Dharwad, Karnataka, India
[a]National Institute of Technology Patna, Patna, India

Abstract

Human behaviour remains the same whether it is a physical or cyber world. They express their emotions like happy, sad, angry, frustrated, bullying, and so on at both places. To express these emotions in cyberspace one of the way is a text post. The impact of these posts lasts forever on social media sites like Twitter, Facebook, and so on. Some posts that contain hate and offensive content affect victims badly and drag them into mental illness. The current paper aims to identify such hate and offensive posts using deep learning-based models such as CNN, and LSTMs. The Twitter posts in English, Hindi, and German languages used in this study are a part of *HASOC-2020* competition. The model submitted for English sub-task A outperformed all other models submitted in the competitions by securing the 1st rank and *F1-macro average* score of 0.5152.

Keywords

Hate speech, offensive, deep learning, machine learning

1. Introduction

Social media platforms such as Facebook, WhatsApp, Twitter, Instagram, to name a few, have become essential components of today's life [1, 2, 3]. These platforms initially aimed to connect people across the world by expressing and sharing their feelings like happy, sad, angry, and so on [4]. Currently, social platforms serve many more purposes such as enable governments to engage people, allow consumers to make informed decisions, etc. [5]. The higher accessibility of social media content requires to handle them effectively with care for creating knowledge and formulating strategies because all available contents on these sites are not clean or friendly [6, 7]. Such unfriendly contents are often termed as hate or offensive and may not provide a decent environment for individuals or groups. Although, uses of offensive language (directly, or in some other public conversations) impacts negatively on almost all age groups but, teenagers are most vulnerable. The increase in the use of offensive language could lead to mental illness such as sleeping disorder, frustration, depression, and a large change in user behaviour [8]. Therefore, it is essential to stop the propagation of bad or offensive languages in text conversations.

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

ankitmishra.co.in@gmail.com (A.K. Mishra); sunil.saumya@iiitdwd.ac.in (S. Saumya); abhinavanand05@gmail.com (A. Kumar)

The identification of offensive content or hate speech in social media texts is well investigated in the last couple of years. Many natural language processing methods or tools are applied to the said problem. The most popular way to identify offensive content is classifying every content in one of two classes offensive or not offensive, a binary classification approach. Recently, [9, 10, 11] used a bidirectional encoder representations from transformers (BERT) language model to effectively identify the offensive content written in English texts. [12] proposed a convolutional neural network (CNN)-based model for the same task in code mixed Hindi datasets. A summary of shared task *"Aggression and Cyberbullying (TRAC - 2) at LREC 2020"* for identifying offensive contents in three different languages Bengali, Hindi, and English is explained in the paper [13]. The shared task involved two sub-tasks: identification of aggression (sub-task A) in which systems classified each YouTube posts as aggressive, covertly aggressive, and non-aggressive, and identification of gendered aggression (sub-task B) in which each post was categorized into gendered or non-gendered. They reported that best performing models obtained similar accuracy for all three languages for sub-task A, but for sub-task B accuracy was varying for different languages. Similarly, [14] reported that the performance of models for identifying offensive contents varies across languages. Further, they said that deep learning models such as CNNs, RNNs (recurrent neural networks), or transfer models such as BERT behave similarly but better than conventional classifiers.

In line with the above works, the current paper examines the robustness of several deep learning models for identifying offensive content on Indo- European languages. In particular, the paper utilizes three different corpora English, Hindi, and German provided in the *HASOC 2020 track* for the identification of hate and offensive contents. Several conventional and deep learning models were trained and evaluated on said corpora. The experimental results on English *sub-task A* reported that the LSTM-based model outperformed all other models submitted in the challenge *HASOC 2020* with a 0.5152 F1 macro average score and secured the first rank. For other datasets, the performance of our best models was decent.

The organization of the paper is as follows: the tasks proposed in *HASOC 2020* and their data description is explained in Section 2. This is followed by the model description. The results of various experiments are explained in Section 3. The current paper summarizes the most important findings in Section 4.

2. Methodology

The current paper proposes multi-task classification methods for the identification of offensive and hate contents. A detailed flow of the proposed methods is shown in Figure 1.

2.1. Task description

The Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC) 2020 task is one of the ten tracks proposed in the peer-reviewed conference Forum for Information Retrieval Evaluation (FIRE) 2020 [15]. HASOC 2020 hosted tasks in three different languages English, Hindi, and German. For every language, two tasks were proposed. Altogether 6 different tasks were proposed. The first task "sub-task A" in every language is a coarse-grained level classification task that required the system to classify each twitter post

Table 1
Train and Dev datasets description

Language	Task 1 (sub-task A)					Task 2 (sub-task B)						
	HOF		NOT		Total	HATE		OFFN		PRFN		Total
	Train	Dev	Train	Dev		Train	Dev	Train	Dev	Train	Dev	
English	1856	423	1852	391	**4522**	158	25	321	82	1377	293	**2156**
Hindi	847	197	2116	466	**3626**	234	56	465	87	148	27	**1017**
German	673	134	1700	392	**2899**	146	24	140	36	387	88	**821**

into one of the two classes “Hate and Offensive (HOF)", and “Not Hate and Offensive (NOT)". Whereas, the second task, “sub-task B" in every language is a fine-grained level classification task that required the proposed system to classify each “Hate and offensive" post from sub-task A further into three categories “Hate speech (HATE), Offensive (OFFN), and Profane (PRFN)".

2.2. Data description

The train datasets were released initially in three different files, each for English, Hindi, and German. Every dataset file had the following fields: *tweet_id, text, task1, task2, and ID*. The “tweet_id" field referred to the unique id for each piece of text, the “text" field contained the tweet posts, “task1" had the general label of tweet post for sub-task A, “task2" had the general label of tweet post for the sub-task B, and “ID" indicated the unique id generated by the system for each data point. Similar to train, the development (dev) dataset also contained three files, each for a given language. The dataset dimension in both train and dev for both subtasks is shown in Table 1. As can be seen in the last column of Table 1, the total data on which the proposed model was trained and developed were 4522, 3626, 2899 for English, Hindi, and German respectively for task 1, and 2156, 1017, 821 for English, Hindi, and German respectively for task 2.

2.3. Data preprocessing

The preprocessing steps were employed in every language datasets (train and dev) “text" field. In all cases, the steps followed were almost similar. Initially, the texts were converted into lowercase, then all punctuations were removed from the texts. The cleaned texts were then tokenized and encoded into the sequence of token indexes. Finally, to make all texts of equal length, padding was performed with the maximum length of 20.

2.4. Proposed Model

The detailed model diagram is shown in Figure 1. As can be seen in Figure 1, the maximum number of words passed to the model was 20. Every word was then represented in a one-hot vector form having the dimension of vocabulary (1 × 11484). The one-hot input representation was a high dimensional sparse vector, having a single ‘1’ (that represents the index of the token), and all zeros. The embedding layer takes this high dimensional sparse input vector as an input and outputs a low dimensional dense embedded vector (1×300). Each embedded input vector was then fed to a one layered LSTM network in a time-stamp manner. The last LSTM

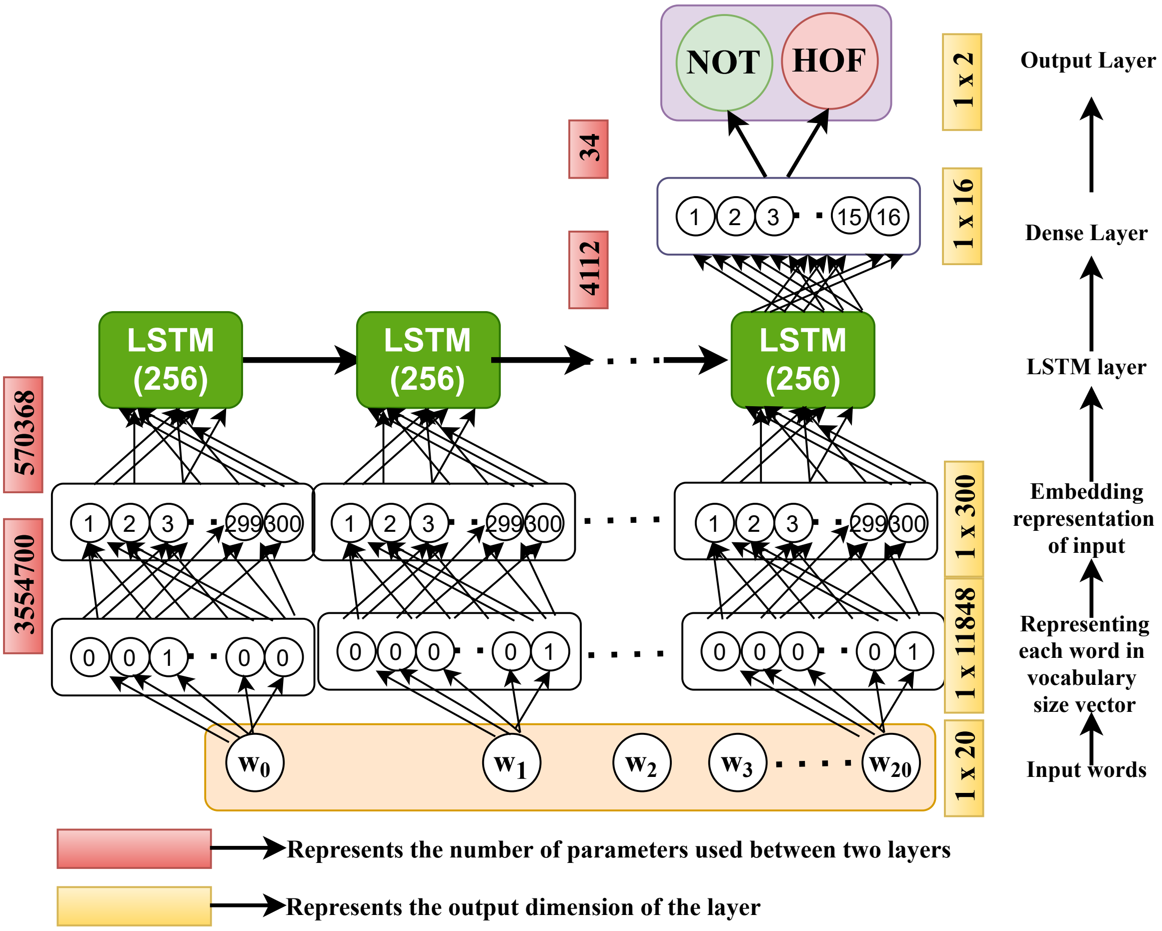
NOT
HOF
1 x 2
Output Layer
34
1
2
3
15
16
1 x 16
Dense Layer
4112
LSTM (256)
LSTM (256)
LSTM (256)
LSTM layer
570368
1
2
3
299
300
1 x 300
Embedding representation of input
3554700
0
0
1
0
0
0
1
1 x 11848
Representing each word in vocabulary size vector
w_0
w_1
w_2
w_3
w_{20}
1 x 20
Input words
Represents the number of parameters used between two layers
Represents the output dimension of the layer

Figure 1: The proposed model flow diagram

unit represents the sentence representation of the tweet. The output of the LSTM network (1 × 256) was passed to the fully connected layer that outputs in (1 × 16) dimensional vector. The output layer predicted one of the two classes NOT or HOF for every input tweet. The proposed model was trained with “*Adam*" optimizer and “*binary crossentropy*" loss function.

3. Results

Extensive experiments were performed utilizing several conventional machine learning models such as support vector machine (SVM), random forest (RF), logistic regression (LR), gradient boosting (GB), and a few deep learning models such as a vanilla deep neural network (DNN), convolutions neural networks (CNN), long short term memory (LSTM) networks, and a hybrid CNN+LSTM network. These models were trained against all six tasks. After training, each model listed above was validated with the development dataset. Here, we are reporting the development (*dev*) results of only the best models against each task in Table 2. As can be

Table 2
Results on Development and Test Dataset

Language	Tasks	Model	Embedding/ Feature	Dev Macro F1	Test Macro F1	Rank Obtained	1st Ranked team/Test Macro F1
English	Sub-task A	LSTM	GloVe	0.86	**0.5152**	**1**	**IIIT_DWD/0.5152**
	Sub-task B	LSTM	GloVe	0.55	0.2341	17	chrestotes/0.2652
Hindi	Sub-task A	CNN	Random	0.67	0.5121	11	NSIT_ML_Geeks/0.5337
	Sub-task B	CNN	Random	0.42	0.2374	9	Sushma Kumari/0.3345
German	Sub-task A	RF	TF-IDF	0.78	0.5019	12	ComMA/0.5235
	Sub-task B	CNN	Random	0.48	0.2513	13	ComMA/0.2831

seen in Table 2, for English sub-task A and sub-task B, the best model was LSTM with GloVe embedding and it obtained *F1 macro-average* scores of 0.86 and 0.55, respectively. For Hindi sub-task A and sub-task B, the best-reported model was CNN with random embedding and it achieved 0.67 and 0.42 *F1 macro-average*, respectively. A similar case was for German sub-task B where CNN with random embedding achieved *F1 macro-average* 0.48. Unlike other sub-tasks, for German sub-task A, a conventional classifier RF performed best with 0.78 *F1 macro-average*.

The best models obtained from *dev* data evaluation were then submitted in the competition for final evaluation by *HASOC-2020* organizers. They evaluated each model submitted by all participating teams against each sub-task by themselves on test data. As the test data was private, its dimension is unknown to us. Based on the test *F1 macro-average* score, the final ranking for all submitted models was published. Table 2 shows the test macro F1 score obtained by the corresponding model against each task. As can be seen in Table 2, for English sub-task A, the model submitted by the current paper (team IIIT_DWD) identified as best with *F1 macro-average* 0.5152. In the last column of Table 2, the 1st ranker team names and their *F1 macro-average* score against each sub-task is tabulated.

4. Conclusion

The current paper identified hate or offensive contents in Twitter posts using several machine learning and deep learning-based models. The said task was proposed in the *HASOC-2020* track of *FIRE-2020*. In particular, there were two sub-tasks proposed for three different languages English, Hindi, and German. In sub-task A, a Twitter post, written in English, Hindi, or German, was categorized as *hate and offensive (HOF)* and *not hate and offensive (NOT)*. In sub-task B, each hate and offensive content was further classified as hate, offensive, and profane. The model submitted for English sub-task A, obtained 0.5152 *F1 macro average* and secured 1st rank among all models submitted in the *HASOC-2020* competition. The results obtained for other sub-tasks were decent.

References

[1] A. Kumar, N. C. Rathore, Relationship strength based access control in online social networks, in: Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2, Springer, 2016, pp. 197–206.

[2] A. Kumar, J. P. Singh, Location reference identification from tweets during emergencies: A deep learning approach, International journal of disaster risk reduction 33 (2019) 365–375.

[3] 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,.

[4] 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.

[5] S. Saumya, J. P. Singh, Y. K. Dwivedi, Predicting the helpfulness score of online reviews using convolutional neural network, Soft Computing (2019,https://doi.org/10.1007/s00500-019-03851-5) 1–17.

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

[7] 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.

[8] S. Kawate, K. Patil, Analysis of foul language usage in social media text conversation, International Journal of Social Media and Interactive Learning Environments 5 (2017) 227–251.

[9] H. Ahn, J. Sun, C. Y. Park, J. Seo, Nlpdove at semeval-2020 task 12: Improving offensive language detection with cross-lingual transfer, arXiv preprint arXiv:2008.01354 (2020).

[10] W. Dai, T. Yu, Z. Liu, P. Fung, Kungfupanda at semeval-2020 task 12: Bert-based multi-task learning for offensive language detection, arXiv preprint arXiv:2004.13432 (2020).

[11] M. Ibrahim, M. Torki, N. El-Makky, Alexu-backtranslation-tl at semeval-2020 task [12]: Improving offensive language detection using data augmentation and transfer learning, in: Proceedings of the International Workshop on Semantic Evaluation (SemEval), 2020.

[12] K. Kumari, J. P. Singh, Ai_ml_nit_patna@ trac-2: Deep learning approach for multilingual aggression identification, in: Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, 2020, pp. 113–119.

[13] R. Kumar, A. K. Ojha, S. Malmasi, M. Zampieri, Evaluating aggression identification in social media, in: Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, 2020, pp. 1–5.

[14] S. Modha, T. Mandl, P. Majumder, D. Patel, Tracking hate in social media: Evaluation, challenges and approaches, SN Computer Science 1 (2020) 1–16.

[15] 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.