# IIIT-Hyderabad at HASOC 2019: Hate Speech Detection

Vandan Mujadia, Pruthwik Mishra, Dipti Misra Sharma

MT & NLP Lab, LTRC, IIIT-Hyderabad
{pruthwik.mishra, vandan.mu}@research.iiit.ac.in, dipti@iiit.ac.in

**Abstract.** Automatic identification of offensive language in various social media platforms especially Twitter poses a great challenge to the AI community. The repercussions of such writings are hazardous to individuals, communities, organizations and nations. The HASOC shared task attempts for automatic detection of abusive language on Twitter in English, German and Hindi languages. As a part of this task, we (team A3-108) submitted different machine learning and neural network based models for all the languages. Our best performing model was an ensemble model of SVM, Random Forest and Adaboost classifiers with majority voting.

**Keywords:** Machine Learning · Neural Networks · Adaboost · LSTM · Linear SVM · Random Forest · TF-IDF

## 1 Introduction

Social media is a great platform to communicate with people from different demographic groups. With the exponential rise of use of hand-held devices across the world, people are spending considerable amount of time on social media like Facebook, Twitter, Instagram. Recent studies [2] suggest that most of the online content generated on these platforms contains different forms of abusive language. Cyberbullying and cyberterrorism have become a big menace for the human society. A lot of disparaging tweets [9] target people based on their color, race, ethnicity, nationality, religion, caste. The administrators of social media have started employing methods to tackle the adversarial effects of the contents being generated at their ends. HASOC [8] tries to automatically identify hate speech and hurtful language in 3 different languages namely English, German and Hindi. The distribution of different labels across tasks for each language in the provided training data is shown table 1. The first task is a binary classification task to identify whether a tweet is offensive or not. The other two tasks deal with finer categories of hate speech and offensive posts.

## 2 Approach

Two kinds of approaches were employed for the identification tasks.

| Language | Task1 | | Task2 | | | | Task3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | HOF | NOT | HATE | NONE | OFFN | PRFN | NONE | TIN | UNT |
| English | 2261 | 3591 | 1143 | 3591 | 451 | 667 | 3591 | 2041 | 220 |
| German | 407 | 3412 | 111 | 3412 | 210 | 86 | - | - | - |
| Hindi | 2469 | 2196 | 556 | 2196 | 676 | 556 | 2196 | 1545 | 924 |

**Table 1.** Class Distribution Across Tasks and Languages

– Machine Learning Techniques
– Neural Network Approaches

### 2.1 Prepossessing

Preprocessing is essential when we are dealing with textual data. For machine learning approaches, we used the spacy [1] tokenizer for English and German, the nltk [2] Twitter tokenizer for tokenizing the input. We also normalized the Twitter handles and hashtags as "USRTOK", and urls as "URLTOK".

### 2.2 Feature Engineering

We used generic features for the representation of each tweet as we did not want to design any language specific features. Each word appearing in the tweet was reduced to its lemma for English and German. We did not lemmatize the Hindi words as there was no publicly available spacy model. The features were TF-IDF vectors at character and word levels for all the tasks. We also used length of a tweet as a feature. We experimented with different kinds of classifiers. Each one was trained either individually or was a part of an ensemble classifier. Different voting procedures were also tried out. In hard voting, majority voting is carried out among the participating classifiers to decide each label. A voting classifier with soft voting selects the maximum from the computed sums of the predicted probabilities for the constituent classifiers. The following were implemented using scikit-learn [10] machine learning library.

– Linear SVM
– Adaboost or Adaptive Boosting (AB)
– Random Forest (RF)
– Voting Classifier (VC)

We tried various combinations of word and character level n-grams for the classification. By performing grid-search, we observed that combining word unigrams and character n-grams where $n \epsilon \{2, 3, 4, 5\}$ TF-IDF vectors as well as the combination of character and word level n-grams.

[1] `https://spacy.io`
[2] `http://www.nltk.org`

### 2.3 Neural Network Models

We also used LSTM based neural network classifiers for all subtasks. We used word features, Embedding layer (300 dimensions) as the inputs to the LSTM [4] (128 units) layer, softmax layer (for prediction) in the keras [3] toolkit. In this pipeline, we used categorical_crossentropy as the loss function, the Adam optimizer to optimize the parameters [5] and trained on 50 epochs. In section 3, we show and discuss results in detail.

### 2.4 Our Submissions

We submitted 3 runs for each task in each language. The $1^{st}$ run was the LSTM based approach. The $2^{nd}$ run was an ensemble of SVM, Random Forest and Adaboost classifiers with hard voting. This classifier used TF-IDF features of word unigrams and character 2, 3, 4, 5 grams. The $3^{rd}$ submission was similar to the $2^{nd}$ one with an additional feature of length of every tweet.

## 3 Results

Different classifiers were trained to predict the class of each question. We include the top performing system outputs in table 2. Each model and feature set is shown in the table. Two metrics were used to evaluate the systems. Macro F1 score was the primary metric whereas weighted F1 was the secondary one. Macro F1 is an unweighted mean of the metrics calculated for each label. Weighted F1 is obtained by assigning weights based on the number of samples for each true label.

## 4 Observations

Some tweets are ambiguously labeled.

- "All the best to #TeamIndia for another swimming competition on Sunday against #Pakistan. #INDvPAK #ShameOnICC #CWC19 #CWC19Rains https://t.co/MG2cIE0zib"
- "#ShameOnICC 1. ICC on Dhoni's gloves Vs 2.ICC planning the World Cup https://t.co/4kO3zKt7ln"

The first tweet is not offensive while the second one is marked as offensive in the training set. Although both of them are related to a similar topic, a classifier trained on these kinds of examples will predict them as non-offensive as the number of non-offensive tweets was more. Adaboost [3] was the best performing classifier among the three classifiers used in our submissions. This was due to its ability to combine multiple weak classifiers to create a strong prediction model. But an ensemble of SVM, Random Forest and Adaboost performed even

[3] `https://keras.io`

| Language | Task# | Run# | Model | Features | Macro F1 | Weighted F1 |
|---|---|---|---|---|---|---|
| English | 1 | 1 | LSTM | word embeddings | 0.6015 | 0.6714 |
| | | 2 | SVM+RF+AB | word-uni+char2-5grams | 0.6895 | 0.7591 |
| | | 3 | SVM+RF+AB | word-uni+char2-5grams+lengthOfTweet | 0.6970 | 0.7688 |
| | 2 | 2 | SVM+RF+AB | word-uni+char2-5grams | 0.4142 | 0.7302 |
| | | 3 | SVM+RF+AB | word-uni+char2-5grams+lengthOfTweet | 0.4172 | 0.7250 |
| | 3 | 1 | LSTM | word embeddings | 0.3874 | 0.6565 |
| | | 2 | SVM+RF+AB | word-uni+char2-5grams | 0.4377 | 0.7516 |
| | | 3 | SVM+RF+AB | word-uni+char2-5grams+lengthOfTweet | 0.4221 | 0.7412 |
| German | 1 | 1 | LSTM | word embeddings | 0.4481 | 0.7329 |
| | | 2 | SVM+RF+AB | word-uni+char2-5grams | 0.4633 | 0.7686 |
| | | 3 | SVM+RF+AB | word-uni+char2-5grams+lengthOfTweet | 0.477 | 0.7728 |
| | 2 | 1 | LSTM | word embeddings | 0.2455 | 0.747 |
| | | 2 | SVM+RF+AB | word-uni+char2-5grams | 0.2283 | 0.767 |
| | | 3 | SVM+RF+AB | word-uni+char2-5grams+lengthOfTweet | 0.2283 | 0.767 |
| Hindi | 1 | 1 | LSTM | word embeddings | 0.7468 | 0.7483 |
| | | 2 | SVM+RF+AB | word-uni+char2-5grams | 0.8032 | 0.8038 |
| | | 3 | SVM+RF+AB | word-uni+char2-5grams+lengthOfTweet | 0.8024 | 0.8031 |
| | 2 | 1 | LSTM | word embeddings | 0.499 | 0.6136 |
| | | 2 | SVM+RF+AB | word-uni+char2-5grams | 0.5253 | 0.6522 |
| | | 3 | SVM+RF+AB | word-uni+char2-5grams+lengthOfTweet | 0.5113 | 0.6416 |
| | 3 | 1 | LSTM | word embeddings | 0.4888 | 0.6484 |
| | | 2 | SVM+RF+AB | word-uni+char2-5grams | 0.5559 | 0.7449 |
| | | 3 | SVM+RF+AB | word-uni+char2-5grams+lengthOfTweet | 0.5754 | 0.7361 |

**Table 2.** Accuracy of Models on Test Data

better than each classifier. Soft voting performed worse than the technique of hard voting while the final predictions were made. In Twitter, the number of spelling variations is high due to character constraints. So character n-gram based TF-IDF was superior to its word counterparts individually. When we combined both word and character n-gram, the increase in performance was marginal. Adding length of the tweet as a feature did not improve the model. It is wrong to assume that the tweets containing abusive language are usually short. Machine learning approaches outperformed the neural networks in almost all the tasks. This conforms to the hypothesis that the machine learning techniques are superior their neural network counterparts in a low resource setting. This could be due to the higher number of parameters that deep learning approaches try to learn from a very limited amount of data. We also figured out that the

classifiers performed well when the classes were balanced. Predicting profane tweets was difficult as the frequency of such tweets was the least across the data for each language. All our classifiers performed very poorly for all the tasks in German. The systems were unable to capture any form of hate speech. A lexicon containing German slur, profane and abusive words can prove to be useful. Usually an offensive tweet is full of words portraying negative sentiments. German sentiment lexicons can be looked up to identify such tweets.

We also observed that we missed a lot of important cues when we replaced all Twitter handles and hashtags by a generic token. "#BorisJohnsonShouldNotBePM", "#bloodonhishands", "#TrumpIsATraitor" are made up of multiple words. These words in isolation can be a useful feature for the identification task.


**Fig. 1.** EN word clusters


**Fig. 2.** HI word clusters


**Fig. 3.** DE word clusters

From the above results, one can do case by case analysis to find out the improvement possibilities that can help different classifiers on different tasks.

We also performed an analysis to understand the complexity of the task by automatically applying Latent Dirichlet Allocation (LDA) on the provided training data. Before applying LDA [1], by using Gensim toolkit [11], we performed basic tokenizaion [6], text normalization and stop-word removal. For this analysis, we focused only on sub-task-1 ( Non Hate-Offensive vs Hate and Offensive) to understand the task difficulty on a coarse level. Figures 1, 2 and 3 show these derived text clusters from LDA for English (EN), Hindi (HI), German (DE) respectively. From this clusters, we can argue that they do not give any inherent separation on given task labels.

As a next step to this, we also used learned LDA model of subtask-A or task-1 to visualise training data by plotting each instances using T-SNE [7] in 2D. Figures 4, 4 and 4, represent the training text and corresponding labels that we got from LDA (left) with blue and orange colors for respective languages . The graph on the right represents the actual labeled data in two dimensions for respective languages. We can observe from figures that the given task of classification is quite difficult as simple topic modeling do not provide any major incites for the classification. This also suggests the need of feature engineering and use of external resources for further improvement.

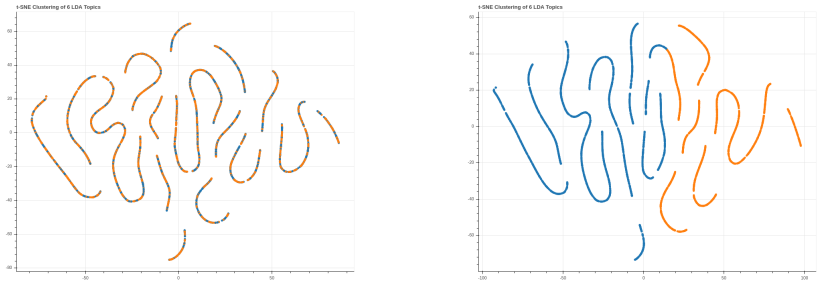

**Fig. 4.** T-SNE representation of data labeled with LDA (left) and actual labeled (right) on subtask-A (EN).

## 5 Conclusion and Future Work

We presented our supervised approaches for the FIRE task of Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC). Our experiments provide strong cues to go for traditional machine learning algorithms with feature engineering instead of recent neural network based approaches when the number of samples is very few and the class distribution is heavily skewed. An ensemble classifier with word and character TF-IDF features performed the best among all the classifiers. Detecting offensive language in tweets is hard

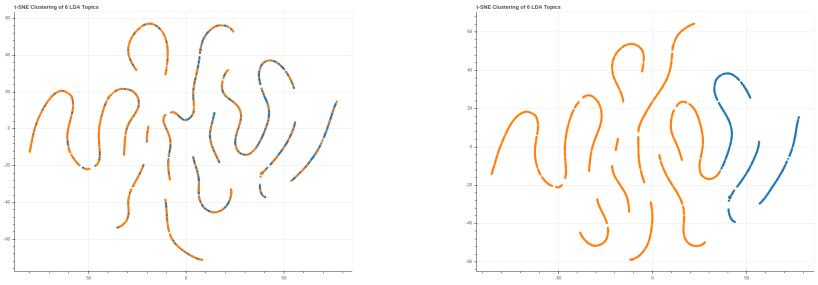

**Fig. 5.** T-SNE representation of data labeled with LDA (left) and actual labeled (right) on subtask-A (HI).

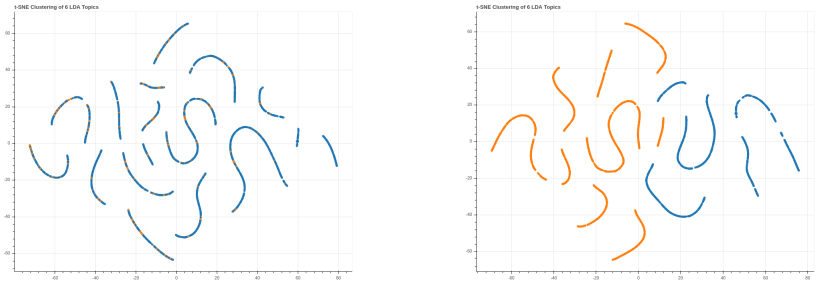

**Fig. 6.** T-SNE representation of data labeled with LDA (left) and actual labeled (right) on subtask-A (DE).

when explicit keywords indicative of such forms are missing e.g "I don't know how much more I can take! 45 is a compulsive liar! #Trump30Hours #TrumpIsATraitor". We can explore unsupervised techniques on raw tweets for learning a better representation of implicit form of hate speech. Convolutional neural networks (CNN) could be used to model the interactions between character n-grams in the tweets.

## References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
2. Chen, Y., Zhou, Y., Zhu, S., Xu, H.: Detecting offensive language in social media to protect adolescent online safety. In: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing. pp. 71–80. IEEE (2012)
3. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences **55**(1), 119–139 (1997)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)

5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. Loper, E., Bird, S.: Nltk: the natural language toolkit. arXiv preprint cs/0205028 (2002)
7. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
8. Modha, S., Mandl, T., Majumder, P., Patel, D.: Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In: Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation (December 2019)
9. Nockleby, J.T.: Hate speech. Encyclopedia of the American constitution **3**, 1277–1279 (2000)
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
11. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), `http://is.muni.cz/publication/884893/en`