An Ensemble Model for Hate Speech and Offensive Content Identification in Indo-European Languages

Anusha M D, H L Shashirekha

Department of Computer Science, Mangalore University, Mangalore, Karnataka, India

Abstract

Hate speech and offensive content is an attack that is coordinated towards a gathering of individuals or society based on their religion, gender, color, and so on and poses a threat to society. This type of content is increasing day by day with the increasing use of social media such as Facebook, WhatsApp, Instagram, etc. Identifying such texts at the earliest to avoid them getting viral on social media creating a negative impact on society is the need of the day. Analyzing these voluminous and every growing text manually is challenging, time-consuming, and error-prone. Further, much of the existing works to identify hate speech and offensive content focuses on high resource languages such as English, Spanish, etc. In this paper, we, team MUM, describe the work submitted to Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC) 2020, a shared task in Forum for Information Retrieval Evaluation (FIRE) 2020. In the proposed methodology, we combine CountVectorizer and TF-IDF transformer with additional text-based features to build an ensemble of Gradient Boosting, Random Forest and XGBoost classifiers, with soft voting. The proposed approaches obtained 5th, 14th, 7th, 3rd, 13th and 6th rank for English, German and Hindi Subtasks A and B respectively.

Keywords

Machine learning, Text Classification, Ensemble learning

1. Introduction

Social media is a global place wide open for online users to share their thoughts and opinions. Various benefits of social media come with several challenges including hate speech, offensive and profane content getting published targeting an individual, a group or a society. Hate speech and other offensive and objectionable content in online socialization have seriously affected people's daily life leading to depression or suicide in the worst case. Many nations have restricted the online media content subject to the condition that the content should not target an individual or a group or trigger any crime. Further, social media companies such as, YouTube, Facebook, and Twitter have their own approaches to eliminate the hate speech content or anything which negatively affects the society. However, detecting such objectionable content at the earliest to curb the menace of spreading such news online is still a major challenge faced by social media companies [1] and researchers.

Characterizing and understanding hate speech is hard. Researchers have explored several algorithms starting from the early lexicon based approaches to the latest Neural Network

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

EMAIL: anugowda251@gmail.com (Anusha M D); hlsrekha@gmail.com (H L. Shashirekha)
URL: https://mangaloreuniversity.ac.in/dr-h-l-shashirekha (H.L. Shashirekha)

approaches to detect hate speech and objectionable content. Nevertheless, some algorithms work better on some datasets, but the same algorithms may not give even an average performance on some other datasets. So, it is hard to generalize that a specific approach is good for all datasets. Further, many of the hate speech or objectionable content detection research work focus on the high resource languages such as English and very little attention is directed to other languages in general and Indian languages in particular. In this paper, we, team MUM, propose an ensemble of Machine Learning (ML) algorithms for hate speech and offensive content identification in Indo-European languages namely, English, Hindi, and German in a shared task called HASOC 2020[1] of Forum for Information Retrieval Evaluation (FIRE) 2020[2].

HASOC 2020 involves two subtasks for each language: i) Subtask A is a typical binary classification problem which identifies whether a given text contains “HOF” i.e., hate, offensive, and profane content or “NOT” i.e., no hate, offensive, and profane content and ii) Subtask B is a multi-class classification problem of identifying whether the “HOF” labeled text in Subtask A contains hate speech, offensive or profane content and labeling it as HATE, PRFN, or OFFN respectively. More details about the tasks are given in competition page and reference paper [2]. The rest of the paper is sorted out as follows. Section 2 highlights the related work and the proposed methodology is described in Section 3. Experiments and results are portrayed in Section 4 and the paper finally concludes in Section 5.

2. Related Work

Hate Speech detection is of incredible significance and is attracting numerous researchers. Several recent papers give a good introduction to the issues associated with hate speech identification [3]. Some of the important research works related to detecting hate speech and offensive content is given below:
Davidson et. al. [4] trained a set of multi-class classifiers namely Logistic Regression, Naive Bayes, Decision Trees, Random Forests and Linear SVM to characterize tweets into one of three classes, namely, hate speech, offensive but not hate speech, neither offensive and nor hate speech. Their model obtained a precision of 0.91, recall of 0.90 and an F1 score of 0.90 for the best performing model. A hate speech and offensive content detection model submitted to HASOC 2019 by Urmi Saha et. al. [5] uses a list of hate words for feature engineering to build ML approaches for English and their approach on the test set provided by HASOC 2019 achieved accuracies of 0.68%, 0.65% and 0.66% for English language subtasks 1, 2 and 3 respectively. An automatic tweet categorization system proposed by Gaydhani et. al. [6] categorizes tweets into three classes namely, hateful, offensive and cleaned. They performed a comparative analysis of the ML classifiers namely Naïve Bayes, Logistic Regression and SVM, considering a few estimations of *n* in n-grams and TF/IDF standardization strategies on the Twitter dataset. For a combination of two versions of Crowdflower[3] [4] and hate speech[5] datasets they obtained an accuracy of 95.6% as best performance for LR classifier. Rajalakshmi et. al. [7] presented

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

[2] http://fire.irsi.res.in/fire/2020/home

[3] https://data.world/crowdflower/hate-speech-identification

[4] https://data.world/ml-research/automated-hate-speech-detection-data

[5] https://github.com/ZeerakW/hatespeech

an ensemble method using Logistic Regression, Support Vector Machine and Random Forest Classifiers (RFC) and submitted to HASOC 2019 shared task. They used Mutual Information and CHI square feature selection methods and applied for Subtask A to classify a given input as "HATE" or "NOT". Best performance of the system is reported for RFC using CHI square feature selection method with accuracy of 81% and 64% on German and Hindi language tweets respectively on the dataset provided by organizers. The framework presented by Hamada et. al. [8] in HASOC 2019 shared task includes the Stochastic Gradient Descent (SGD) optimization algorithm that has been utilized for optimizing the parameters of the linear classifier, while SVM with linear kernel utilizes Lagrange multipliers to tackle the improvement issue. The misfortune work used in the linear classifier was the "Pivot" loss function. Their model accomplished a overall macro-averaged F1-score of 66.12%, 42.36% and 42.23% for English subtask 1, 2 and 3 respectively using SVM, 46.37%, 23.46% for German Subtask 1 and 2 respectively using MLP, and 75.94%, 47.20% and 52.38% on Hindi Subtask 1 using MLP, Subtask 2 using SVM and Subtask 3 using MLP respectively.

3. Methodology

The architecture of the proposed Ensemble model to detect hate speech and offensive content using the training set provided by the organizers of HASOC 2020 is as shown in Figure1 and it contains the following modules:

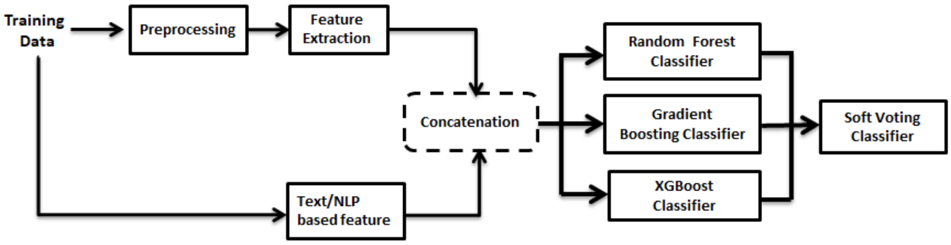

Figure 1: Architecture for building an Ensemble model

3.1. Preprocessing

Preprocessing transforms the textual content into a structure that is acceptable by the ML algorithms. The initial step in preprocessing is converting emojis to corresponding text in English language dataset and removing them in Hindi and German language dataset as we didn't get any libraries to convert emojis to text for these languages. Emojis are visual representation of emotions, object or symbol which can be inserted individually or together to create a string. As they are powerful representations of emotions converting them to text will give valuable words. All punctuation symbols, numeric data, stop words, frequently occurring words and uninformative words which do not contribute to the text classification are removed. Further, Lemmatization is applied to reduce the words into their root forms. The remaining words are given as input to the feature extraction step to extract features.

3.2. Feature Extraction

In this step, text data will be transformed into feature vectors using the following feature extraction methods:

- **CountVectorizer**[6] is a highly flexible feature representation for text and is a simple method of changing text to vector. It tokenizes the text to construct a vocabulary of the words present in the corpus and checks how frequently each word from the vocabulary is available in every single text in the corpus.
- **TF-IDF Transformer**[7] computes the IDF values by calling tfidf_transformer on the word counts. TF-IDF which speaks the relative importance of the word in the document and the whole corpus is a common method utilized in any text analysis application including text classification. It gains more information from longer documents and is ideal for problems with many words and larger document files.
- **Text based features** helps in improving text classification models. The total number of word count, character count, punctuation count, and the normal length of the words present in the corpus is utilized for all three languages. Further, upper case count, title word count, the frequency distribution of Part of Speech tags i.e., noun count, verb count, adjective count, adverb count, pronoun count is applied only to English language text.

The preprocessed text is converted into vector representation using CountVectorizer and TF-IDF transformer and is combined with text-based features to improve text classification models.

3.3. Classifier Construction

In this phase, the extracted features are used to train Random Forest (RF), Gradient Boosting (GB) and XGBoost Classifier models to identify hate speech and offensive content. RF classifiers are reasonable good for managing the high-dimensional noisy text data [9]. GB classifiers are a group of powerful ML algorithms that have indicated significant success in a wide range of applications. They are exceptionally altered to the specific needs of the application, like being learned with respect to different loss functions [10]. XGB classifier also has gradient boosting at its core. Nevertheless, the contrast between simple GB algorithm and XGB algorithm is that unlike in GB, the process of addition of weak learners does not happen one after the other; it takes a multi-threaded approach whereby proper utilization of the CPU core of the machine is utilized, leading to greater speed and performance [11].

The performance of a classifier depends on the dataset as well and no classifier gives good results for all the datasets. Hence, no classifier can be generalized as the best classifier. So instead of considering a single classifier, an ensemble of classifiers where the weakness of one classifier is overcome by the strength of other classifier is proved to give good results when

[6] https://scikit-learn.org/stable/modules/generated/sklearn.feature$_{e}xtraction.text.CountVectorizer.html$

[7] https://scikit-learn.org/stable/modules/generated/sklearn.feature$_{e}xtraction.text.TfidfVectorizer.html$

compared to an individual classifier. In this work, an ensemble of three classifiers namely, RF, GB and XGB is used for the prediction of input text based on soft voting.

The ensemble classifier accepts the test data provided by the organizers of HASOC 2020 and assigns either a 'NOT' or 'HOF' label for Subtask A and one of 'HATE', 'PROF', and 'OFF' for Subtask B after preprocessing and feature extraction. Architecture of Ensemble model for predicting the class label of test data is shown in Figure 2.

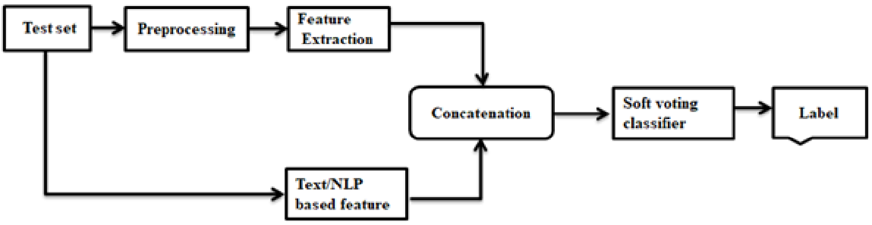

Figure 2: Ensemble model for predicting the class labels of test data

4. Experiments and Results

To study the performance of the proposed method on English, German and Hindi datasets, various experiments were conducted and submitted to the organizers of HASOC 2020 for further evaluation. However, in this paper, we have described the model selected by the organizers as the best among our other models. The code is implemented in Python using Scikit-learn, Machine Learning in Python[8].

Training and development dataset provided by the organizers of HASOC 2020 shared task for two Subtasks A and B for English, German and Hindi is shown in Table 1. Statistics of dataset show that for English Subtask A, both training and development set are balanced but German and Hindi training and development set are imbalanced. Further, for English Subtask B training and development set are heavily imbalanced and German and Hindi training and development set are not balanced. As all posts with 'NOT' labels in Subtask A will be 'NONE' for Subtask B, the posts with 'NOT' labels are excluded during training models for Subtask B and for submission they are added with 'NONE' label directly. The submissions were evaluated on 15% of the private dataset and the results obtained in terms of F1 Macro average and ranks are shown in Table 2.

[8] https://scikit-learn.org/stable

Table 1
Details of datasets provided by HASOC 2020

Subtasks	No. of posts	Train set			Development set		
		English	German	Hindi	English	German	Hindi
Subtask A	**HOF**	1856	673	847	423	134	197
	NOT	1852	1700	2116	391	392	466
Subtask B	**PRFN**	1377	387	148	293	88	27
	HATE	158	146	234	25	24	56
	OFFN	321	140	465	82	36	87
	NONE	1852	1700	2116	414	378	493
Total		**3708**	**2373**	**2963**	**814**	**526**	**663**

Table 2
Performance of proposed approach in terms of F1 Macro average

Language Subtask	English		German		Hindi	
	F1 Macro average	Rank	F1 Macro average	Rank	F1 Macro average	Rank
Subtask A	0.5046	5	0.5106	7	0.5033	13
Subtask B	0.2596	15	**0.2595**	**3**	0.2488	6

5. Acknowledgements

We would like to thank HASOC 2020 shared task organizers for organizing this interesting shared task and for replying promptly to all our inquiries. We further thank the anonymous reviewers for their valuable feedback.

6. Conclusion

In this paper, we describe the work submitted by our team MUM to HASCO 2020 shared task in FIRE 2020 to identify hate and offensive content in Indo-European Languages. The problem of identifying hate and offensive content is studied experimentally on English, German and Hindi language datasets provided by the organizers. Using an ensemble of three classifiers namely, Random Forest, Gradient Boosting and XGBoost, with soft voting our team obtained competitive results for both the subtasks of all three languages. We would like to explore more features and more classifiers to identify hate and offensive content in Indian languages.

References

[1] R. Kumar, A. K. Ojha, S. Malmasi, M. Zampieri, Benchmarking aggression identification in social media, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 1–11.

[2] 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 Iden-

tification in Indo-European Languages), in: Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, CEUR, 2020.

[3] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages, in: Proceedings of the 11th Forum for Information Retrieval Evaluation, 2019, pp. 14–17.

[4] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, arXiv preprint arXiv:1703.04009 (2017).

[5] U. Saha, A. Dubey, P. Bhattacharyya, Iit bombay at hasoc 2019: Supervised hate speech and offensive content detection in indo-european languages., in: FIRE (Working Notes), 2019, pp. 352–358.

[6] A. Gaydhani, V. Doma, S. Kendre, L. Bhagwat, Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach, arXiv preprint arXiv:1809.08651 (2018).

[7] R. Rajalakshmi, B. Y. Reddy, Dlrg@ hasoc 2019: An enhanced ensemble classifier for hate and offensive content identification., in: FIRE (Working Notes), 2019, pp. 370–379.

[8] H. A. Nayel, H. Shashirekha, Deep at hasoc2019: A machine learning framework for hate speech and offensive language detection., in: FIRE (Working Notes), 2019, pp. 336–343.

[9] M. Z. Islam, J. Liu, J. Li, L. Liu, W. Kang, A semantics aware random forest for text classification, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1061–1070.

[10] A. Natekin, A. Knoll, Gradient boosting machines, a tutorial, Frontiers in neurorobotics 7 (2013) 21.

[11] S. Ramraj, N. Uzir, R. Sunil, S. Banerjee, Experimenting xgboost algorithm for prediction and classification of different datasets, International Journal of Control Theory and Applications 9 (2016) 651–662.