Quartet@LT-EDI 2024: A Support Vector Machine Approach For Caste and Migration Hate Speech Detection

Shaun Allan H
Sri Sivasubramaniya Nadar College of Engineering
shaunallan2210716@ssn.edu.in

Samyuktaa Sivakumar
Sri Sivasubramaniya Nadar College of Enginerring
samyuktaa2210189@ssn.edu.in

Rohan R
Sri Sivasubramaniya Nadar College of Enginerring
rohan2210124@ssn.edu.in

Nikilesh Jayaguptha
Sri Sivasubramaniya Nadar College of Engineering
nikilesh2210219@ssn.edu.in

Durairaj Thenmozhi
Sri Sivasubramaniya Nadar College of Engineering
theni_d@ssn.edu.in

Abstract

Hate speech refers to the offensive remarks against a community or individual based on inherent characteristics. Hate speech against a community based on their caste and native are unfortunately prevalent in the society. Especially with social media platforms being a very popular tool for communication and sharing ideas, people post hate speech against caste or migrants on social medias. The Shared Task LT–EDI 2024: Caste and Migration Hate Speech Detection was created with the objective to create an automatic classification system that detects and classifies hate speech posted on social media targeting a community belonging to a particular caste and migrants. Datasets in Tamil language were provided along with the shared task. We experimented with several traditional models such as Naive Bayes, Support Vector Machine (SVM), Logistic Regression, Random Forest Classifier and Decision Tree Classifier out of which Support Vector Machine yielded the best results placing us 8th in the rank list released by the organizers.

1 Introduction

Hate is a very strong emotion or feeling of not liking someone or something. Hate expresses intense hostility towards others. Hate speech reflecting the same, refers to the offensive remarks or comments uttered by a person targeting a community or an individual person. Hate speeches are often uttered based on the target's inherent characteristics such as race, ethnicity, religion or gender.

In recent days, it can be said that social media platforms have enormously changed the way people communicate with one another (O'Keeffe et al., 2011). Social media can be seen as an immensely great tool for people to share their thoughts and ideas with the world. This has allowed the people to voice out their opinions broadening their freedom of speech. While this can benefit people a lot, it also comes with its own disadvantages.

Social media platforms are a place where individuals voice out their opinions, but there are also people who spread hate against a community or individual. We see hate speeches being posted in social media platforms very often (Mondal et al., 2017). Hate speeches targeting a particular community based on their caste and native place are prevalent in social media.

Hate speech inflicts immediate harm on its victims and also contributes to discrimination against the targeted community or individual. These sort of hate comments against a caste or migrants must be obliterated from the society. With social media platforms being an inevitable and popular tool in the modern society, it becomes imperative to moderate the hate speech posts. It is essential for a more positive and inclusive society.

Sentiment analysis, also known as opinion mining can be employed to detect and moderate the hate comments prevailing on social media platforms. Sentiment analysis is the process of determining the emotional tone that a digital text manifests (Taboada, 2016). Textual data can be analyzed and determined if the text expresses a positive, negative or neutral sentiment. Sentiment analysis is an immensely powerful tool to automate the process of detecting caste and migration hate speech on social media by analyzing the sentiment or emotion that the text manifests. Sentiment analysis can be carried out by supervised learning in case of availability to a well labelled and quality training data. In situations where one has no access to training data, unsupervised learning can also be utilized to perform sentiment analysis (Schouten et al., 2018).

The Shared Task LT–EDI 2024: Caste and Migration Hate Speech Detection was created with the motive to build an automation system that detects and classifies the text in Tamil language on social media platforms as caste and migration hate speech or not. Datasets containing text in Tamil language

Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion, pages 211–215

were provided along with the shared task.

This paper is organized as follows: Section 2 encompasses the related works as per the literature survey; Section 3 is entails information about the task and data; Section 4 pertains to the methodology used to build the classification system; Section 5 shows the results and analysis; Section 6 entails the conclusion; Sections 7 and 8 pertains to the limitaions of the model and the ethics statement respectively.

2 Related Works

Sentiment analysis is a field in which constant works and researches are being carried on. They have many applications on social and e-commerce platforms.

Rajput et al. (2021) proposed a hate speech detection classifier by replacing or integrating the word embeddings (fastText(FT), GloVe(GV) or FT + GV) with static word BERT embeddings. With extensive experimental traits it is observed that the performance of a neural network with static BERT embeddings is better than that wit FT, GV or FT + GV.

A large-scale analysis of multilingual hate speech in 9 languages from 16 different sources was conducted by Aluru et al. (2020). It was observed that in low resource setting, simple models such as LASER embedding with logistic regression performs the best, while in high resource setting BERT based models perform the best.

HateBERT, a re-trained BERT model was proposed by Caselli et al. (2021) for abusive language detection in English. The model was trained on RAL-E, a large-scale dataset of offensive Reddit comments in English.

Saha et al. (2018) built an automatic hate speech detection system against women by generating three types of features from the text: Sentence Embeddings, TF-IDF vectors and BOW vectors. These features were then concatenated and fed into a Logistic Regression model.

Rajalakshmi et al. (2023) experiments several machine learning models to classify hate speech in Tamil texts. Several models including BERT, XLM-RoBERTa, IndicBERT, mBERT, TaMillion and MuRIL were experimented. It was observed that the highest performance was achieved by a combination of stemming the text data, embedding it with MuRIL and using a majority voting ensemble as the downstream classifier. Alatawi et al. (2021) investigates the feasibility of leveraging domain-specific word embedding in Bidirectional LSTM based deep model to automatically detect/classify hate speech. Furthermore, the use of the transfer learning language model (BERT) on hate speech problem as a binary classification task was investigated.

3 Task and Data Description

The objective of the Shared Task LT–EDI 2024: Caste and Migration Hate Speech Detection[1] (Rajiakodi et al., 2024) is to create an automatic classification system that detects and classifies whether a text is caste and migration hate speech or not. Training and development datasets were provided in Tamil language. The dataset encompassed two fields: text and label. The training dataset had a total of 5,355 records out of which 2,052 were labelled 1 representing caste and migration hate speech and 3,303 were labelled 0 representing non caste and migration hate speech.

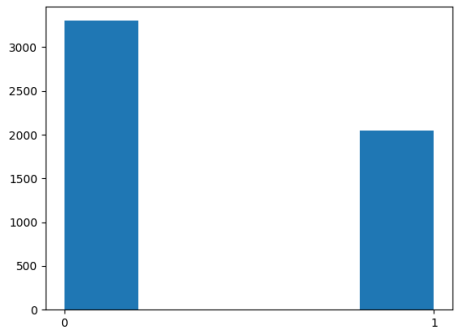

Figure 1: Data Distribution in Training Dataset

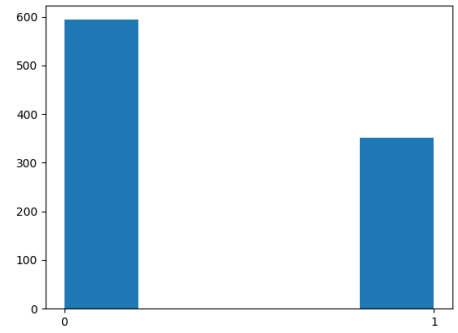

Figure 2: Data Distribution in Development Dataset

[1] https://codalab.lisn.upsaclay.fr/competitions/16089

The development dataset had a total of 945 records out of which 351 were labelled as hate speech and 594 were labelled as non-hate speech.

4 Methodology

4.1 Data Preprocessing

The given textual cannot be directly fed to the machine learning model. Data must be well processed and cleansed in order to yield better results.

1. The given textual data consisted of emoticons and punctuations which don't add any meaning to the text thus contributing nothing to the classification process. Therefore, it is important to remove these emoticons and punctuations before any further process.

2. As most of the embedding systems available work better on English on text than regional languages, the given text which is in Tamil is translated to English using googleTrans[2] library. Translating the text to English increases the accuracy of our classification system.

3. Stop words are redundant words present in a text that don't contribute any emotion for sentiment analysis. These stop words are eliminated from the given text using the NLTK[3] library. Removing these stop words decreases the dataset size and hence the training time of the model also decreases.

4.2 Feature Extraction

Feature extraction is the process of converting raw digital text into vectors containing numerical inputs. As machine learning models cannot work on textual data, texts have to be converted into numerical vectors suitable for the model to work with.

We have employed Term Frequency–Inverse Document Frequency (TF–IDF) vectorizer from the scikit learn library to extract features from the translated English text. Term Frequency refers to the frequency of a term appearing in a particular document while Inverse Document Frequency refers to the measure of how common a term is in the entire corpus of documents. TF-IDF value of a term is defined as the product of its Term Frequency and Inverse Document Frequency.

4.3 Classification using ML Models

We employed several traditional models such as Logistic Regression, Support Vector Machine (SVM), Random Forest Classifier, Decision Tree Classifier

[2] https://pypi.org/project/googletrans/

[3] https://www.nltk.org/

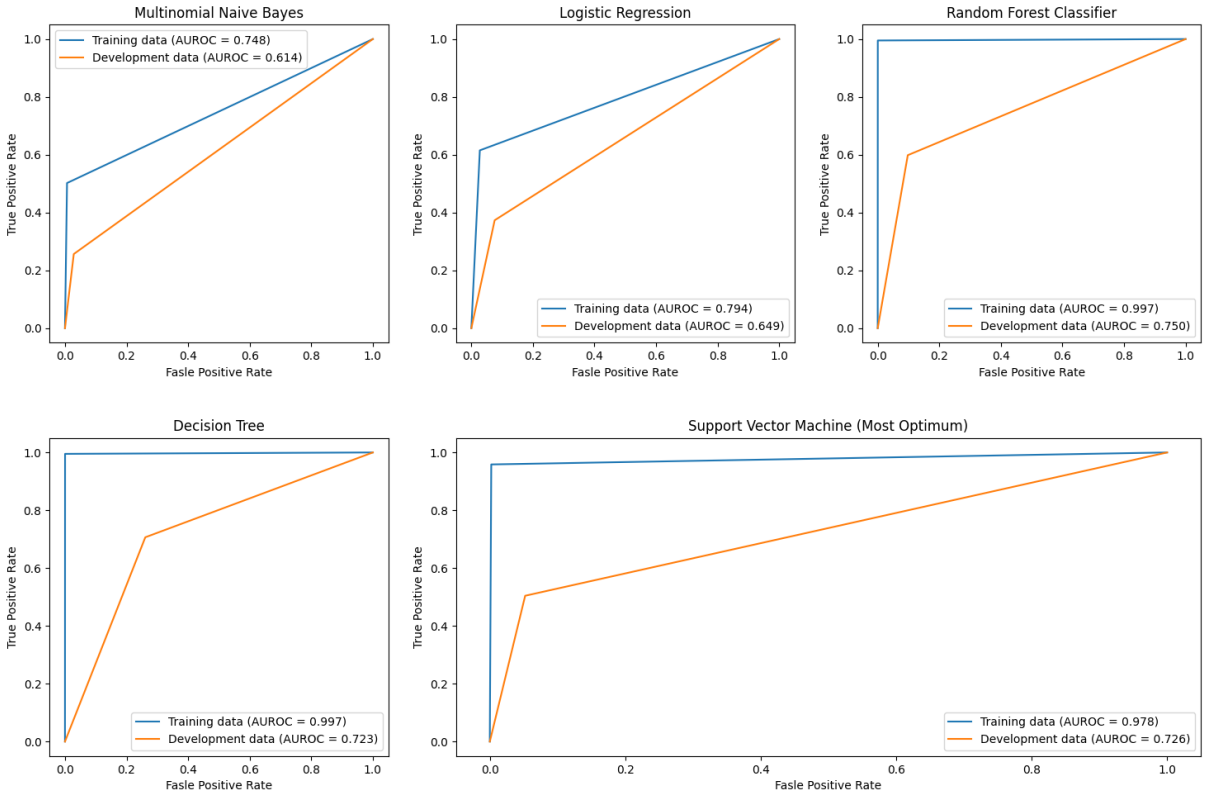

Figure 3: Comparison and Analysis of ROC Curves and AUROC scores

Metric	Logistic Regression	Support Vector Machine	Random Forest Classifier	Decision Tree Classifier	Naive Bayes
Accuracy	0.72	0.78	0.78	0.73	0.70
Macro Average F1 score	0.65	0.74	0.75	0.71	0.60

Table 1: Comparison of metrics on Development Data

	Precision	Recall	F1-score	Support
Non Caste and Migration Hate Speech	0.76	0.93	0.84	973
Caste and Migration Hate Speech	0.82	0.52	0.64	602
Accuracy			0.77	1,575
Macro Avg	0.79	0.75	0.74	1,575
Weighted Avg	0.78	0.77	0.76	1,575

Table 2: Classification Report for SVM on Test Data

and Naive Bayes on the extracted numerical features. After evaluating the metrics of all the models, Support Vector Machine yielded the highest accuracy and macro average of F1 score. Support Vector Machine (SVM) is one of the most popular supervised machine learning algorithms widely used for classification tasks as well as regression tasks. SVM works on finding the best hyperplane that separates data points of different classes in a feature space.

5 Result and Analysis

The performance of various traditional models including Naive Bayes, Support Vector Machine (SVM), Random Forest Classifier, Decision Tree Classifier and Logistic Regression were evaluated and compared. The Receiver Operating Characteristic (ROC) curve was plotted and the Area Under Receiver Operating Characteristic (AUROC) was calculated for all the models. The ROC curve is defined as the curve that is plotted against the True Positive Rate and False Positive Rate of the predictions obtained from a model at varying threshold levels. The ROC curve is a very useful visual representation to analyze and compare the performance of classification models.

On evaluating the metrics of the models, it is found that Support Vector Machine (SVM) produced the best numbers on both training data and development data. It is to be noted that though Random Forest Classifier performed very slightly better than SVM on unseen development data, with the macro average score on training data being 1.00, the model is considered overfitted.

On evaluating with the test data given by the organizers, the SVM model yielded a macro average F1 score of 0.74. We were ranked 8th in the rank list released by the organizers.

6 Conclusion

By means of this paper, we experimented several traditional machine learning models on the features extracted by the TF-IDF vectorizer. The metrics and ROC curves of each model were plotted and analysed to effectively compare the performance of the models. It was observed that out of all models, Support Vector Machine (SVM) gave the best metrics and ROC curve. While this model has good performance over the other models, it is to be noted that better results can be obtained by utilizing neural networks and more complex embedding systems.

7 Limitations

Thought the TF-IDF vectorizer which was used to extract the features from was digital text performs well in most cases, comes with its own inherent limitations. The TF-IDF vectorizer makes no use of the semantic relations between words for feature extraction. Also, feature extraction can be slow when handling with large vocabularies because it

computes document similarity directly in the word-count space.

As the model is built with SVM algorithm, when trained with immensely large datasets, the SVM model fails to perform well and also consume a lot of time and memory for training. The final model is difficult to understand an interpret as a result of which small calibrations cannot be done to the model. Also, a probabilistic interpretation of the result cannot be produced as the SVM algorithm is incapable of producing such probabilistic results.

8 Ethics Statement

We ensured that the ACL Code of Ethics[4] was practiced throughout the process of working on the Shared Task. The main notion behind building the classification system is to make social media platforms a safe and inclusive environment for all community of people to thrive and exist by detecting and moderating caste and migration hate speech in social media platforms. Credits have been given to all authors whose existing works and ideas has been referenced or utilized in References section. Data privacy is a priority in our solution as it does not provide any access on data to random individuals or organizations ensuring no leak of information.

The given task was used as an opportunity to upgrade and enhance our skills while practicing the principles of professional competence. The proposed solution abides by the local, regional, national and international laws and regulations.

References

Hind S Alatawi, Areej Alhothali, and Kawthar Moria. 2021. Detection of hate speech using BERT and hate speech word embedding with deep model. *ArXiv, abs/2111.01515*.

Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models for multilingual hate speech detection.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for Abusive Language Detection in English.

Mainack Mondal, Leandro Araújo Silva, and Fabrício Benevenuto. 2017. A measurement study of hate speech in social media. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, HT '17, page 85–94, New York, NY, USA. Association for Computing Machinery.

Gwenn Schurgin O'Keeffe, Kathleen Clarke-Pearson, et al. 2011. The impact of social media on children, adolescents, and families. *Pediatrics*, 127(4):800–804.

Ratnavel Rajalakshmi, Srivarshan Selvaraj, Faerie Mattins R., Pavitra Vasudevan, and Anand Kumar M. 2023. HOTTEST: Hate and Offensive content identification in Tamil using Transformers and Enhanced Stemming. *Computer Speech Language*, 78:101464.

Saranya Rajiakodi, Bharathi Raja Chakravarthi, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Sathiyaraj Thangasamy, Bhuvaneswari S, and Charmathi Rajkumar. 2024. Overview of Shared Task on Caste and Migration Hate Speech Detection. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity and Inclusion*, Malta. European Chapter of the Association for Computational Linguistics.

Gaurav Rajput, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. 2021. *Hate Speech Detection Using Static BERT Embeddings*, page 67–77. Springer International Publishing.

Punyajoy Saha, Binny Mathew, Pawan Goyal, and Animesh Mukherjee. 2018. Hateminers : Detecting hate speech against women.

Kim Schouten, Onne van der Weijde, Flavius Frasincar, and Rommert Dekker. 2018. Supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data. *IEEE Transactions on Cybernetics*, 48(4):1263–1275.

Maite Taboada. 2016. Sentiment analysis: An overview from linguistics. *Annual Review of Linguistics*, 2(1):325–347.

[4] https://www.aclweb.org/portal/content/acl-code-ethics