Machine Learning Models for Hate Speech and Offensive Language Identification for Indo-Aryan Language: Hindi

Purva Mankar [1], Akshaya Gangurde [2], Deptii Chaudhari [3] and Ambika Pawar [4]

[1] *Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune*
[2] *Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune*
[3] *Hope Foundation's International Institute of Information Technology, Pune*
[4] *Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune*

Abstract

Automated recognition and detection of Hate Speech and Offensive language on different Online Social Networks, mainly Twitter, presents a challenge to the community of Artificial Intelligence and Machine Learning. Unfortunately, sometimes these ideas communicated via the internet are intended to promote or incite hatred or humiliation of an individual, community, or even organizations. The HASOC shared task is to attempt to automatically detect abusive language on Twitter in English and Indo-Aryan Languages like Hindi. To participate in this task and provide our input, we Team Data Pirates presented several machine learning models for Hindi Subtasks. The datasets provided allowed the development and testing of supervised machine learning techniques. The top 2 performing models for sub-task A were Naïve Bayes and Logistic Regression with the same Macro F1 score of 0.7394. The top 2 performing models for sub-task B were Logistic Regression and CatBoost, with Macro F1 scores of 0.4828 and 0.4709, respectively. This overview intends to provide detailed understandings and to analyze the outcomes.

Keywords

Hate Speech, Machine Learning, TF-IDF, Logistic Regression, Text Classification, CatBoost, HASOC

1. Introduction

Social media has brought people from different demographic areas closer than ever. It has become a space for people to build and grow together. It has become a hub to share your thoughts and opinions and reach a broad audience. Now, much constructive work takes place on these digital platforms. However, there are also many negative things that found their existence with the rise of social media. The anonymity that comes up with these platforms and made people willingly support Hate Speech towards a community, religion, or race[9]. The language that contains Hate Speech and Profanity has severely affected people in their lives and behavior, leading them to depression and even suicide. Detection of Hate Speech has been a challenge but several research efforts from over the globe over the past years have worked and identified the Hate content using Natural Language Processing and Machine Learning[1]. A significant technique for progressing such systems is to employ supervised learning with an annotated dataset. Considerable work has been done in several languages, with English being one of them. However, for most other languages, there is a dearth of research on this topic[6].

The Hate Speech and Offensive Content Identification (HASOC) provides a data challenge for research on identifying inappropriate content. We participated in the identification of Hate content in the Hindi Language. The organizing team provided altogether thousands of annotated tweets from Twitter. We (Data Pirates) as a team took part in both Subtasks A (Identifying Hate, offensive, and profane content) and Subtask B (Discrimination between Hate, profane and offensive posts).

Forum for Information Retrieval Evaluation, December 13-17, 2021, India
EMAIL: purva.mankar.btech2018@sitpune.edu.in (P. Mankar); akshaya.gangurde.btech2018@sitpune.edu.in (A. Gangurde); deptiic@isquareit.edu.in (D. Chaudhari); ambikap@sitpune.edu.in (A. Pawar)

Sub-task A: Hate and Offensive (HOF) and Non-Hate-Offensive (NOT)

Sub-task B: Hate (HATE), Offensive (OFFN), Profane (PRFN)

Once the dataset was pre-processed, we applied Machine Learning techniques to extract features using methods like TF-IDF, Countvectorizer, and Word2Vec. Various experiments were conducted using Machine Learning techniques to design a predictive model. Models such as Logistic Regression, Support Vector Machine, Random Forest Classifier, Naïve Bayes, CatBoost were made in use. Experiments with the Naive Bayes model and Logistic Regression model on Subtask A gave us the most accuracy, and for Subtask B Logistic Regression model outshined the rest of the models followed by CatBoost Classifier.

The paper structure follows as Section 2 presents the previous work and dataset from collections. Section 3 describes the description of sub-tasks from HASOC. Section 4 presents the approach used for the Twitter dataset, while Section 5 presents our submission, and Section 6 gives a detailed description of results for the assigned HASOC tasks, followed by Section 7 with observations drawn from these experiments. Finally, for Section 8 and Section 9, we have conclusions for the paper and future work for these tasks.

2. Previous Work and Dataset

Significant work has been done for various dialects, including English. However, there is not much research and work for different regional languages like Hindi and Marathi. Collections like HASOC play a vital role in methods that use supervised classification mechanisms[13]. For the detection of Hate Speech on online platforms, several previous initiatives have created a corpus that professionals can use to solve the issue of the presence of Hate Speech on the internet. HASOC is an attempt to generate a labelled dataset for a language with few resources.

Numerous languages apart from English are in a growing market. HASOC would be the first collaborative project that created a database for three languages and promoted multilingual study.

Many people trying to create Hate Detection methods confront difficulties with data collection and data sampling. Data sampling is a crucial job in any data challenges competition[2]. These statistics include tweets about themes such as hatred towards women, racism, immigration, harsh political comments, and comments directed at celebrities. A current tendency is to implement a fine-grained categorization. Some data issues need comprehensive analysis for hateful remarks, such as detecting the target or the type of hate speech. In contrast, others focus on the intensity of the post.

3. Description of Subtasks

To detect hateful matter on Online Social Media platforms by Machine Learning and Deep Learning models, HASOC provides only the textual content of the post and leaves out meta-data like time associated features or the network of the sender and recipient or the context of the post which makes these tasks impractical to some extent. Due to legal obligations, HASOC Platform does not distribute a user's meta-data for a post. HASOC 2021 had offered the following subtasks:

Sub-task A: This sub-task mainly aims at the binary classification of Hate speech and Offensive language identification and is offered for English, Hindi, and Marathi. We chose our language as Hindi for model building and evaluation. The submitted model is expected to categorize the tweets into two classes, i.e., Non- Hate Offensive (NOT) and Hate and Offensive (HOF).

1. Non-Hate Offensive (NOT) - Said tweet does not include any Hate Speech and/or Offensive content.
2. Hate and Offensive (HOF) - Said tweet includes Hate, Offensive, and/or Profane content.

The training dataset was labelled and annotated for both the sub-tasks. The testing dataset only included the tweet content. The model must separately predict the labels for both sub-tasks.

Sub-task B: This sub-task targets the fine-grained classification of sub-task A. Tweets marked as HOF in sub-task A are additionally categorized into three classes.

1. (OFFN) Offensive: Tweets comprising offensive content.
2. (HATE) Hate speech: Tweets comprising Hate speech content.
3. (PRFN) Profane: Tweets comprising profane words.

OFFENSIVE: Something that is likely to elicit sentiments of hurt, wrath, contempt, disapproval, or revulsion, or is linked with an aggressive attack, is defined as offensive. This category includes tweets and posts that are degrading, dehumanizing, insulting, or threatening with violent acts.

HATE: Distinguishing a group of people based on their commonalities or differences (e.g., all poor people are stupid). As a result of racism, political opinions, gender preference, sexual identity and other factors such as status in society and health, hateful statements are aimed towards specific groups of people or groups of people in general.

PROFANE: Profane language is censored on television. The term profane can also refer to incredibly insulting behavior that demonstrates a level of regard, particularly for someone's religious convictions. In the lack of slurs and abusive behavior, this is inappropriate language. This usually refers to the use of profanities like (Damn, Fuck, and so on) and cursing (Hell! Holy shit! and so on). These comments are classified as belonging to this class.

NONE: As expected, most posts in the category NOT in sub-task A are labelled as NONE in sub-task B.

4. Approach

4.1 Dataset and Collection

The organizing team supplied the training and testing datasets for the Hindi language. The shared HASOC task included two subtasks for the Hindi language datasets. Sub-task A of HASOC is a Binary Classification that needs tweets to be distinguished into either HOF (Hate and Offensive) or NOT (Non-Hate-Offensive).

Sub-task B is a fine-grained classification with the Hate Speech further classified into four categories: NONE, OFFN, HATE, PRFN. The training dataset has a size of 4594 tweets, and the testing dataset has 1532 tweets. Table 1 gives a detailed description of datasets used in this work for both Sub-task A and Sub-task B. Machine Learning approach was followed for both HASOC tasks.

Table 1

Dataset description of Hindi Corpus

Subtasks	Labels	Training Samples
Sub-task A	NOT	3161
	HOF	1433
Sub-task B	NONE	3161
	OFFN	654
	HATE	566
	PRFN	213

A binary score was assigned to each label in the annotated training dataset, which the models will then use to predict labels in unseen test data.

Table 2

Binary Scoring for sub-task A

Score	Class
0	NOT
1	HOF

Sub-task B: To understand the labels, whether it was NONE, PRFN, HATE, or OFFN we gave a 4-rating score for the annotated dataset, which the models the models will then use to predict labels of an unlabeled test dataset.

Table 3
4-Rating Score for sub-task B

Score	Class
0	NONE
1	OFFN
2	HATE
3	PRFN

Table 4
Sample Tweets from all classes of labels

Classes	Example of the tweet from that class
NOT	केंद्रीय मंत्रियों के बंगाल प्रवेश के लिए निगेटिव आरटीपीसीआर रिपोर्ट जरूरी! - ममता #ModiKaVaccineJumla #MamtaBanerjee
HOF	देश की सारी मीडिया अपनी माँ चूda रहे है जो बंगाल पर मुँह बंद रखा है। 😠😠😠 #बंगाल_हिंसा #BengalBurning
OFFN	सड़क पर बैठी गाय किसी का क्या बिगाड़ रही थी, जो इस सुअर के पिल्ले ने कुचल कर उसको मार डाला इन सूअरों को नीचता के स्तर को समझ पा रहे हो हिंदुओ ??? 😡😡🤯☪️🐷👳🕌📻💣🚩 https://t.co/3z3IUXlE3U
HATE	@aajtak @iSamarthS @abhishek6164 @chitraaum यदि शकल अच्छी न हो तो भी आदमी अच्छा अच्छा तो बोल ही सकता है । मेरे इतना कहने पर नासिका की शल्यचिकित्सा नहीं करवाई, शायद किसी लक्ष्मण का इंतजार है । नाक वक्र होने के कारण जिह्वा भी वक्र हो गई है और उसकी फूहड़ बातों पर अट्टहास लगाते प्यादे भी अंजाम से अनजान...
PRFN	जितिन प्रसाद की जरूरत ही नहीं ऐसे चुटिया लोग को पार्टी में रखना ही नहीं चाहिए ये सूअर खाने वाले लोग हैं अभी और भी आएंगे सामने देखते जाओ 2022 तक गद्दारी बराबर मिलेगी देखने को 🤩🤩🤩 @priyankagandhi
NONE	" डॉ मोहम्मद शहाबुद्दीन साहब एक हीरो थे या विलन ...!! https://t.co/8ZvXwBShq6 #ShahabuddinSaheb #Shahabuddin #JusticeForShahabuddin

4.2 Pre-processing

Text pre-processing is used to prepare text data for model building. It would be the first step in any NLP project. Pre-processing steps include removing punctuation such as (.,! $( ) * % @), URLs, stop words, lower casing, tokenization, stemming, and lemmatization. We used the library of the regular expression, and nltk Twitter tokenizer to tokenize the machine learning methodologies' input.

Stop word removal: Stop words are often used terms that are eliminated from the text because they provide no value to the analysis. These words have little to no meaning. the We made a custom text file

for the Hindi language, including all the stop words that weren't necessary for model building. The clean text was free from URLs, stop words, Hashtags and ready to be fed into the system.

4.3 Feature Engineering

Suitable feature extraction is important in text classification once data has been pre-processed. In this work, we used the Bag-of-Words approach (TF-IDF, Countvectorizer) and Word embedding models like Word2Vec.

1. TF-IDF:
   Term Frequency- Inverse Document Frequency (TF-IDF) is a statistic based on the frequency of the word present in every tweet present in the corpus. It gives out a numerical representation of how substantial a word is for analysis.

2. Countvectorizer:
   Countvectorizer works using Terms Frequency, which entails counting the number of times tokens appear in a document and constructing a sparse matrix of documents x tokens.

3. Word2Vec:
   Word embedding techniques such as Word2Vec produce distributed representations that account for semantics, allowing words with similar meanings to be found close together in vector space. The dimension of vectors generated by Word2vec is similarly limited. As a result, Word2Vec is one of the best options for converting words to vectors.

We pre-processed the dataset with proper lemmatization and stop words and applied methods like the fit_transform() to fit and transform the training dataset to scale it and learn its scaling parameters. Here, the model we developed will learn the mean and variance of the training set's characteristics. Our test data is then scaled using the parameters we've learned.

Sub-task A is a binary classification task. We used models like Logistic Regression and Naïve Bayes to predict the accuracy of the model.

Table 5
Accuracy for top 2 models for sub-task A

Sr.no	Model	Training Accuracy
1.	Logistic Regression (TF-IDF)	83%
2.	Naïve Bayes	90%

Sub-task B is a multiclass classification task. We used models like Logistic Regression and CatBoost to predict the accuracy of the model.

Table 6
Accuracy for top 2 models for sub-task B

Sr.no	Model	Training Accuracy
1.	Logistic Regression	98%
2.	CatBoost	79%

4.4 Assessment Metrics

Classification metrics should mix precision and recall. The F1-score has several variations, such as weighted F1, macro-F1, and micro-F1. The distribution of class labels is frequently uneven in multi-class classification. The weighted F1-score determines the F1 score for each class separately. Once it

combines, it gives each class a weight based on the number of true labels. As a result, it favors the majority. The 'macro' generates the F1 individually for every class but still doesn't utilize weights in the aggregate. This leads to harsher penalties when a system fails to function well for minority groups. The version of the F1-measure used is determined by the task's aim and the distribution of labels in the dataset. Class inequality contributes to hate speech classification issues. As a result, the macro F1 is an obvious choice for the evaluation.

The submissions were ranked based on Macro F1 scores. A classification report is the best option for a detailed report, as it clearly illustrates Precision, Recall, and F1 Score for distinct label predictions.

5. Our Submission on leader board

We submitted five runs for each sub-task for the Hindi language. The 1st run for sub-task A submitted by us used Logistic Regression with TF-IDF. A more sophisticated format was employed to convert the text into numeric vectors: TF-IDF. According to this method, each word counts is divided by the number of documents where it appears to arrive at a normalized count for each word. Logistics regression is a classifier that is used to deal with binary classification difficulties. The logistic regression classifier employs a weighted combination of the input characteristics, which are then passed via a sigmoid function. The Sigmoid function converts any real number to a number between 0 and 1.

Our 2nd submission was the Naïve Bayes model. A Naive Bayes classifier considers that one feature in a class does not affect the presence of any other feature. Our 3rd and 4th runs were XGBoost and Random Forest respectively. XGBoost is a method of ensemble learning. It is not always adequate to depend just on the findings of a single machine learning model. Ensemble learning provides a methodical approach to combining the predictive potential of several learners. The result is a single model that aggregates the output of multiple models. Random Forest also uses ensemble learning, a technique that combines multiple classifiers to solve complicated problems. A random forest method is made up of a large number of decision trees. Our 5th model was again a variation of the Naïve Bayes model from the 2nd run.

For sub-task B, our 1st run, which outperformed others, was the Logistic Regression model with Countvectorizer—followed by our 2nd run, which was CatBoost. It offers two vital algorithmic advances: ordered boosting, a permutation-driven variant to the traditional method, and a novel technique for processing category data. Both approaches employ randomized permutations of the training samples to combat the prediction shift produced by a specific type of target loss found in all current gradient boosting algorithm systems. Our 3rd and 4th runs were based on Random Forest and Naïve Bayes. Finally, the last run of the competition was on Support Vector Machines. It is a supervised machine learning method that may be used to solve classification and regression problems.

6. Results

A total of 5 run submissions were allowed for the Hindi Language for both Subtasks A and B. Logistic Regression with TF-IDF and Multinomial Naïve Bayes had the same Macro F1 score (0.7394) while XGBoost had a relatively small difference compared to former models. Similarly, a small difference was observed between Logistic Regression and CatBoost model with Macro score of 0.4828 and 0.4709 respectively. The positions on the leader board were based on the Macro F1 performance of the model.

Table 7
Performance metrics for top 2 models for sub-task A

Model	Macro F1	Macro Precision	Macro Recall	Accuracy
Logistic Regression (TF-IDF)	0.7394	0.7716	0.7255	78.721%
Naïve Bayes	0.7394	0.7563	0.7297	78.068%

Table 8
Performance metrics for top 2 models for sub-task B

Model	Macro F1	Macro Precision	Macro Recall	Accuracy
Logistic Regression	0.4828	0.5222	0.4606	70.17%
CatBoost	0.4709	0.5968	0.4406	72.977%

Every team was allowed five submissions for each sub-task, and the highest performing model was listed on the leader board. ***Figure 1*** shows how the models performed. Linear fashion is prevalent in the top 3 models which give the most accurate results among all five models.

Table 9
Submitted models and Macro F1 Scores for sub-task A

Sr. No.	Model	Macro F1
1	Logistic Regression	0.7394
2	Naïve Bayes	0.7394
3	XGBoost	0.7317
4	Random Forest	0.7180
5	Naïve Bayes (old)	0.5442

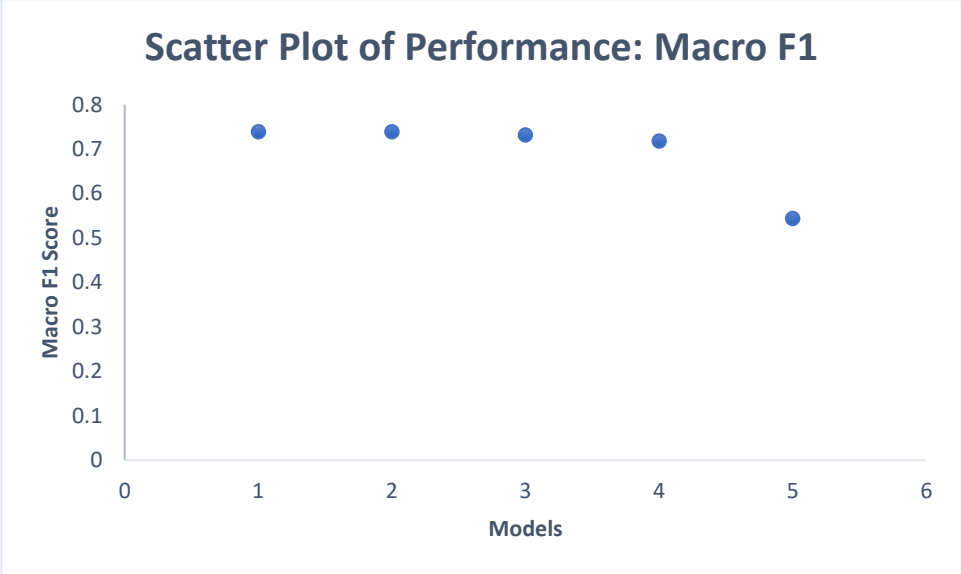

Figure 1: Scatter Plot of Macro F1 Scores for sub-task A

Figure 2 shows the graphical representation of how the models achieved the results. Linear fashion is prevalent in the top 3 models, giving the most accurate results among all five models while the last two models did not reach the mark. Logistic Regression scored the highest, with a score of 0.4828.

Table 10
Submitted models and Macro F1 Scores for sub-task B

Sr. No.	Model	Macro F1
1	Logistic Regression	0.4828
2	CatBoost	0.4709
3	Random Forest	0.4563

4	Naïve Bayes	0.3955
5	SVM	0.3933

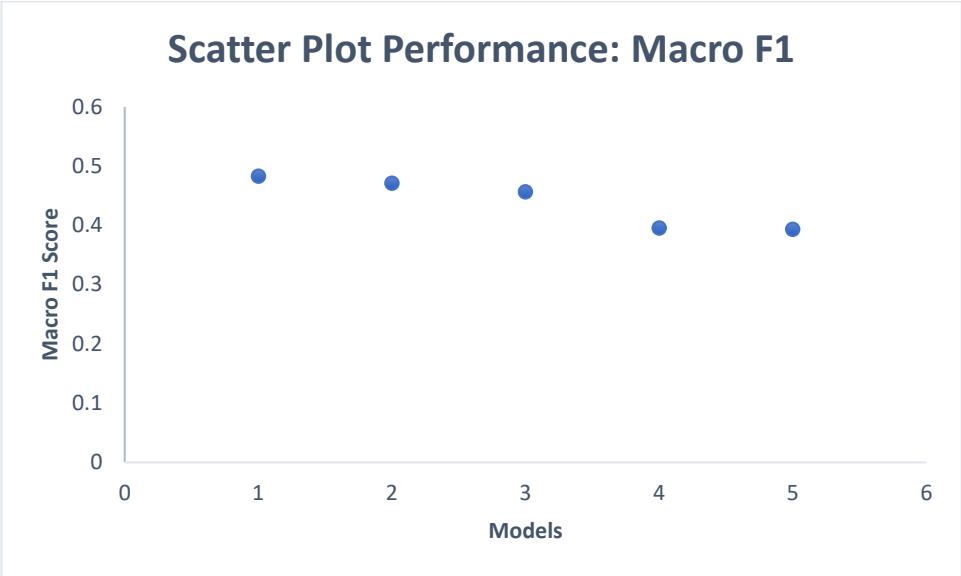

Figure 2: Scatter Plot of Macro F1 Scores for sub-task B

7. Observations

For Sub-task A, the number of NOT tweets in the training dataset are more than HOF. There were 3161 NOT and 1433 HOF entries in the training dataset, which may have affected the prediction performance because of the uneven distribution. The best models for Sub-task A were Naïve Bayes and Logistic Regression, from which Naive Bayes predicted the labels for the testing dataset 1115 as 'NOT' and 418 as 'HOF', which tags each tweet assigned a probability value. Then the tag with the highest probability is returned. Our second model submission, Logistic Regression followed by predicting 1161 tweets as Non-Hate-Offensive and 371 as Hate and Offensive. Both the models had the same Macro F1 score.

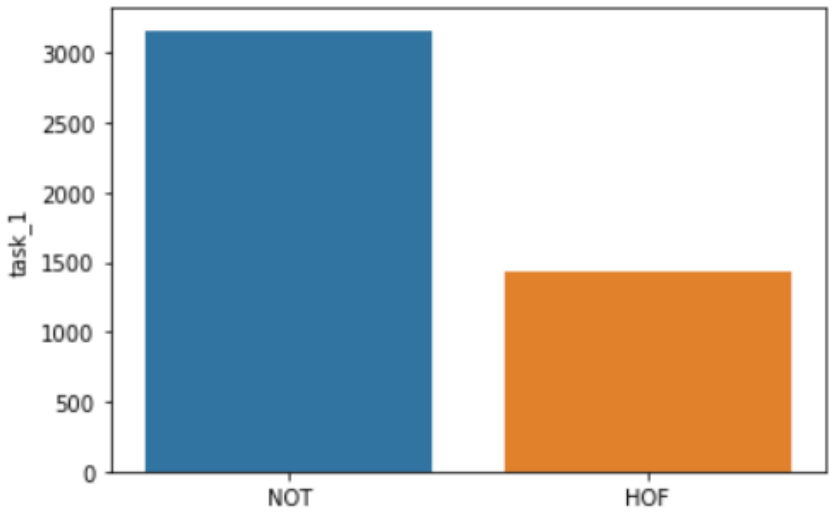

Figure 3: Class distribution for sub-task A

Similarly, when it came to the training dataset for subtask B, there were 3161 NONE, 654 OFFN, 566 HATE, and 213 PRFN tweets. Due to the unbalanced dataset, the models' performance may have been compromised. As a result of analyzing the link between one or more independent factors and a

dependent data variable, Logistic Regression was shown to have the greatest prediction performance. The goal is to estimate event probabilities, which includes establishing a link between variables and the likelihood of specific outcomes. It predicted 1166 tweets as NONE, 179 as OFFN, 136 as HATE, and 52 as PRFN. Coming in the second position was the CatBoost model with 1317 as NONE, 128 as OFFN, 45 as HATE, and 43 as PRFN. It has a Macro Precision of 0.5968 and an accuracy of 72.977%.

Because there were so few profane tweets in the training dataset, the classifiers performed poorly. If the classes are balanced, the classifiers and model will perform better for all class predictions. The root cause of poor performance with traditional machine learning models and evaluation metrics based on a balanced class distribution.

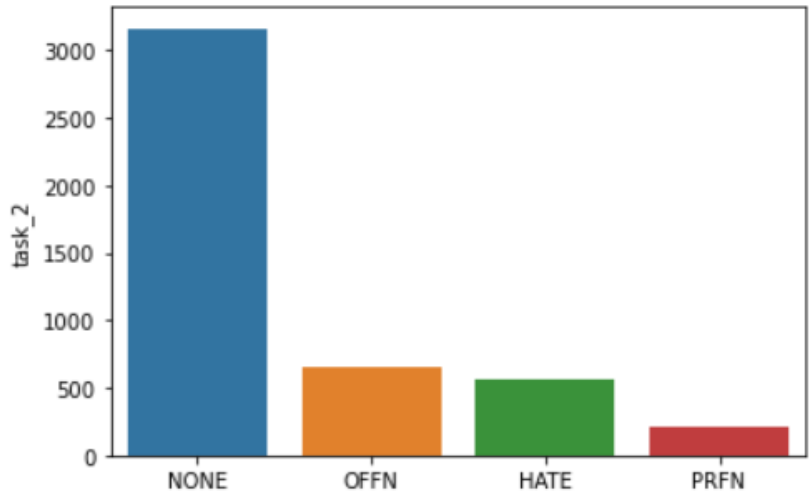

Figure 4: Class distribution for sub-task B

This supports the idea that Machine Learning parameters were inefficient when compared to Deep Learning techniques that attempted to learn from a limited quantity of data and a greater number of parameters.

In our observations, we found that Naïve Bayes, when used with TF-IDF (accuracy: 71.80%), showed poor performance compared to Naïve Bayes when used with Countvectorizer (accuracy: 78.068%). But traditionally, TF-IDF outperforms Count Vectorizers because it considers the frequency of words in the corpus and their importance.

One important remark is that, rather than deleting the entire hashtag with the phrases, it is preferable to delete the sign ”#” from the hashtag. These are made up of many words, each of which can be a valuable characteristic for the identification job on its own.

HASOC evaluation issues were frequently associated with the use of language registers such as youth chat, irony, or indirectness, which may have led to dataset mislabeling.

The boxplot of total system throughput for both sub-tasks reveal the Median of the tests is reasonably near to the best performance.

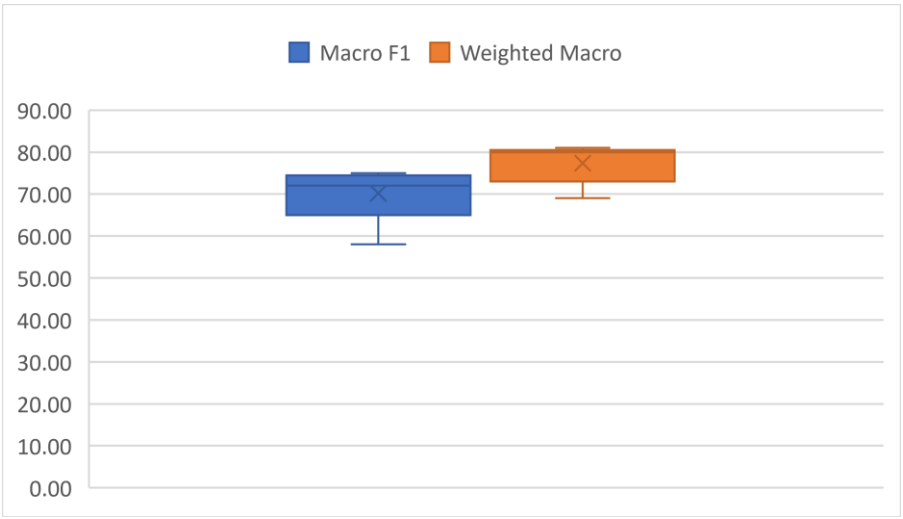

Figure 5: Boxplot of performance of all models for sub-task A

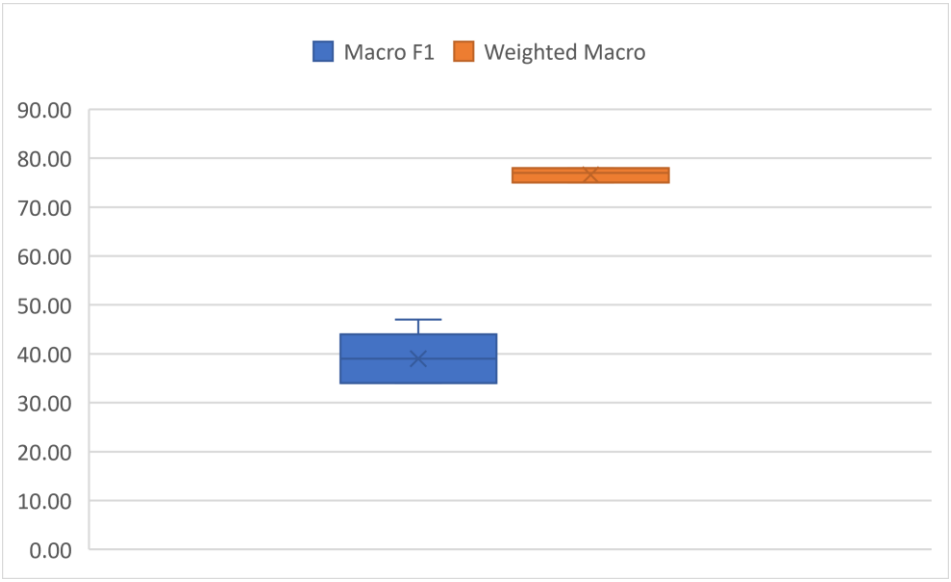

Figure 6: Boxplot of performance of all models for sub-task B

8. Conclusion

For text classification, our submissions to HASOC have shown that Machine Learning models are a great fit. According to the findings, the best way to classify hate speech is based on the language of the corpus, classification granularities, and distribution of each class label. The classification system's performance may decrease if the training dataset is unequal.

9. Future Work

In the future, we intend to include different languages and create a robust technology capable of dealing with multilingual data and transfer learning approaches capable of exploiting learning data across languages. Furthermore, we envision exploring deep learning models and using the transfer learning approach for better results.

10. Acknowledgements

Congratulations to all of the participants for their submissions and research effort. Thank you to the FIRE organizers for your help in getting this event together. We appreciate it.

11. References

[1] A. Matamoros-Fernández, J. Farkas, Racism, hate speech, and social media: A systematic review and critique, Television & New Media 22 (2021) 205–224.

[2] N. Vashistha, A. Zubiaga, Online multilingual hate speech detection: Experimenting with hindi and english social media, Information 12 (2021). URL: https://www.mdpi.com/2078-2489/12/1/5. doi:10.3390/info12010005.

[3] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, O. Frieder, Hate speech detection: Challenges and solutions, PloS one 14 (2019) e0221152.

[4] A. Schmidt, M. Wiegand, A survey on hate speech detection using natural language processing, in: Proceedings of the fifth international workshop on natural language processing for social media, 2017, pp. 1–10.

[5] S. Modha, T. Mandl, G. K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri, Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages and conversational hate speech, in: FIRE 2021: Forum for Information Retrieval Evaluation, Virtual Event, 13th-17th December 2021, ACM, 2021.

[6] S. Satapara, S. Modha, T. Mandl, H. Madhu, P. Majumder, Overview of the HASOC Subtrack at FIRE 2021: Conversational Hate Speech Detection in Code-mixed language , in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[7] R. Joshi, P. Goel, R. Joshi, Deep learning for hindi text classification: A comparison, in: International Conference on Intelligent Human Computer Interaction, Springer, 2019, pp. 94–101.

[8] D. Kakwani, A. Kunchukuttan, S. Golla, G. N.C., A. Bhattacharyya, M. M. Khapra, P. Kumar, IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages, in: Findings of EMNLP, 2020.

[9] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer, et al., Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.

[10] C. Ezeibe, Hate speech and election violence in nigeria, Journal of Asian and African Studies 56 (2021) 919–935.

[11] R. Joshi, R. Karnavat, K. Jirapure, R. Joshi, Evaluation of deep learning models for hostility detection in hindi text, in: 2021 6th International Conference for Convergence in Technology (I2CT), IEEE, 2021, pp. 1–5.

[12] A. Wani, I. Joshi, S. Khandve, V. Wagh, R. Joshi, Evaluating deep learning approaches for covid19 fake news detection, in: International Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation, Springer, 2021, pp. 153–163.

[13] R. Joshi, R. Joshi, Evaluating input representation for language identification in hindienglish code mixed text, arXiv preprint arXiv:2011.11263 (2020).

[14] P. Mishra, M. D. Tredici, H. Yannakoudakis, E. Shutova, Abusive language detection with graph convolutional networks, 2019. arXiv:1904.04073.

[15] Z. Zhang, L. Luo, Hate speech detection: A solved problem? the challenging case of long tail on twitter, 2018. arXiv:1803.03662.

[16] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schäfer, T. Ranasinghe, M. Zampieri, D. Nandini, A. K. Jaiswal, Overview of the HASOC subtrack at FIRE 2021: Hate speech and offensive content identification in English and Indo-Aryan languages, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021. URL: http://ceur-ws.org/.