DETECTION OF OFFENSIVE COMMENTS IN TAMIL: SPARSE ATTENTION MECHANISM AND ENSEMBLING STRATEGIES ON TRANSFORMERS

A FINAL YEAR PROJECT ABSTRACT

Submitted by

DHINAKARAN V S (2019103517)

NAVEEN G (2019103040)

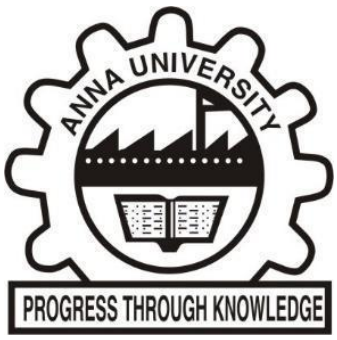
PRAVEEN V M (2019103559)

TEAM NUMBER - 22

GUIDED BY – Dr. P.VELVIZHY

for the course

CS 6811 – PROJECT WORK



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

DECEMBER 2022

CONTENTS

INTRODUCTION

People all over the world using social media to share information and communicate. Usage of this social media platform may have parallel negative impact on people's well-being. These hateful and offensive comments are spread by the toxic users. If such toxic behaviour is not addressed in a timely manner, it can have a cascading effect and discourages other people from being involved in online community. Offensive language detection is not available for low resource languages like Tamil. So, our system uses deep learning techniques and transformer models to identify offensive Tamil comments.

OVERALL OBJECTIVES

The main objective is to identify the offensive content in Tamil posted on social media platforms, YouTube comments and so on. Use multilingual transformer models which performs cross-lingual transfer learning to identify offensive language of code-mixed text in Tamil language. We use sparse attention mechanism to increase the performance of the transformer models and ensemble of transformer models to give the final optimized model.

LITERATURE SURVEY

The project aims to develop an offensive language detection in Tamil using deep learning techniques and transformer models. Many research papers have dealt with the topic of offensive language detection.

[1]Gaydhani, A., Doma, V., Kendre, S., Bhagwat, L., 2018. Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. An approach to automatically classify tweets on Twitter into three classes: hateful, offensive and clean. Require a well- defined feature extraction strategy.

[2]Dave, B., Bhat, S., Majumder, P., 2021. Irnlp_daiict @ dravidianlangtech-eacl2021: offensive language identification in Dravidian languages using TF-IDF char.n-grams and MuRIL. In: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages.Classified the YouTube comments in Tamil, Malayalam, and Kannada into five classes using LR and linear SVM models. It uses traditional ML classifiers which works best in English alone.

[3]Dowlagar, S., Mamidi, R., 2021. Hasocone@ fire-hasoc2020: Using BERT and multilingual BERT models for hate speech detection. An approach to automatically classify hate speech and offensive content. Number of parameters to train is very large.

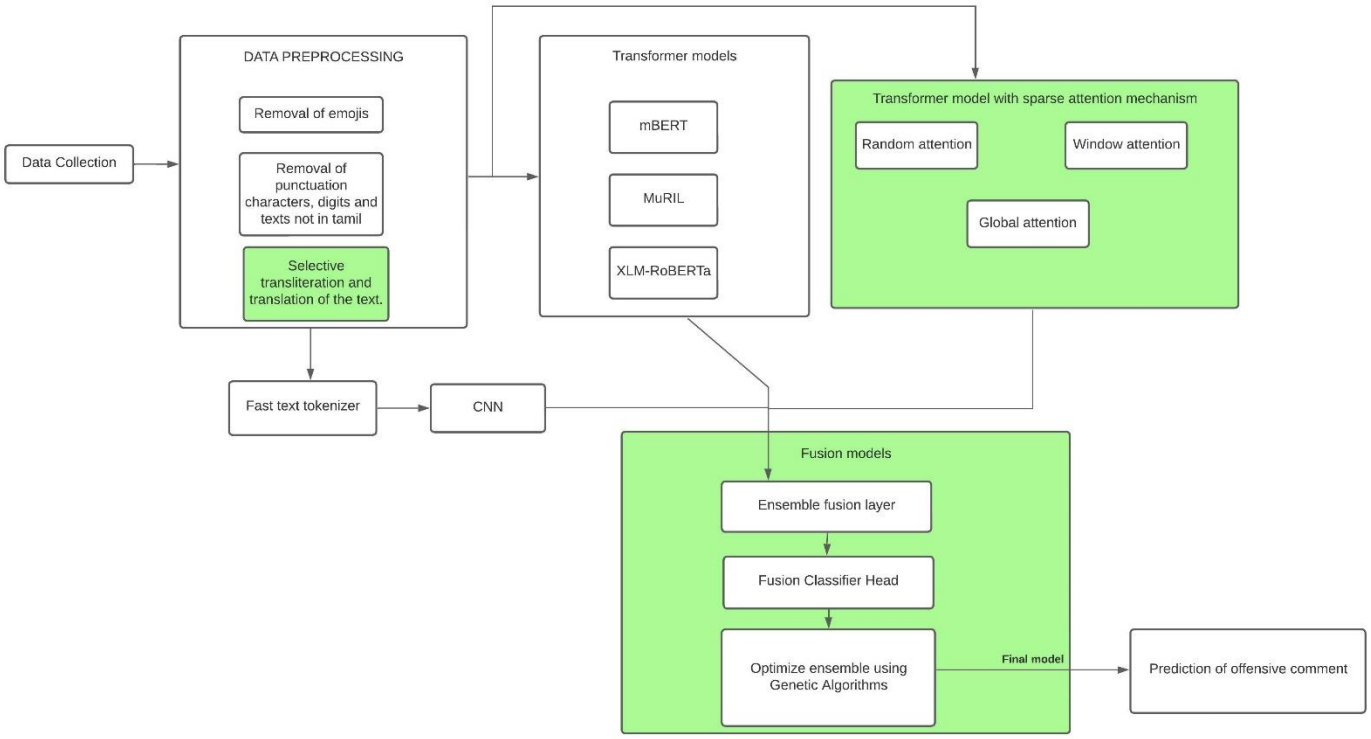
[4] Priyadharshini, R., Chakravarthi, B.R., Chinnaudayar Navaneethakrishnan, S., Durairaj, T., Subramanian, M., Shanmugavadivel, K., U Hegde, S., Kumaresan, P.K.,2022. Findings of the shared task on abusive comment detection in tamil. In: Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages. Association for Computational Linguistics. A comment/post-level classification task. Given a YouTube comment, the systems submitted by the participants should classify it as abusive categories. Fail to classify the sentences whenever the sentences contain only the English transliterated words.

[5] Nayel, H.A., Shashirekha, H., 2019. Deep at HASOC2019: A machine learning framework for hate speech and offensive language detection. In: FIRE (WorkingNotes). pp. 336–343. Hate Speech and Offensive Content Identification for three languages namely, English,

Germany and Hindi. Uses less effective ML models and it is not developed for low resourced languages like tamil.

[6] Ayo, F.E., Folorunso, O., Ibharalu, F.T., Osinuga, I.A., 2020. Machine learning techniques for hate speech classification of twitter data: State-of-the-art, futurechallenges and research directions. Comput. Sci. Rev. 38, 100311. An approach to automatically classify tweets on Twitter into three classes: hateful, offensive and clean. Issues of generic metadata architecture, scalability, class imbalance data, threshold settings and fragmentation.

BLOCK DIAGRAM



DETAILS OF MODULES

The list of modules involved in the entire process are as follows:

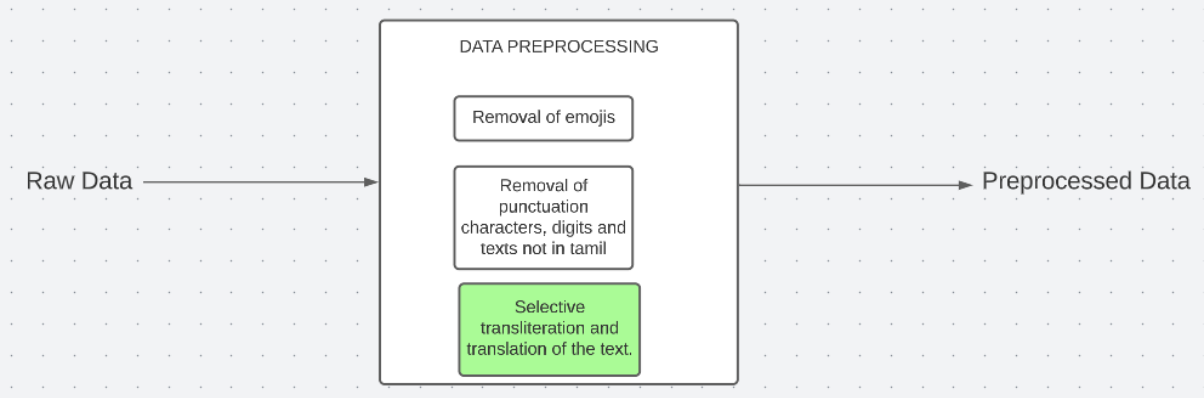
- Data Preprocessing
- Transformer Models
- Transformer model with Sparse Attention Mechanism
- Fusion Models

A. DATA PREPROCESSING

Converting raw data into a format which are clean is essential so that the NLP models can understand. We remove unwanted data from the corpus such as emojis, punctuation characters and words that were not in Tamil so that the rest of the data could be processed properly. Emojis are pictorial representations of an idea or emotion. They can be removed by substituting equal textual form of that emoji or by removing them altogether. We choose to remove the emojis and emotions from the text because they do not generally convey any semantic value. We also handle various punctuation characters such as !,?,etc and numerical digits by removing them completely as they do not serve our classification goal. Transformer architectures like mBERT and DistilmBERT are trained on multiple languages in the native script but not in the romanized script. As a solution to this process we propose selective transliteration and translation of the text. If a word is in English dictionary, then we translate the word into native Tamil script. We ignore the word if the detected language is native Tamil. If the word not processed by these two cases, then we transliterate the romanized word into native Tamil. The cross-lingual nature of the transformer-based models are used to efficiently handle these types of mixed script sentences.

Input: Raw data

Output: Preprocessed data



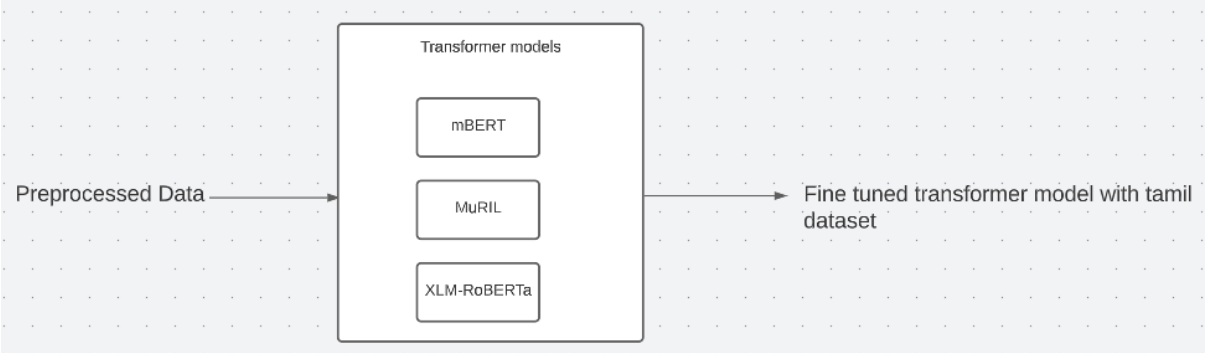
B. TRANSFORMER MODELS

- **mBERT:** An open source machine learning framework for natural language processing (NLP)which is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context.
- **muRil**: Latest multilingual model launched by Google and is aimed at improving interoperability from one language to another. As per Google, MuRIL supports 16 Indian languages.
- **XLM-ROBERTa:** A self-supervised transformer model which uses the MLM technique without the Next Sentence Prediction(NSP) technique.Entails sampling streams of text from various languages and masking some tokens so that it can anticipate the missing tokens.

For fine tuning a classification layer is added on top of the pre trained transformer models. The entire pre trained models are then retrained on training dataset. The output is fed to the softmax classification layer. While training the models, the model's pretrained weights are adjusted in accordance with the training dataset.

Input: Preprocessed data

Output: Fine tuned transformer model with Tamil dataset.



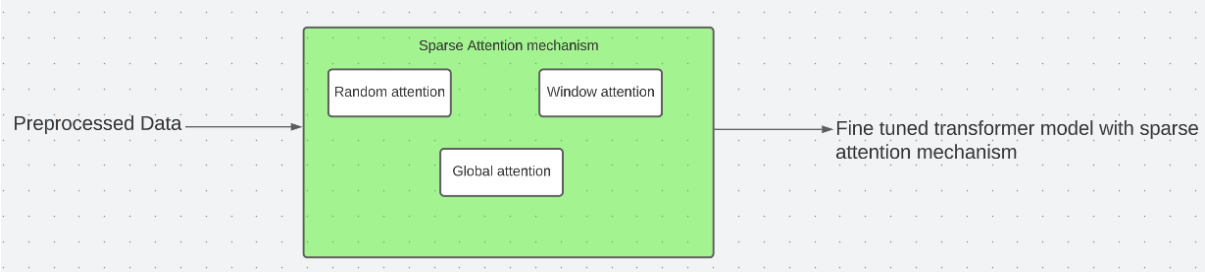
C. SPARSE ATTENTION MECHANISMS

Sparse attention mechanisms improve the performance of language models by allowing them to focus on specific parts of the input when making predictions. This mechanism only attends to a small subset of the input at a time, rather than the entire input. This allows the model to more efficiently process long sequences of text and improve its ability to handle tasks such as machine translation and text summarization. This uses a hierarchical structure that allows it to focus on more fine-grained information when necessary, which can help improve performance on tasks such as language understanding and translation.

- **Random Attention:** The attention weights are generated in a random manner for a subset of the input tokens, rather than computing attention for all tokens. By selecting a random subset of tokens encourages diversity in its attention which are beneficial for language generation and machine translation.
- **Window Attention:** The model will attend to a fixed-size window of source tokens around the target token being translated, rather than attending to the entire source sentence. This will help the model to focus better on the relevant context for each target token. It reduces the computational cost of the attention mechanism while still allowing the model to attend to relevant information in the input.
- **Global Attention:** The model attends to all input tokens, rather than a subset of them. This considers entire input when making predictions for text classification. The attention scores are computed between all pairs of query and key vectors, with the resulting attention weights being used to compute the weighted sum of values for each query. It can also be computationally expensive for long sequences as the model attends to all relevant information in the input.

Input: Preprocessed data

Output: Fine tuned transformer model with sparse attention mechanism

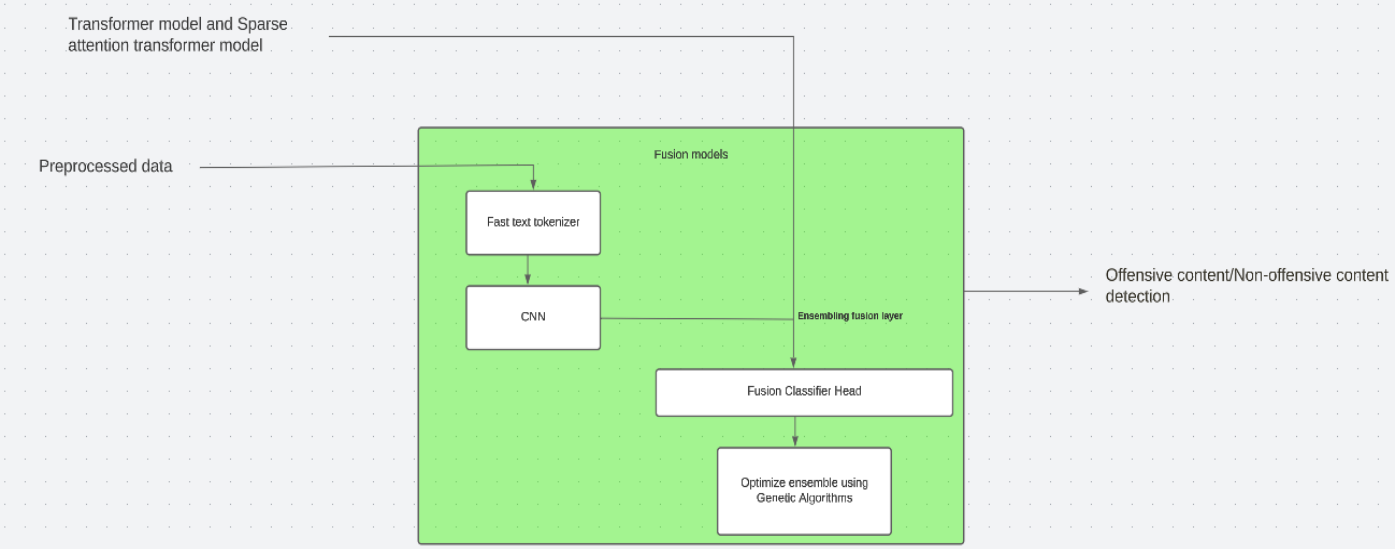


D.FUSION MODELS

We train a single classification head on the concatenated embeddings from different BERT,CNN models and sparse attention transformers. BERT models were initialized with the fine-tuned weights in the former section and the weights were frozen. The number of BERT models in a single fusion model was kept flexible with maximum number of models fixed to three, due to memory limitation. Ensemble of different models turn out better predictors than using a single classifier. One of the strategies to reduce the influence of weak models is using weights for BERT,CNN and sparse attention transformer models based on their performance. Genetic algorithm (GA) based technique is used to set the weights of different models in an ensemble. In this strategy instead of selecting the models with the highest weights for the final ensemble, we directly use the weights to compute the weighted average ensemble.

Input: Fine-tuned Transformer models, CNN, Sparse attention transformer models

Output: Offensive content/Non-offensive content detection



PERFORMANCE MEASURES

- **Accuracy**

$$Accuracy = (T\ P + TN)/(T\ P + TN + FP + FN)$$

- **Recall**

$$Recall = T\ P/(T\ P + FN)$$

- **Precision**

$$Precision = T\ P/(T\ P + FP)$$

- **F1-score**

$$F1score = (2 - Precision - Recall)/(Precision + Recall)$$

DATASET

Dataset used in the project are

1. **Dravidian Code-Mix FIRE 2021**

REFERENCES

- [1] Malliga Subramanian a, Rahul Ponnusamy b, Sean Benhur c, Kogilavani Shanmugavadivel a, Adhithiya Ganesan a, Deepti Ravi a, Gowtham Krishnan Shanmugasundaram a, Ruba Priyadharshini d,Bharathi Raja Chakravarthi e Offensive language detection in Tamil YouTube comments by adapters and cross-domain knowledge transfer.
- [2] Gaydhani, A., Doma, V., Kendre, S., Bhagwat, L., 2018. Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. arXiv preprint arXiv:1809.08651.
- [3] Dave, B., Bhat, S., Majumder, P., 2021. Irnlp_daiict @ dravidianlangtech-eacl2021: offensive language identification in Dravidian languages using TF-IDF char n-grams and MuRIL. In: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pp. 266–269.
- [4] Dowlagar, S., Mamidi, R., 2021. Hasocone@ fire-hasoc2020: Using BERT and multilingual BERT models for hate speech detection. arXiv preprint arXiv:2101.09007. Gao, L., Huang, R., 2017. Detecting online hate speech using context aware models. arXiv preprint arXiv:1710.07395.
- [5] Priyadharshini, R., Chakravarthi, B.R., Chinnaudayar Navaneethakrishnan, S., Durairaj, T., Subramanian, M., Shanmugavadivel, K., U Hegde, S., Kumaresan, P.K.,2022. Findings of the shared task on abusive comment detection in tamil. In: Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages. Association for Computational Linguistics.
- [6] Charangan Vasantharajan, Uthayasanker Thayasivam :Towards Offensive Language Identification for Tamil Code-Mixed YouTube Comments and Posts.
- [7] Hande, A., Puranik, K., Yasaswini, K., Priyadharshini, R., Thavareesan, S., Sampath, A., Shanmugavadivel, K., Thenmozhi, D., Chakravarthi, B.R., 2021. Offensive language identification in low-resourced code-mixed dravidian languages using pseudo-labeling. arXiv preprint arXiv:2108.12177.
- [8] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, Amr Ahmed. Big Bird: Transformers for Longer Sequences

- [9] Debjoy Saha , Naman Paharia , Debajit Chakraborty, Punyajoy Saha, Animesh Mukherjee:Hate-Alert@DravidianLangTech-EACL2021: Ensembling strategies for Transformer-based Offensive language Detection