Supernova@DravidianLangTech2023 @Abusive Comment Detection in Tamil and Telugu - (Tamil, Tamil-English, Telugu-English)

A Ankitha Reddy
SSN College of Engineering
ankithareddy2210178@ssn.edu.in

Ann Maria Thomas
SSN College of Engineering
annthomas2210391@ssn.edu.in

Pranav Moorthi
SSN College of Engineering
pranav2210176@ssn.edu.in

Durairaj Thenmozhi
SSN College of Engineering
theni_d@ssn.edu.in

B. Bharathi
SSN College of Engineering
bharathib@ssn.edu.in

Gayathri G L
SSN College of Engineering
gayathri2010090@ssn.edu.in

Krithika S
SSN College of Engineering
krithika2010039@ssn.edu.in

Abstract

This paper presents our submission for Abusive Comment Detection in Tamil and Telugu - DravidianLangTech 2023 (Tamil, Tamil-English, Telugu-English. The aim is to classify whether a given comment is abusive or not. Support Vector Machines (SVM), Logistic Regression and Linear SVC Classifiers paired with Term Frequency–Inverse Document Frequency feature extraction were used and contrasted to make the classification models. The lack of annotated and balanced datasets for low-resource languages has also been acknowledged.

1 Introduction

Those statements that harbour ill feelings towards a person or a group of people are categorised as abusive comments. These comments consist of either profanity or racist, sexist, xenophobic, homophobic or transphobic connotations targeting members belonging to certain communities (Balouchzahi et al., 2022).

Identification of abuse over online social networks has proven to be a tedious task due to the overwhelming volume of content generated through social media (Ravikiran et al., 2022). These platforms offer a wide reach and the provision of anonymity can empower individuals to partake in hate speech, as they perceive themselves to be protected from facing immediate consequences for their actions.

A large portion of the users on social media platforms engage with other users in their respective native languages and most abusive content detection models are not trained to handle the diversity that exists in these numerous regional languages. The task of identifying abusive comments within the Tamil and Telugu languages (Priyadharshini et al., 2023b) is notably intricate and complex owing to the lack of linguistically tailored resources. This complexity stems from the scarcity of well-annotated datasets (Priyadharshini et al., 2022, 2023a) and proficiently trained models specific to these languages. The distinctive linguistic structures and contextual intricacies inherent in Tamil and Telugu pose obstacles to the creation of precise algorithms for detecting abusive language.

Abusive comments can lead to a hostile online environment, discouraging users from engaging in discussions or expressing themselves freely. Abusive comment detection in these languages allows for targeted content moderation that aligns with the linguistic and cultural context.

Overall, abusive comment detection in social media texts is essential for promoting user safety and enhancing user experience while fostering inclusivity and sustaining a respectful online environment.

2 Related Work

"A Comparison of Classical Versus Deep Learning Techniques for Abusive Content Detection on Social Media Sites" (Chen et al., 2018) addressed the fact that classifiers such as support vector machines (SVM), combined with bag of words or n gram feature representation, have traditionally dominated in text classification for decades. However, in the recent past, concepts under the domain of deep neural networks have begun gaining traction. They explored the impact of numerous levels of training set imbalances on different classifiers. In comparison, it was revealed that deep learning models (CNNs and KNNs) outperformed the traditional SVM classifier when the associated training dataset is seriously imbalanced. However, it was inferred that the performance of the SVM classifier could be dramatically improved through the method of oversampling, surpassing the deep learning models.

Though much work has been done to identify offensive content in major languages such as English (Chakravarthi et al., 2021), it is an arduous

Third Workshop on Speech and Language Technologies for Dravidian Languages,
pages 225–230, Varna, Bulgaria, Sep 7, 2023.
https://doi.org/10.26615/978-954-452-085-4_032

task to identify and flag offensive and abusive content in low-resource languages, in the scope of our study, Dravidian languages, due to scarcity and unavailability of annotated datasets (Khan et al., 2021). Due to the predominance of the English script, the datasets involve multiple data points incorporating elements of code-switching or code-mixing (Chakravarthi et al., 2021; Ashraf et al., 2022; Shanmugavadivel et al., 2022). However (Akhter et al., 2021),attempted to detect the same in Urdu and Roman Urdu using, analysing and comparing five diverse ML models (SVM, NB, Logistic, IBK and JRip) and four DL models (BLSTM, CLSTM, LSTM and CNN). It was found that the convolutional neural network outperforms the other models and achieves 96.2 and 91.4 percentage accuracy on Urdu and Roman Urdu. The results also revealed that the one-layer architectures of deep learning models give better results than two-layer architectures.

More relevant to our cause, (Sazzed, 2021) annotated a Bengali corpus of 3000 transliterated Bengali comments and found that support vector machine (SVM) shows the highest efficacy for identifying abusive content. However, it is important to note that the dataset created was unbiased which may potentially be the cause for the outperformance of SVM. The paper delves deeper into the ubiquity of transliterated Bengali comments in social media as it renders monolingual approaches futile. It also addresses the issue of the lack of publicly available data for such low-resource languages. Other notable contributions allied with the scope of our study include (Kannan et al., 2014; Daumé III, 2004) pre-processing, SVM), each providing an extensive analysis of the pre-processing phase and the usage of SVM classifiers respectively, highlighting their merits and efficacy.

3 Dataset Analysis

The task has been furcated into three subdivisions based on the language of choice, namely Tamil-English, Tamil and Telugu-English. The target variables of the given datasets have been described below. The provided labels for the Tamil-English and Tamil tasks include None-of-the-above, Transphobic, Counter-speech, Misandry, Homophobia, Hope-Speech, Xenophobia and Misogyny while the Telugu-English includes hate and non-hate labels. The data distribution of each dataset is provided below.

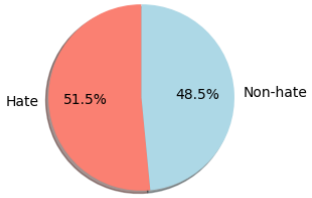

Figure 1: Data distribution of Telugu-English

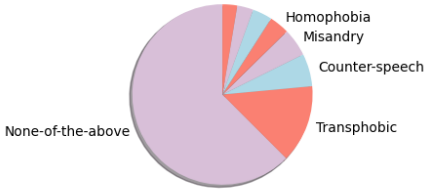

Figure 2: Data distribution of Tamil-English

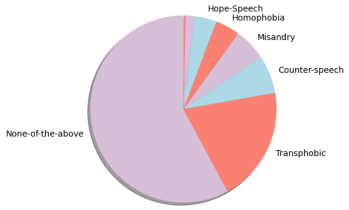

Figure 3: Data distribution of Tamil

Category	Telugu
Hate	1939
Non-Hate	2061

Table 1: Data distribution of Telugu

Category	Tamil	Tamil-English
None of the above	1296	3715
Misandry	446	830
Counter-Speech	149	348
Misogyny	125	211
Xenophobia	95	297
Hopespeech	86	213
Homophobia	35	172
Transphobic	6	157
Not Tamil	2	

Table 2: Data distribution of Tamil and Tamil-English

Analysis of the data distribution provides insights into the class imbalances which could potentially hinder the performance of the models without

the appropriate measures in place.

The definition and usage of stop-words is crucial for the effectiveness of such code-mixed and code-switched datasets. While stop-words lists for languages such as English and Spanish have been implemented in the nltk.corpus library, the manual creation of stop-words lists for certain low-resource languages was necessitated. This was executed with the usage of previous domain and linguistic knowledge as well as online resources.

Tamil_stopwords = [ஒரு, என்று, மற்றும், இந்த, இது, என்ற, கொண்டு, என்பது, பல, ஆகும், அல்லது, அவர், நான், உள்ள, அந்த, இவர், என, முதல், என்ன, இருந்து, சில, என், போன்ற, வேண்டும், வந்து, இதன், அது, அவன், தான், பலரும், என்னும், மேலும், பின்னர், கொண்ட, இருக்கும், தனது, உள்ளது, போது, என்றும், அதன், தன், பிறகு, அவர்கள், வரை, அவள், நீ, ஆகிய, இருந்தது, உள்ளன, வந்த, இருந்த, மிகவும், இங்கு, மீது, ஓர், இவை, இந்தக், பற்றி, வரும், வேறு, இரு, இதில், போல், இப்போது, அவரது, மட்டும், இந்தப், எனும், மேல், பின், சேர்ந்த, ஆகியோர், எனக்கு, இன்னும், அந்தப், அன்று, ஒரே, மிக, அங்கு, பல்வேறு, விட்டு, பெரும், அதை, பற்றிய, உன், அதிக, அந்தக், பேர், இதனால், அவை, அதே, ஏன், முறை, யார், என்பதை, எல்லாம், மட்டுமே, இங்கே, அங்கே, இடம், இடத்தில், அதில், நாம், அதற்கு, எனவே, பிற, சிறு, மற்ற, விட, எந்த, எனவும், எனப்படும், எனினும், அடுத்த, இதனை, இதை, கொள்ள, இந்தத், இதற்கு, அதனால், தவிர, போல, வரையில், சற்று, எனக்]

4 Methodology

4.1 Preprocessing

Preprocessing of data is done to improve the efficiency of the model. The performance metrics of a model could vary drastically with efficient data preprocessing. The different steps involved in preprocessing of data are listed below.

1. Text Normalisation: By expanding contractions, and converting all the characters to lowercase, the text becomes more uniform and easier to analyse.

2. Removal of special characters, symbols and emojis: Special characters such as punctuation marks and emoticons do not donate any meaning to the text. Removal of these characters aids the machine learning model as it reduces the volume of text the model has to sort through.

3. Removal of stop words: Stop words refer to frequently occurring words that lack substantial semantic meaning or contribute minimally to the holistic comprehension of a given text. By eliminating these words, the data payload is reduced, resulting in expedited processing durations and enhanced computational efficacy.

4. Stemming of data: Stemming seeks to minimise words to their morphological base or root form. By stemming words, occurrences of related words are combined, giving a more accurate representation of their true frequency. This process is important for tasks such as sentiment analysis. Through vocabulary size reduction, stemming facilitates expedited model training and diminished memory demands for Natural Language Processing systems.

4.2 TF-IDF feature extraction

TF–IDF or Term Frequency–Inverse Document Frequency is a methodology used to create features from text data. It is a statistical measure of how important a word is in a collection of text or document.

$$TF = \frac{number\ of\ times\ term\ occurs\ in\ document}{total\ number\ of\ terms\ in\ document}$$

$$IDF = log\left(\frac{number\ of\ documents\ in\ corpus}{number\ of\ documents\ in\ corpus\ that\ contain\ the\ term}\right)$$

Words exclusive to a small proportion of documents receive higher importance than words recurring in all documents (e.g., a, the, and). TF-IDF vectorizer matches each feature to a corresponding numerical feature that is calculated from its TF-IDF score. The term frequency and inverse document frequency are multiplied to obtain the score. The term will have a higher TF-IDF score based on its relevance.

For this task, we used the TF-IDF to vectorize the preprocessed data into a classification model as it allows the conversion of unstructured text data into structured numerical representations that natural language processing models can work with. This representation enables the model to identify meaningful patterns and relationships in the text, facilitating sentiment analysis.

4.3 SVM Classifier

Support Vector Machine (SVM) is a supervised learning algorithm used to classify text data into different categories based on the features extracted from the text. SVM uses linear functions in a high dimensional feature space to categorise data using statistical learning theory (Cristianini and Shawe-Taylor, 2000; Daumé III, 2004). By drawing a hyperplane to segregate the classes in an n-dimensional space, it plots the data points as support vectors.

The TF-IDF feature vectors of the training samples are fed into the SVM algorithm. The SVM algorithm learns to find an optimal decision boundary that separates the feature vectors of different classes. Then, the algorithm classifies unlabeled text samples based on the patterns it discovered during the training phase.

4.4 Logistic Regression

Logistic regression establishes a relationship between independent variables and a categorical response or outcome variable by approximating the likelihood that the outcome belongs to a particular class. The regression model serves two objectives: (1) It aids in estimating the outcome variable when faced with new sets of predictive variable values (2) It is instrumental in providing insights into queries related to the subject under investigation. This is achieved through the utilisation of coefficients assigned to each predictive variable, which offer a clear understanding of the extent of each variable's contribution to the final result. (Vimal and Kumar, 2020)

The logistic regression function effectively converts any input values into a numeric range spanning from 0 to 1. The mathematical transformation executed by the logistic function serves to convert the initial linear combination into a reliable probability estimation.

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

4.5 Linear SVC

Linear SVC creates a hyperplane using a linear kernel function to classify the different data points. The data points are grouped into classes with common features. Maximising the margin width between the hyperplanes results in better classification. The support vectors drawn from each point to lines of separation are used to mathematically compute the goal function.

5 Results and Analysis

The evaluation of the task is done based on the following performance metrics: Precision, Recall and F1- score. Recall measures the classifier's ability to identify positive instances correctly while precision is a measure of how accurate the positive predictions are.

$$Recall = \frac{TP}{TP+FN} \quad Precision = \frac{TP}{TP+FP}$$

F1 score provides a harmonised assessment of a model's performance when both precision and recall are important.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

	precision	recall	f1-score	support
hate	0.70	0.78	0.74	465
non-hate	0.79	0.71	0.75	535
accuracy			0.74	1000
macro avg	0.75	0.75	0.74	1000
weighted avg	0.75	0.74	0.74	1000

Figure 4: Classification report of SVM on Telugu dataset

	precision	recall	f1-score	support
hate	0.70	0.77	0.73	465
non-hate	0.78	0.71	0.74	535
accuracy			0.74	1000
macro avg	0.74	0.74	0.74	1000
weighted avg	0.74	0.74	0.74	1000

Figure 5: Classification report of Logistic Regression on Telugu dataset

	precision	recall	f1-score	support
hate	0.69	0.77	0.73	465
non-hate	0.78	0.70	0.74	535
accuracy			0.73	1000
macro avg	0.73	0.73	0.73	1000
weighted avg	0.74	0.73	0.73	1000

Figure 6: Classification report of Linear SVC on Telugu dataset

On exploration of the plethora of prospective models, the weighted average and macro average F1 scores of the models implemented for the Telugu dataset were as follows: Logistic Regression (0.74), SVM (0.74) and Linear SVC (0.73) (figures 6, 7 and 8). SVM and Logistic Regression had

similar results, which outperformed Linear SVC marginally. Nonetheless, in comparison to Logistic Regression, the precision, recall and class-wise performance metrics of SVM were superior, corroborating the initial hypothesis on the class imbalances in the dataset.

	precision	recall	f1-score	support
0	0.69	0.99	0.81	917
1	0.00	0.00	0.00	40
2	0.00	0.00	0.00	95
3	0.78	0.50	0.61	218
4	1.00	0.02	0.05	43
5	1.00	0.02	0.04	53
6	1.00	0.40	0.57	70
7	1.00	0.02	0.04	50
accuracy			0.70	1486
macro avg	0.68	0.24	0.26	1486
weighted avg	0.69	0.70	0.62	1486

Figure 7: Classification report of SVM on Tanglish dataset

	precision	recall	f1-score	support
0	0.71	0.97	0.82	917
1	0.00	0.00	0.00	40
2	0.58	0.12	0.19	95
3	0.72	0.56	0.63	218
4	1.00	0.07	0.13	43
5	0.17	0.02	0.03	53
6	1.00	0.43	0.60	70
7	1.00	0.04	0.08	50
accuracy			0.71	1486
macro avg	0.65	0.28	0.31	1486
weighted avg	0.70	0.71	0.65	1486

Figure 8: Classification report of Logistic regression on Tanglish dataset

	precision	recall	f1-score	support
0	0.77	0.93	0.84	917
1	0.38	0.07	0.12	40
2	0.51	0.27	0.36	95
3	0.64	0.64	0.64	218
4	0.60	0.21	0.31	43
5	0.31	0.09	0.14	53
6	0.90	0.54	0.68	70
7	0.57	0.24	0.34	50
accuracy			0.73	1486
macro avg	0.59	0.38	0.43	1486
weighted avg	0.70	0.73	0.70	1486

Figure 9: Classification report of Linear SVC on Tanglish dataset

However, a point of interest is the superiority of Linear SVC over Logistic Regression and SVM classifiers on the Tamil and Tamil-English datasets. The classes in Linear SVC are separable by a linear hyperplane as opposed to SVM wherein kernel functions are employed to convert the non linear spaces to linear spaces by transforming data into a higher dimension. Hence, the better performance of the SVC classifier could be attributed to the linear separability of the Tamil and Tamil-English datasets since it minimises the probability of inaccurate classifications.

Furthermore, a deeper analysis of the classification reports substantiated the hypothesis regarding the impact of class imbalances on the model's performance. Evident from the aforementioned performance metrics, though the accuracy of the classifiers are similar for all the datasets utilised, the classifiers proved to perform significantly better with regard to the macro and weighted F1-scores on the Telugu dataset than the Tamil and Tamil-English datasets due to the lack of parity in the latter. Specifically on analysing the label-wise metrics, a predominant inference is the inability of the SVM, Logistic Regression and Linear SVC classifiers to generalise on the test data with a smaller quantity of data points for each label.

6 Conclusion

Our paper describes the models implemented that detect abusive comments in Tamil and Telugu. The objective was to classify comments as abusive or non-abusive. Data preprocessing was undertaken to ensure uniformity and three models were implemented along with Term Frequency–Inverse Document Frequency feature extraction. Support Vector Machines (SVM) Classifiers, Logistic regression and Linear SVC were utilised to build our classification models.

The SVM model performed the best on the Telugu dataset with a macro average and weighted F1-score of 0.74, while the Linear SVC model proved to perform better on the Tamil and Tamil-English datasets due to the relatively linear nature of the datasets utilised. This discrepancy is quite apparent particularly with regard to the macro average F1-score.

From a rudimentary perspective, the issues in dealing with datasets involving low-resource languages were acknowledged and rectified by appropriate measures such as the creation of stop words lists for such languages. Another cardinal drawback of the methods explored is due to the class imbalances. In future works, this could potentially be remedied by implementing clustering methods, bootstrapping or data enrichment techniques.

References

Muhammad Pervez Akhter, Zheng Jiangbin, Irfan Raza Naqvi, Mohammed AbdelMajeed, and Tehseen Zia. 2021. Abusive language detection from social media comments using conventional machine learning and deep learning approaches. *Multimedia Systems*, pages 1–16.

Nsrin Ashraf, Mohamed Taha, Ahmed Abd Elfattah, and Hamada Nayel. 2022. Nayel@ lt-edi-acl2022: Homophobia/transphobia detection for equality, diversity, and inclusion using svm. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 287–290.

Fazlourrahman Balouchzahi, Sabur Butt, Grigori Sidorov, and Alexander Gelbukh. 2022. Cic@ lt-edi-acl2022: Are transformers the only hope? hope speech detection for spanish and english comments. In *Proceedings of the second workshop on language technology for equality, diversity and inclusion*, pages 206–211.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Sajeetha Thavareesan, Dhivya Chinnappa, Durairaj Thenmozhi, Elizabeth Sherly, John P McCrae, Adeep Hande, Rahul Ponnusamy, Shubhanker Banerjee, et al. 2021. Findings of the sentiment analysis of dravidian languages in code-mixed text. *arXiv preprint arXiv:2111.09811*.

Hao Chen, Susan McKeever, and Sarah Jane Delany. 2018. A comparison of classical versus deep learning techniques for abusive content detection on social media sites. In *Social Informatics: 10th International Conference, SocInfo 2018, St. Petersburg, Russia, September 25-28, 2018, Proceedings, Part I 10*, pages 117–133. Springer.

Nello Cristianini and John Shawe-Taylor. 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Hal Daumé III. 2004. Support vector machines for natural language processing. *Lecture Notes*.

Subbu Kannan, Vairaprakash Gurusamy, S Vijayarani, J Ilamathi, Ms Nithya, S Kannan, and V Gurusamy. 2014. Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks*, 5(1):7–16.

Lal Khan, Ammar Amjad, Noman Ashraf, Hsien-Tsung Chang, and Alexander Gelbukh. 2021. Urdu sentiment analysis with deep learning methods. *IEEE Access*, 9:97803–97812.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Subalalitha Chinnaudayar Navaneethakrishnan, Malliga Subramanian, Kogilavani Shanmugavadivel, Premjith B, Abirami Murugappan, Sai Prashanth Karnati, Rishith, Chandu Janakiram, and Prasanna Kumar Kumaresan. 2023a. Findings of the shared task on Abusive Comment Detection in Tamil and Telugu. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages DravidianLangTech 2023*. Recent Advances in Natural Language Processing.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Subalalitha Cn, Thenmozhi Durairaj, Malliga Subramanian, Kogilavani Shanmugavadivel, Siddhanth U Hegde, and Prasanna Kumaresan. 2022. Overview of abusive comment detection in Tamil-ACL 2022. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 292–298, Dublin, Ireland. Association for Computational Linguistics.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Malliga S, SUBALALITHA CN, Kogilavani S V, Premjith B, Abirami Murugappan, and Prasanna Kumar Kumaresan. 2023b. Overview of shared-task on abusive comment detection in tamil and telugu. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, Varna, Bulgaria. Recent Advances in Natural Language Processing.

Manikandan Ravikiran, Bharathi Raja Chakravarthi, Anand Kumar Madasamy, Sangeetha Sivanesan, Ratnavel Rajalakshmi, Sajeetha Thavareesan, Rahul Ponnusamy, and Shankar Mahadevan. 2022. Findings of the shared task on offensive span identification from code-mixed tamil-english comments. *arXiv preprint arXiv:2205.06118*.

Salim Sazzed. 2021. Abusive content detection in transliterated bengali-english social media corpus. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 125–130.

Kogilavani Shanmugavadivel, Sai Haritha Sampath, Pramod Nandhakumar, Prasath Mahalingam, Malliga Subramanian, Prasanna Kumar Kumaresan, and Ruba Priyadharshini. 2022. An analysis of machine learning models for sentiment analysis of tamil code-mixed data. *Computer Speech & Language*, 76:101407.

Bhartendoo Vimal and S Anupama Kumar. 2020. Application of logistic regression in natural language processing. *Int J Eng Res*, 9.