On finetuning Adapter-based Transformer models for classifying Abusive Social Media Tamil Comments

Malliga Subramanian (mallisenthil.cse@kongu.edu)
Kongu Engineering College
Kogilavani Shanmugavadivel
Kongu Engineering College
Nandhini Subbarayan
Kongu Engineering College
Adhithiya Ganesan
Kongu Engineering College
Deepti Ravi
Kongu Engineering College
Vasanth Palanikumar
Chennai Institute of Technology
Bharathi Raja Chakravarthi
National University of Ireland Galway

Research Article

Keywords: Abusive language, Social Media, Tamil, Transformer, Adapter, mBERT, Muril, XLM-RoBERTa, Hyper-parameters, Optuna

Posted Date: February 22nd, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2601766/v1

Abstract

Speaking or expressing oneself in an abusive manner is a form of verbal abuse that targets individuals or groups on the basis of their membership in a particular social group, which is differentiated by traits such as culture, gender, sexual orientation, religious affiliation etc. In today's world, the dissemination of evil and depraved content on social media has increased exponentially. Abusive language on the internet has been linked to an increase in violence against minorities around the world, including mass shootings, murders, and ethnic cleansing. People who use social media in places where English is not the main language often use a code-mixed form of text. This makes it harder to find abusive texts, and when combined with the fact that there aren't many resources for languages like Tamil, the task becomes significantly challenging. This work makes use of abusive Tamil language comments released by the workshop “Tamil DravidianLangTech@ACL 2022” and develops adapter-based multilingual transformer models namely Muril, XLMRoBERTa and mBERT to classify the abusive comments. These transformers have been utilized as fine-tuners and adapters. This study shows that in low-resource languages like Tamil, adapter-based strategies work better than fine-tuned models. In addition, we use Optuna, a hyperparameter optimization framework to find the ideal values of the hyper-parameters that lead to better classification. Of all the proposed models, MuRIL (Large) gives 74.7%, which is comparatively better than other models proposed for the same dataset.

1. Introduction

The Internet allows people all over the world to communicate quickly and helps us to stay in touch with friends and family while also meeting new people from all over the world. Technological advancements have made it simple to share information, ask for help, and do other things. And, while the majority of these interactions is kind and respectful, some people use abusive language. An Abuse language refers to any type of insult, vulgarity, profanity, sexism, or misogyny [1] that debases the target, as well as anything that causes aggravation [2]. The term abusive language is frequently reframed as offensive language [3], and hate speech [4]. In recent years, an increasing number of users have witnessed offensive behavior on social media. Abusive comments upset regardless of who they are is directed at or who witnesses them, but it is especially upsetting when they affect children and teenagers who may lack the experience or emotional maturity to get over it, digest it, or know where to look. This is not only humiliating for some, but it also lowers their self-esteem, leading to depression, rage, and antisocial behaviour. So, abusive language can harm individuals, communities and society. As a result, the major social media firms have turned to a range of methods, such as human reviewers, user reporting systems, and other similar practices, in order to remove abusive comments and texts from their platforms. Still, the issue has not been resolved despite the various attempts that have been made. The subjectivity and context-dependent characteristics of abusive language detection are the main reasons for its failure [5]. Even, human annotators find it difficult to detect abusive language, making it challenging to build a large and reliable dataset. For the detection of abusive language on social media, Chatzakou et. al. [5] prepared datasets ranging from 10K to 35K in size. Because of the rise of social media, users from societies with

multilingualism have recently begun posting and leaving comments in a code-mixed way. In the code-mixed format, the vocabulary and grammar of a sentence are drawn from multiple languages [6]. Because the multilingual people are unable to convey themselves in a single language, they use code-mixed text to communicate [7]. Code-mixed content is widely disseminated on social media platforms like Facebook and Twitter due to a lack of oversight. Automated annotation of social media content is necessary as it is quiet challenging to manually identify such contents in the vast amount of data created on social media. Furthermore, abusive comment detection has rarely been explored for low-resource languages due to scarcity and unavailability of annotated dataset [8]. To address this issue, a shared task set has been created by Priyadharshini et. al.[1] that contains abusive comments gathered from social media and this task has invited researchers to build a variety of models to classify the abusive comments.

A lot of researchers in the field of Natural Language Processing (NLP) have been working on systems that can limit the spread of abusive contents or get rid of them entirely by using the most advanced NLP techniques. Several NLP systems have been proposed that can automatically find texts that are abusive. These systems can be put into two groups: those that use machine learning and deep learning, and those that use multilingual transformer models. The goal of this study is to use multilingual transformer models to classify abusive texts from a dataset of Tamil comments and posts from YouTube[1]. The study only looks at Tamil, which is a classical Indian language spoken in Sri Lanka, Malaysia, and Singapore, as well as India's southernmost state, Tamil Nadu. Recent research on abusive language detection suggests a greater usage of transformer-based deep learning models. These approaches necessitate a considerable quantity of labeled data, thus limiting their usefulness in many sectors where annotated resources are few [9]. In such circumstances, models capable of extracting linguistic information from unlabeled data might be used instead of manually labeling the data. Transfer learning is a learning method where a model is initially trained using self-supervised learning on large unlabeled text corpora before being applied to labeled text corpora.

Recently transformer-based models have attracted researchers for their capability to identify and categorize abusive texts through contextual and semantic learning. Examples of such models include Bidirectional Encoder Representations from Transformers (BERT) [9], Robustly Optimized BERT (RoBERTa) [10], Cross-lingual Language Model RoBERTa (XLM-RoBERTa) [11], Multilingual Representations for Indian Languages (MuRIL) [12] etc. These models performed remarkably well in accurately classifying code-mixed texts from several languages [13]. Two ways to explore these transformer models include (1) fine-tuning and (2) integrating an adapter and training. The pre-trained models, such as BERT, DistilBERT, RoBERTa, HateBERT, MuRIL, etc., are completely retrained on a new downstream task in the fine-tuning technique. Therefore, a significant number of parameters must be retrained in order to use this strategy. Conversely, adapters are light-weight, compact modules that are inserted between the layers of a transformer [14–16]. While model tuning on a new task, the weights of the original transformer layers are not modified, but only the weights at layers of adapters are updated. He et. al. [18] demonstrated that, for low-resource and cross-lingual tasks, adapter-based tuning

outperforms fine-tuning. The proposed study builds fine-tuned and transformer-based models to identify abusive language contents comprising of Tamil YouTube comments collected through the HASOC-Offensive Language Identification track in Tamil DravidianLangTech@ACL 2022. This study addresses the following research questions:

RQ1: How well do pre-trained and adapter-based transformer models distinguish and classify abusive social media Tamil texts?

To respond to this question, we fine-tuned and integrated adapters into a set of transformer models and evaluated their performance.

RQ2 : Does optimization of hyper-parameters have impact on the performance of the transformer modes?

To address this research question, we have used Optuna, a hyperparameter optimization framework to find the ideal values for a set of hyper-parameters

The following are the study's primary contributions:

1. Fine-tuned the transformer models including mBERT, MuRIL (Base and Large), and XLM-Roberta (Base and Large).
2. Integrated a light-weight, compact adapter module into pre-trained transformer models and evaluated the adapter efficacy in abusive language recognition in Tamil.
3. Investigated the models' performance to choose the best model for the classification task and performed error analysis.
4. Bayesian Optimization has been used to tune the hyperparameters to find the optimal values.
5. Augmented the text using NLPAug to address the imbalanced dataset.

Adapter-based transformer models introduce additional adapters between layers of the pre-trained models. As a result, the models only require a small number of task-specific parameters to be supplied. Due to frozen network settings, parameter sharing between the tasks is possible. Adapters employ common transformers, which perform well in NLP tasks like text categorization and others. The widespread use of abusive language obscures the potential benefits of social media platforms; hence, our effort focuses on finding solutions to this problem. Incorporation of light-weight adapters into transformer models and adaptation of produced models to the dataset under consideration are the innovative aspects of this research. Further, we use Bayesian optimization for fine-tuning a set of hyper-parameters.

The rest of the article is structured as follows: In Section 2, we take a quick look at how abusive language has been studied using deep learning and transformers. In Section 3, a description of the task, a summary of the dataset, and the steps used to prepare the data for the study are given. Section 4 describes the proposed models. The experimental details and training procedure are described in Section 5. Section 6 discusses about the findings from the experiments. In addition, Section 6 provides an in-

depth study of the errors and misclassifications of the texts. In Section 7, the proposed work is summed up and suggestions for future research are given.

2. Literature Survey

Due to a growth in internet and social media platform users, abusive language identification as well as hate speech detection have been the focus of study over the past decade. Because of the emergence of transformers and pretrained language models, present solutions for detecting abusive language rely heavily on deep learning techniques. In addition, various shared tasks on low-resource languages have been published to direct the focus of researchers, and academicians have worked to develop models for these tasks. Several of these initiatives are outlined below:

2.1 Shared tasks

In 2020, the first shared task[1] on identifying the abusive language in Dravidian languages such as Tamil, Malayalam, and Kannada has been released. The purpose of this task is to engage academic researchers to create models for recognizing abusive/ offensive language content in the code-mixed dataset collected from social media comments/posts in Dravidian Languages. The findings of this shared task have been reported by [17], and the authors have also provided an overview of the dataset used for this task as well as the methodologies and results of the systems proposed for this task. This shared task has stimulated interest in low-resource languages and encouraged further research. Another shared task[2] has been released to classify abusive comments as homophobia, misandry, counter speech, misogyny, transphobia, and so on. Models using machine/deep learning algorithms and transformers have been proposed for this shared task. The results of this shared task were analyzed by [18] and found that the transformer-based MuRIL model did the best out of all the others.

Chakravarthi et. al. [6] collected comments and posts in Dravidian languages (Malayalam-English and Tamil-English) from social media and released them as a shared task [3]. This shared task of figuring out which texts in Dravidian languages were offensive was summed up, and the results were published [6]. This report reveals that numerous models employ transformers and pre-trained embedding systems. In addition, Chakravarthi et. al. [19] has released a shared task[4] with the primary objective of detecting homophobic and transphobic texts in social media comments in Tamil, English, and Tamil-English and also reported on the results of this shared task. For this shared task, numerous pre-trained models and transformer models, such as BERT, mBERT, XLM-RoBERTa, IndicBERT, HateBERT, etc., have been utilized. Moreover, it has been found that the most effective approach utilized pre-trained XLM RoBERTa language model for zero-shot learning to address data imbalance and multilingualism. To detect hate speech and offensive contents in both English and Indo-Aryan languages, a new shared task [5] has been posted. The authors of [20] gave an overview of this shared task, which included the descriptions of the tasks, the data, and the results. Thus, the shared tasks are intended to motivate academics to address and advance

problems related to abusive/offensive text recognition and to draw attention to the need for more study into the identification of abusive contents in under-resourced languages.

2.2 Deep learning models

Since machine learning based models depend on well-defined feature extraction strategy, automated feature extraction models have come into use. In addition, these automated models are increasingly using text representation and deep learning approaches to detect abusive comments in order to enhance performance. We provide a quick summary of these models below.

Ashraf et. al. [21] investigated YouTube comments for identification of offensive comments. Several baseline machine learning models, including Multi-Layer Perceptron (MLP), AdaBoost, Random Forest (RF), Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), and Decision Tree (DT), as well as two neural network models namely Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM), were tested in this study. This work produced F1-scores of 91.96% for Ada-boost and 91.68% for CNN respectively. Lee et. al. [22] looked at neural network-based models including CNN, Recurrent Neural Networks (RNN), and the Gated Recurrent Unit (GRU) network models in addition to conventional machine learning techniques to learn about hate speech and abusive language detection on Twitter. With an F1-score of 80.5%, a bidirectional GRU network based on word-level features and Latent Topic Clustering modules did better than the other models. Emon et. al. [23] tested machine and deep learning algorithms such as Linear Support Vector Classifier (Linear SVC), LR, MNB, RF, Artificial Neural Network (ANN), and RNN with a Short Term Memory (LSTM) to check if they could find abusive Bengali texts. With an accuracy of 82.20%, the RNN algorithm with LSTM does better than other algorithms. In an attempt [24], transformer-based deep neural network models like BERT, ELECTRA etc. have been used. These models were tested on a new set of data with 44,001 comments from Facebook posts. Both BERT and ELECTRA had test accuracy rates of 85% and 84.9%, respectively. Sharif et. al. [25] proposed a few machine learning models (LR and SVM), deep learning techniques (LSTM and LSTM + Attention), and transformers (m-BERT, Indic-BERT, and XLM-R) to find offensive texts in the shared task[1] dataset. The authors showed that XLM-R performed better than other methods for Tamil and Malayalam comments, but m-BERT achieved the best score for Kannada comments.

Around 6,175 user-generated comments in code mixed Kannada were gathered by Hande et al. [26] from YouTube and classified as either hope speech or not-hope speech. Additionally, they developed DC-BERT4HOPE, a two-channel model that uses the English translation as additional training to strengthen the ability to recognize the word "hope". The weighted F1-score for this method is 75.6%, which is better than other models compared in their work. A detection strategy based on the ensemble of RNN classifiers that integrates user-related information, such as racism or sexism has been proposed by

Pitsilis et. al. [27]. The user-related information and word frequency vectors from the text have been submitted to the classifiers. The classifiers have been evaluated on a public corpus of 16k tweets, and the

results showed that the proposed classifiers can recognize racism and sexism messages from normal text better than existing state-of-the-art algorithms. [28] examined various machine learning techniques for identifying hope speech in short, casual texts written in English, Malayalam, and Tamil. The authors showed that, given enough training data, even extremely simple baseline algorithms do pretty well on this task. In this work [28], however, cross-lingual transfer learning, with XLM-RoBERTa, is found to be the best-performing algorithm. Glazkova et. al. [29] created models for the Shared Task 2021[5], for Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages. The authors used a one-versus-the-rest technique based on Twitter-RoBERTa to identify hateful, offensive, and profane comments, and these models obtained F1scores of 81.99% and 65.77% for two subtasks of the shared task, respectively. For the Marathi tasks, the authors presented a language-independent BERT Sentence Embedding system (LaBSE) and it produced an F1score of 88.08%. Steimel et. al. [30] recommended machine/deep learning models to classify comments in English and German into abusive and not abusive. The authors have experimented with several promising architectures, including fully connected neural networks and CNN, along with different word embeddings, with both BERT and Flair embeddings. In this work [30], it has been concluded that that a multilingual optimization of classifiers is not possible even in settings where comparable datasets are used.

El-Alami et. al. [31] presented a transfer learning-based method for classifying offensive language in multilingual texts. The transformer models, including BERT, mBERT, and AraBERT, which were honed for the multilingual offensive detection problem, served as the foundation for this approach. The findings of this study demonstrated that the proposed models were both more accurate and received higher F1score. Sundar et. al. [32] came up with a multilingual model that used a stacked encoder architecture to automatically find hope speech. In this work, language-independent cross-lingual word embeddings were used because the dataset was made up of mixed-code YouTube comments. An empirical analysis was also done, and the proposed architecture was tested against different traditional methods, transformer, and transfer learning methods. With this method, the F1 score for Tamil was 61% and for Malayalam it was 85%. Apart from developing models for classification, the researchers have also constructed benchmark datasets in various research efforts [33–35]. These attempts have created datasets for hope speech detection, abusive text identification etc. and made them publicly available to the research community. A set of metrics for evaluating and categorizing a dataset is also presented in these research attempts. These datasets will spur further research.

To summarize, we have investigated the use of fine-tuned deep learning-based transformer models to detect abusive comments. Despite the considerable amount of work on abusive comments detection using fine-tuned transformer models, we find the adapter-based models have not yet been tried out. Adapters accomplish the same functionalities as fine-tuning, but by adding layers to the pre-trained model and updating the weights of these additional layers while maintaining the pre-trained model's weights in a frozen state. Therefore, fine-tuning updates the weights of the pre-trained layers. As a result, adapters are significantly more time and storage efficient than fine-tuning. From the literature survey, we understand that no work uses an adapter-based transformer model and believe that such models will

improve the efficacy. So, in this work, we integrate adapters into the transformer models and evaluate the performance.

3. Taskset

3.1 Dataset Description

The dataset for the current work is taken from the shared task[2] that contains abusive comments collected from social networking sites. The task contains comments in Tamil that are classified into one of the eight predefined classes namely Misogyny, Misandry, Homophobia, Transphobia, Xenophobia, Counter-speech, Hope speech, and None-of-the-above. The goal of this shared task is to develop systems that can find abusive comments in a given set of Tamil texts. Although some of the comments in the dataset comprise multiple sentences, the corpora's average sentence length is one [18]. There are 2240 comments in the training dataset including 2 comments which are not in Tamil and we ignore these two comments. In the validation dataset, there are 560 comments and in the test dataset, there are 699

comments. As was said above, each comment in the training and validation dataset has been annotated with one of the eight labels. Table 1 shows the class distribution for each type of abusive comments in the dataset.

Table 1
Distribution of classes in taskset

Label	No. of Comments		
	Training set	Validation set	Testing set
Hope-speech	86	11	26
None-of-the-above	1296	346	416
Homophobia	35	8	8
Misandry	446	104	127
Counter-speech	149	36	47
Misogyny	125	24	48
Xenophobia	95	29	25
Transphobic	6	2	2

Table 2
Sample texts from the dataset for each class

Text	Label	Class definition
. . . . I was looking forward to your speech. Good description Awesome. You have registered in a reasonable manner. Congratulations on living a long life.	Hope-speech	Expressions evoking the optimism and similar pleasant emotions.
ena paninga? What did you do both together?	Homophobia	Dislike of or prejudice against gay people.
... What can be done to those who think that it is a disgrace to listen to wife's speech ?	Misandry	Hatred, dislike, or mistrust of men
That's right most women can't accept debt but if you think it will change brother.	Counter-speech	Combating a hateful speech.
Women do not have brains	Misogyny	Dislike of, contempt for, or ingrained prejudice against women.
You came through the Khyber Pass and you are not Indian	Xenophobia	Dislike of or prejudice against people from other countries
, You are not a guy, you are an ugly transgender	Transphobic	Having or showing a dislike of or prejudice against transsexual or transgender people
Mr. Srinivasan is a very good man	None of the above	Does not belong in any of the above categories

3.2 Preprocessing

The comments in the datasets used for this shared task have words mostly in Tamil and a few in English. Since the texts have been collected from social media, they are and have URLs, hashtags, other characters, and so on. Before the text is mined, the noise and unwanted characters are to be removed from the raw, unstructured textual data to create meaningful features that could help classify the samples. The following steps have been performed to prepare the data:

Getting rid of emojis

The text messages in the dataset have emojis and emoticons. They can be replaced with a textual term, or they can be taken out completely. In this study, their equivalents in the text have been used instead. For example, the clapping emoji has been changed to "clapping" ( ) in Tamil.

Getting rid of punctuation, numbers, and text that isn't in Tamil

Not only did we get rid of emojis and emoticons, but we also got rid of extra white spaces, punctuation marks like !,?, etc., and digits. Even though, these characters make it easier to read, they are not useful for figuring out what stance someone is taking.

3.3 Text Augmentation

Text augmentation is a process that allows us to enhance the size of training data artificially by producing many versions of real texts without actually gathering the data. To enhance images, we may simply rotate, sharpen, or crop various portions of the images, and the new data will still make sense. However, enhancing text data is quite challenging. Changing the sequence of words, for instance, may appear acceptable at first, but it can drastically alter the meaning. To assist in text augmentation, NLPAug, a python library is provided and it supports three different types of augmentation namely character level, word level and sentence level augmentation. In this work, we have used a few word and sentence level augmentation techniques such as replacing a few words with their synonyms and replacing words that have similar word embeddings to those words, replace words based on the context using powerful transformer models etc. Since the number of instances for all the classes except “None-of-the-above” is less, we used NLPAug to increase the number of instances of these classes. As “None-of-the-above” is the most frequently occurring class in the dataset, we have used random undersampling to balance uneven datasets by keeping all of the augmented data in the minority class and decreasing the size of the “None-of-the-above” class. Hence, after augmentation and undersampling, the training dataset became balanced. Following the preprocessing and augmentation, the dataset has become complete and more conducive to efficient data analysis.

4. Transformer-based Classifiers

RNNs and CNNs have been formerly utilized to construct models for NLP-related tasks such as offensive/abusive language detection, sarcasm detection [36], sentiment analysis [37, 38], text summarization, and so on. The problem with these networks is that they cannot keep up with the context and meaning of lengthy phrases that is, dealing with long-range dependencies remains difficult. Moreover, these model designs cannot be parallelized due to their sequential nature. These limitations have been addressed by focusing on the word that is presently being processed using transformer-based models. From the literature work, we learned that a transformer model may be used in two ways: fine-tuning and incorporating an adapter. We train and test both techniques in our study. Below, we present a brief overview of the models used in this study.

4.1Transformer Models

Due to their ability to capture the context and attention mechanism, transformer models have recently gained widespread use in NLP. Transformers-based models are state-of-the-art for a variety of downstream NLP tasks, and Hugging Face has made their implementation and fine-tuning quite easy and accessible. Transformers are designed to handle long-range dependencies while solving sequence-to-sequence problems. Vaswani et. al. [39] came up with a transformer, which is a model architecture that doesn't use recurrence and instead uses an attention mechanism to find global dependencies between input and output. The transformer lets a lot more parallelization happen. The transformer is the first transduction model to generate input and output representations without using sequence-aligned RNNs or convolution. Instead, it uses self-attention, which is a way to pay attention that relates different words in a single sequence so that a representation of the sequence can be made. With the self-attention mechanism, transformers use an encoder-decoder structure. A stack of numerous identical encoders and decoders make up the encoder and decoder blocks. One of the configurable hyperparameters is the number of units in the encoder and decoder stacks and is the same for both stacks. In [39], six encoders and decoders were used. The encoder is composed of encoding layers that iteratively process the input, while the decoder is composed of decoding layers that all do the same operation on the encoder's output. Each encoder layer is responsible for generating encodings that indicate which parts of the inputs are significant to one another. It passes its encodings as inputs to the subsequent encoder layer. Each decoder layer generates an output sequence from all the encodings by utilizing the contextual information they contain [40]. Each encoder and decoder layer uses the attention technique to do this. When a sentence is supplied to a transformer model, attention weights are simultaneously determined for each token. For every token in the context, the attention unit generates embeddings that include information about the token itself and a weighted combination of other relevant tokens [41]. The outputs of preceding decoders are used by the additional attention mechanism in each decoder layer before the decoder layer uses information from the encodings. Also, both the encoder and decoder layers have a feed-forward neural network. This lets them process the outputs even more [41]. Figure 1 depicts the architecture of a transformer model.

The transformer model extracts the features for each sentence/comment using a self-attention mechanism to determine the relative importance of all other words in the sentence to the word in question. In addition, no recurrent units are utilized to obtain these characteristics. The transformer architectures are either used in part or entirety in current state-of-the-art NLP models. In recent years, the debut of Google's Bidirectional Encoder Representations from Transformers (BERT) has been hailed as the beginning of a new era in the development of NLP. BERT uses bidirectional representation from unlabeled text using left and right context in all layers. Subsequently, BERT pre-trained models such as ALBERT, RoBERTa, XLM-RobertA, mBERT, MuRIL [42] and many more have been developed. These transformer-based NLP models were built for transfer learning [9, 43]. Transfer learning is self-supervised learning on large unlabeled text corpora [44] and then it can be fine-tuned on a downstream NLP task[9]. Labeled NLP datasets are generally tiny and training a model on a tiny dataset without pre-training would lower the results. Hence, pre-trained models are preferred in such cases. Also, the pre-trained models may

be fine-tuned to handle various NLP downstream tasks like text categorization, summarization, or question answering. The following section provides an overview of three pretrained conceptual frameworks upon which this study's models are built.

4.1.1 mBERT

mBERT, a self-supervised model [42], is trained on a massive multilingual dataset, which has unlabeled raw text. Rather than using monolingual English data for training, mBERT is trained using articles in 104 languages collected from Wikipedia and a vocabulary derived from common word components. It does not employ a language input marker and there is no method to force translation-equivalent pairings with identical representations.

4.1.2 XLM-RoBERTa

A multilingual version of the RoBERTa is called XLM-RoBERTa [45] and was trained on 2.5 TB of CommonCrawl data collected from 100 languages. This model utilizes the same training process as the RoBERTa model, and it does not require the usage of lang tensors in order to comprehend the language that is being used. This model excludes the next sentence prediction method and makes use of the masked language model. In order for the model to anticipate the words that are missing, the training process entails gathering text streams from a variety of languages and masking some words. The approach can accommodate code-switching because no linguistic embeddings are used. XLM-RoBERTa has shown exceptional performance in a wide variety of NLP tasks having multilingualism.

4.1.3 MuRIL

MuRIL, Google's most recent multilingual model, supports 17 Indian languages and attempts to increase linguistic interoperability. MuRIL's main objective is to boost the effectiveness of some downstream NLP operations while attempting to address issues with Indian languages such as transliteration and spelling differences. On every task in the cross-lingual XTREME test, MuRIL outperforms mBERT [45].

4.2 Fine-tuning the pre-trained transformer models

Generally, large datasets have been used to create the pre-trained models. A technique known as "model fine-tuning" allows these models to be further enhanced by running them on a small dataset. Fine-tuning means taking the weights from a model that has already been trained and adjusting them for the new task. It cuts down on the cost of computing and lets us use the best models without having to start from scratch. During fine-tuning, we add fully connected layers and a classification layer to the pre-trained model and train the models using a new classification task. In this attempt, the mBERT, XLM-RoBERTa (Base and Large), and MuRIL (Base and Large) transformer models have been fine-tuned and tested.

4.3 Adapter-based transformer models

Although they both fine-tuning and feature-based transfer learning need a new set of weights for each task, recent research shows that fine-tuning typically outperforms the other [46]. But when a new model

needs to be trained and fine-tuned for every downstream task, it leads to an excessive number of parameters, thus parameter inefficiency happens. To solve this issue, Houslby et. al. [16] have suggested a unique, parameter-efficient bottleneck adapter module. Recently, adapters have excelled at multi-tasking and cross-linguistic transfer learning [10, 47]. When deep networks are fine-tuned, a change is made to the top layers of the network. It is common practice to train both the new top layers and the original weights at the same time when fine-tuning a model. This is necessary because the label spaces for the original and downstream tasks are different. When using adapter modules, however, a small number of new parameters are introduced to a model, which is subsequently trained on the downstream task. Adapter tuning, on the other hand, preserves the original network's parameters so that they can be used by several tasks. The adapter tuning technique, in particular, entails incorporating new layers into the original network. There are many architectural choices for integrating adapter modules into transformers. Houslby et. al. [16] presented a simple design that gave a significant performance and is shown in Fig. 2.

Figure 2 shows that the standard transformer is used with an adapter layer added after each sub-layer and before the skip connection. The function of the adapters is to reduce the size of the original model's features to a smaller size and then increase them back to the original size. This keeps the number of parameters much smaller than in the original model. During training on the target task, all weights of the pre-trained language model are kept fixed. The only weights to be updated are those introduced by the adapter modules. This is what makes adapters so effective. Houlsby et. al. [16], Peters et. al. [10], Pfeiffer et. al. [48] and Kim et. al. [49] have recently employed adapter-based models to improve pre-trained transformer models. In their efforts, [16]and [48] used a benchmark dataset that contains different tasks such as sentiment analysis, question-answering etc. To handle a wide range of target tasks, including natural language inference, phrase pair tasks, relationship classification, sentiment review, and many more, Peters et al. [10] and Kim et. al. [49] employed adapter-based models. However, we found that the performance of the adapters on low-resource languages like Tamil has not been tested. In the current work, we use the adapter model proposed by Houlsby et. al. [16] to classify abusive comments in Tamil.

In this work, we develop models using the transformer models such as mBERT, MuRIL and XLM-Roberta. Two versions of MuRIL and XLMRoberta such as, base case and large case are available. The number of transformer layers in both cases of MuRIL is 12 and 17 respectively. In XLMRoberta, they are 12 and 24 respectively. We repurpose these models in two ways as fine tuner and adapter-based. The overall workflow of the proposed study is depicted in Fig. 3.

The incorporation of adapters and customization of the models for the dataset under consideration constitute the novelty of this work. Integrating adapter modules facilitates parameter efficiency without sacrificing performance. In addition, Bayesian optimization is used to fine-tune a set of hyper-parameters.

5. Experimental Setup And Results

This section describes the experimental setup, training process of all the pre-trained models and analyzes their performance. A series of tests have been conducted to assess the performance of the fine-tuned and

adapter-based transformer models. The experiments are detailed below.

5.1 Experimental Platform

The proposed models have been trained and tested using Python on Colab notebook environments. The pre-trained models have been imported from HuggingFace's Transformers library [50]. An Adapter-transformer is an extension of HuggingFace's Transformers library that integrates adapters into cutting-edge language models by using AdapterHub, a repository of pre-trained adapter modules. Transformer-based models need a lot of power and high-performance hardware to operate effectively, thus we executed the suggested models on a Graphical Processing Unit (GPU). Additionally, we used the Python library, NLPAug for textual augmentation.

5.2 Training the transformer models

This study looked at five different types of transformers: mBERT, XLM-RoBERTa with Base and Large and, MuRIL with Base and Large. There are two ways to use these transformers: to fine-tune all of the weights or to train only the adapter modules. A softmax classification layer has been placed on top of the pre-trained models having number of neurons equal to the number of classes in the dataset.

5.2.1 Fine-tuning

To fine-tune, we copy the weights from a pre-trained model and update these weights for the new task. So, we have initialized each model with its pre-trained parameters. The parameters were then fine-tuned using data from the downstream dataset that have been labeled. The pre-trained weights of the models were modified to fit the training dataset while the back propagation has been used to reduce the error.

5.2.2 T uning with Adapters

In fine-tuning, the parameters have been fine-tuned for the task under consideration. Furthermore, the original weights and the weights of the newly added fully connected and classification layers have been co-trained, lowering compactness. As a result, if the lower levels of a network are shared by the original and new task, fine-tuning is parameter efficient. Hence, a parameter-efficient module, adapter, has been integrated into transformer models. This requires only a smaller number of trainable parameters per task, and when the new tasks are introduced, old tasks do not need to be retrained. That is, we used transfer learning on the dataset without retraining the models. The weights of the layers in the pre-trained transformer models were left fixed, whilst the weights of the layers in the adapter were fine-tuned to their full potential. So, adapter weights are encased inside the transformer, which forces them to have representations that are compatible and similar across tasks. The training procedure is enumerated below:

1. Attach adapters to each of the different models of the transformer.
2. Append a softmax classification layer with eight neurons.
3. Configure the training parameters.

4. Freeze the weight of original transformer models
5. Update the task specific parameters in the adapters

5.3 Hyperparameter Tuning

Hyperparameters control the process of learning and changing these parameters affects the model's accuracy. The process of finding the optimal settings for hyperparameters is known as hyperparameter optimization. Hyperparameters employed in this work include the training epochs, learning rate, training batch size, evaluation batch size, and weight decay. An autonomous hyperparameter optimization software framework called, Optuna, specifically built for machine learning, has been utilized to optimize the hyper-parameters. Optuna's primary characteristics include automated search for optimal hyperparameters, the efficient search of vast spaces, and the elimination of unproductive trials for faster results. Tree-structured Parzen Estimator (TPE) is the default Bayesian optimization algorithm implemented by Optuna. In addition, grid search, random search, etc. are also supported. Table 3 shows the set of hyperparameters tuned in this study, as well as their search space and optimal values found after a series of trials.

Table 3
Hyperparameters with search space and tuned values

Hyperparameters	Search Space	mBERT	MuRIL (Base)	MuRIL (Large)	XLM-RoBERTa (Base)	XLM-RoBERTa (Large)
Learning rate	0.01 to 0.00004	0.00009	0.00007	0.0009	0.00086	0.00007
Number of Training Epochs	40 to 200	142	75	95	98	103
Weight decay	0.01 to 0.00004	0.000042	0.000064	0.00092	0.00079	0.000091
Batch size for training	32,64,128	64	64	128	32	64
Batch size for evaluation	32,64,128	32	64	64	32	32

6. Experimental Results And Findings

In this section, we provide the results of training and testing of various models described in Section 4 for detecting abusive texts.

6.1 Metrics for performance evaluation

The accuracy, precision, recall, F1 score, and macro and weighted average were used to measure how well the different classification models worked. We used True Positive (TP), True Negative (TN), False

Positive (FP), and False Negative (TN) to calculate the values of the above metrics. To calculate TP, FP, TP, and TN, we use Equations (1) to (4), where i = 1,2,3…, upto 8 denoting the eight classes.

$$tp_i = c_{ii}$$

1

$$fp_i = \sum_{l=1}^{n} (c_{li}) - tp_i$$

2

$$fn_i = \sum_{l=1}^{n} (c_{il}) - tp_i$$

3

$$tn_i = \sum_{l=1}^{n} \sum_{k=1}^{n} (c_{lk}) - tp_i - fp_i - fn_i$$

4

Using TP, TN, FP and FN, the accuracy, precision, recall and F1score as given in Equations (5) to (8)

$Accuracy$ = $((TP + TN)/ (TP + TN + FP + FN)$ (5)

$Recall$ =$TP /(TP$/FN) (6)

$Precision$ =$TP/ (TP$+FP) (7)

$F1score$ = (2*$Precision$*$Recall$)/(P$recision$+$Recall$) (8)

The number of occurrences of each class is not evenly distributed, as seen in Table 1. This suggests that a weighted average that is skewed toward the more common classes may be more likely to underestimate the error in the class that occurs less frequently. This issue can be resolved by using the macro average, which treats each class in the same manner. Therefore, in infrequent cases, the performance of the model can be more accurately represented.

6.2 Experimental results and discussion

In this section we present the performance of the proposed models and also, compare these results with performance of the models that have classified the dataset what we have considered.

6.2.1 Fine-tuned models - Results

During fine-tuning, we have retrained the models for the dataset under consideration. To find whether the fine-tuned models which have been fit on the training dataset gives an unbiased evaluation over an unseen dataset, we ran the models using the test dataset. The results of the same is presented in Table 3.

Table 3
Performance of Finetuned transformer models for test dataset

Classifiers	Class Labels	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
mBERT	Hope-speech	53.791%	26.923	24.138	25.455
	None-of-the-above		55.048	83.273	66.281
	Homophobia		25.000	66.667	36.364
	Misandry		66.929	41.463	51.205
	Counter-speech		51.064	22.430	31.169
	Misogyny		45.833	37.288	41.121
	Xenophobia		28.000	33.333	30.435
	Transphobic		0	0	0
	Macro Average		37.350	38.574	35.254
	Weighted Average		55.091	54.013	51.120
XLM-RoBERTa -Base	Hope-speech	61.946%	57.692	28.302	37.975
	None-of-the-above		61.779	88.316	72.702
	Homophobia		50.000	50.000	50.000
	Misandry		70.866	58.824	64.286
	Counter-speech		63.83	32.258	42.857
	Misogyny		50.000	36.364	42.105
	Xenophobia		52.000	38.235	44.068
	Transphobic		0	0	0
	Macro Average		50.771	41.537	44.249
	Weighted Average		61.946	62.906	60.213
XLM-RoBERTa -Large	Hope-speech	63.09%	64.286	36.735	46.753
	None-of-the-above		63.043	88.776	73.729
	Homophobia		55.556	55.556	55.556
	Misandry		72.519	60.127	65.744

	Counter-speech		62.000	33.696	43.662
	Misogyny		45.238	31.148	36.893
	Xenophobia		47.826	32.353	38.596
	Transphobic		50.000	50.000	0
	Macro Average		57.559	48.549	45.117
	Weighted Average		65.090	66.524	65.187
MuRIL - Base	Hope-speech	56.080%	61.538	23.881	34.409
	None-of-the-above		54.327	87.597	67.062
	Homophobia		62.500	55.556	58.824
	Misandry		77.953	47.143	58.754
	Counter-speech		40.426	28.358	33.333
	Misogyny		43.75	28.767	34.711
	Xenophobia		36.000	60.000	45.000
	Transphobic		0	0	0
	Macro Average		47.062	41.413	41.51153
	Weighted Average		59.509	57.249	54.773
MuRIL - Large	Hope-speech	59.94%	56.522	24.074	33.766
	None-of-the-above		65.261	86.23	74.294
	Homophobia		22.727	31.25	26.316
	Misandry		65.854	53.289	58.909
	Counter-speech		52.381	26.506	35.2
	Misogyny		45.098	39.655	42.202
	Xenophobia		36.364	40.000	38.095
	Transphobic		0	0	0
	Macro Average		43.026	37.626	38.598
	Weighted Average		59.943	67.242	62.131

From Table 3, it is understood that MuRIL (Large) gives better accuracy compared to other models. Since, MuRIL has been trained on transliterated data as well, a phenomenon commonly found in the Indian context, it has performed better than other models. Figure 4 depicts the confusion matrices for the fine-tuned models for the test dataset.

6.2.2 Adapter-based transformer models - Results

Since fine-tuning requires a large number of parameters to be trained, we have integrated adapter-based modules into the transformer, trained and tested them instead, which only needs a small number of parameters to be trained. Like the fine-tuned models, we have also tested the adapter-based models which have been fit on the training dataset to see whether they give an unbiased evaluation over an unseen dataset. The results of the same is presented in Table 4.

Table 4
Performance of adapter-based transformer models for test dataset

Classifiers	Class Labels	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
mBERT	Hope-speech	56.08	42.308	20.000	27.16
	None-of-the-above		57.933	80.602	67.413
	Homophobia		62.500	83.333	71.429
	Misandry		65.354	47.701	55.15
	Counter-speech		44.681	22.826	30.216
	Misogyny		41.667	48.78	44.944
	Xenophobia		40.000	33.333	36.364
	Transphobic		50.000	50.000	50.000
	Macro Average		50.555	48.322	47.834
	Weighted Average		55.080	56.553	54.529
XLM-RoBERTa -Base	Hope-speech	58.286%	50.000	24.528	32.911
	None-of-the-above		59.135	85.714	69.986
	Homophobia		75.000	66.667	70.588
	Misandry		67.717	52.121	58.904
	Counter-speech		51.064	24.742	33.333
	Misogyny		50.000	39.344	44.037
	Xenophobia		36.000	33.333	34.615
	Transphobic		0	0	0
	Macro Average		48.614	40.806	43.0469
	Weighted Average		58.369	58.714	56.888
XLM-RoBERTa -Large	Hope-speech	58.083%	61.538	27.586	38.095
	None-of-the-above		57.452	85.663	68.777
	Homophobia		62.500	45.455	52.632
	Misandry		70.079	51.149	59.136

	Counter-speech		61.702	26.852	37.419
	Misogyny		39.583	46.341	42.697
	Xenophobia		36.000	36.000	36.000
	Transphobic		0	0	0
	Macro Average		48.606	39.881	41.844
	Weighted Average		60.083	58.096	56.431
MuRIL - Base	Hope-speech	59.51%	50.000	23.636	32.099
	None-of-the-above		60.817	84.899	70.868
	Homophobia		62.5	71.429	66.667
	Misandry		70.079	48.108	57.051
	Counter-speech		51.064	31.579	39.024
	Misogyny		45.833	42.308	44
	Xenophobia		40.000	38.462	39.216
	Transphobic		0	0	0
	Macro Average		47.537	42.553	43.616
	Weighted Average		59.514	60.368	57.547
MuRIL - Large	Hope-speech	74.68%	66.667	52.381	58.667
	None-of-the-above		78.250	91.520	84.367
	Homophobia		77.778	63.636	70.000
	Misandry		73.643	68.345	70.896
	Counter-speech		65.306	45.714	53.782
	Misogyny		63.043	54.717	58.586
	Xenophobia		74.194	56.098	63.889
	Transphobic		50.000	100.000	66.667
	Macro Average		68.610	66.551	65.856
	Weighted Average		74.678	77.857	75.683

Figure 5 depicts the confusion matrices that have been generated by the adapter-based models for the test dataset. As can be seen in Fig. 5, Muril-large case gives accuracy of 74.68% being the highest and mBERT gives accuracy of 56.08% being the least among the proposed models. Muril large case has a larger vocabulary, allowing the model to identify any word, gives more parameters, and the added complication is justified by the improved performance. So, it gives the highest accuracy. Figure 6 compares the performance of both fine-tuned and adapter-based transformer models. From Fig. 6, we can see that adapter-based MuRIL (Large) model performed well for most the abusive classes.

6.2.3 Performance comparison

As we have used the dataset provided for the shared task[2], we have compared the performance of the proposed adapter-based models with the models that attempted to classify the same dataset. The findings of the such attempts are summarized and presented in [51]. Table 5 also presents the same.

Table 5
Performance Comparison with other models

Reference	Model	accuracy	Macro Average of		
			Precision	Recall	F1 Score
[52]	SVM and TF-IDF	-	0.51	0.28	0.31
	SVM and TF-IDF With SMOTE	-	0.41	0.31	0.33
	SVM and TF-IDF with RKS	-	0.50	0.29	0.32
	SVM and TF-IDF with RKS and SMOTE	-	0.43	0.32	0.34
[53]	MuRIL	-	-	-	0.43
	XLM-R-base				0.43
	M-BERT				0.40
	IndicBERT				0.40
[54]	MuRIL	0.64	-	-	0.17
	Without augmentation	0.55	-	-	0.16
	With augmentation				
[55]	Indic-BERT	0.69	0.22	0.20	0.19
[56]	MuRIL	-	-	-	0.33
[57]	BiLSTM	-	0.74	0.67	0.7
	mBERT	-	0.64	0.7	0.7
[58]	XLM-RoBERTa	0.66	0.65	0.66	0.65
[59]	n-gram-MLP model	-	-	-	0.12
[60]	BERT	0.06	-	-	0.09
Proposed Adapter-based transformer models	**mBERT**	**0.561**	**0.506**	**0.483**	**0.478**
	MuRIL – base	**0.497**	**0.464**	**0.350**	**0.380**
	MuRIL – large	**0.747**	**0.686**	**0.666**	**0.650**
	XLMRoberta – base	**0.583**	**0.486**	**0.408**	**0.430**
	XLMRoberta - large	**0.581**	**0.486**	**0.399**	**0.418**

As can be seen in Table 5, adapter-based MuRIL (Large) obtained better accuracy than other proposed and existing models. Even though, XLM-RoBERTa, proposed by [58] gives better performance than models proposed in the current study, MuRIL-large has outperformed this performance. But, as shown in Table 5, the models proposed in [57] have given better performance than proposed models. To justify the performance of the proposed models, we have compared the parameter efficiency of the transformer based models and the results are discussed in Section 6.3.

6.3 Parameter Efficiency

For every new task, if we fine-tune and retrain models in entirety, it will lead to an excessive amount of parameters and result in Parameter inefficiency. Fine-tuning a model copies and adjusts the weights of a pre-trained network for each new task. This increases the number of parameters to be retrained, thus reducing the compactness of the models. Instead, the adapter modules designed by Houlsby et. al. [16] just add a small number of trainable parameters for each new task and do not necessitate the examination of prior tasks when new tasks are introduced. This work aims to facilitate the transfer learning to downstream tasks without requiring retraining of each model. But, transfer learning the knowledge from pretrained models is advantageous yet parameter inefficient. For fine-tuning M tasks, we need to retrain M times the number of parameters in the models. However, in the suggested adapter models, additional modules called adapters are introduced between the layers of a pre-trained network. When adapters are incorporated into transformers, merely the adapter blocks and normalization weights for each layer are changed and these layers have a small number of parameters. Incredibly, changing from fine-tuning to adapter-based model had almost no influence on accuracy and in fact, there was a gain in performance. Table 6 illustrates the number of trainable parameters that may be adjusted for both the ways of exploring transformer models.

Table 6
Number of parameters retrained

Models	Number of parameters retrained	
	Finetuned	Adapter-based
mBERT	177859592	1491272
MuRIL(Base)	237562376	1491272
MuRIL (Large)	505915400	4229640
XLM-Roberta (Base)	278049800	1491272
XLM-Roberta (Large)	559898632	4229640

As seen in Table 6, the number of trainable parameters is very meagre for adapter-based models when compared to fine-tuned versions. For instance, in case of the adapter-based XLM-Roberta (Large), the

number of trainable parameters to be retrained is 131 times lesser than finetuned version of the same model. Consequently, adapters are significantly more time- and space-efficient than fine-tuning.

6.4 Findings and Discussion

Considering the recent vogue of transformers and their focus of long-range dependencies in sequence-sequence tasks, we have employed transformer-based models. In the present attempt, we experimented with five different transformer models to find their ability to classify the abusive dataset. These models have been used in two ways: fine-tuned models and adapter-based models. While using as a fine-tuner, we have added a classification layer on the top of each model and the weights of the models have been retrained for the dataset under consideration. In case of adapter-based models, an adapter layer is integrated into each layer of the transformers and the weights for the adapter modules have only been retrained. The results of these experiments have been presented in Section 6.2. MuRIL (Large) performed 25% better than other models. This model would have done well on the dataset used in this study because it has been built to support transliteration, different spellings, mixed languages, and other use cases present in the Indian context and languages. So, we assume that MuRIL has outperformed other models. MuRIL significantly beats mBERT, which does not include transliterated text. This model, on the other hand, is complicated and time-consuming to run, resulting in increased number of parameters. However, the model's complexity can be justified by the improved performance.

We have also compared the performance of the proposed models against the performance of other models that have classified the same dataset and the results have been projected in Table 5. These models have been presented in the shared task[2] and the results have been summarized by [18]. From Table 5, it is found that the adapter-based transformer performs equally well with a very a smaller number of parameter retraining. The novelty in this work is that how the adapter modules are integrated to increase parameter efficiency without sacrificing performance. Table 6 shows how well the adapter transformer model works in terms of its parameters. The results show that fine-tuned models need to be trained on a much larger number of task-specific parameters than adapter-based transformer models.

Even though, adapter-based transformer models outperform fine-tuned models, they couldn't perform well on the categories in the test data that have a fewer instances. While testing the dataset without augmentation, we find that the models could correctly classify the instances which are high in number. For the classes like Xenophobia and Homophobia, the values of the different metrics have been 0 only. While investigating the cause of poorer accuracy for specific classes, we found that imbalance in the dataset is one of the reasons. Since the dataset is imbalanced and this imbalance have impact on model’s performance, we have augmented the dataset using NLPAug, a python library for textual augmentation. We performed word-level, contextual word embedding and random augmentation. The classification performance of the models with augmented dataset has been presented in Section 6. When we examine further on why augmented dataset did outperform the original unbalanced dataset, we realize that word-level, contextual word embedding and random augmented sentences lead to the

improved performance. Nevertheless, we believe that the performance of the models on human re-annotation of the augmented dataset may further be improved.

6.2.5 Error Analysis

To better comprehend the difficulties of this classification task, we performed an analysis of the misclassification errors induced by the proposed models. Even though, adapter-based Muril (Large) outperformed other models in classifying the abusive texts, it still has significant misclassifications. In order to determine how well the models performed across the various classes, it is required to look into the misclassification made by the models. Error Analysis is the process of investigating texts in test dataset that the models incorrectly categorised and identifying the underlying reasons of the errors.

For instance, from the Fig. 5(a) which depicts the confusion matrix obtained for mBERT model (adapter based), we can observe that the true positive is 11, meaning that this model successfully identified 11 instances of "Hope Speech" out of the 26 Hope speech texts in the dataset. Table 7 presents a few examples. Take a look at the finding in text 1. The actual class of the text is None-of-the-above, but mBERT has classified it into “Counter-speech”. But, the other models have classified as “Hope-speech”. Since the word “ (support)” seems to be a positive word, these models have classified the text as “Hope-speech”. In case of text 2, even though its actual class is “None-of-the-above” and it seems to give an explanation and expectation, it is classified as “Counter-speech” by mBERT, XLMRoberta (Large) and MuRIL (Base) models and predicted correctly by MuRIL (Large). Since this text has words such as “Tamil Nadu”, “India”, we assume that XLM-Roberta (Base) has tagged this text as “Xenophobia”, which means commenting against people from other countries.

Table 7
Error analysis

Text	Actual class	Predicted class (Adapter based transformer models)				
		mBERT	MuRIL (Base)	MuRIL (Large)	XLM-Roberta (Large)	XLM-Roberta (Base)
1. !!!!	None-of-the-above	Counter-speech	Hope-speech	Hope-speech	Hope-speech	Hope-speech
2. . . ? Sir, he clearly explained that the correct explanation is that the Barbarians are different from the Barbarian ideology. No matter who runs the school hall, the management of that school is fully responsible for the safety and discipline of the students studying in that school. If Sister Madhuvanti Parbanar's brain is high, is Dr. Ambedkar a Parbanar (Brahmins) who drafted the Constitution of India? To cover up the crime, Jeyalalitha, Brahmin, is not currently in power in Tamil Nadu. We expect the law to do its duty	None-of-the-above	Counter-speech	Counter-speech	None-of-the-above	Counter-speech	Xenophobia

Because it's unclear what separates a hope speech from non-abusive comments, text 2 in Table 7, even though it’s actual class is “None-of-the-above”, it might have been classified as “Counter-speech” by MuRIL based models. Especially, there are a few misclassifications between “None-of-the-above”,

“Counter-speech” and “hope-speech” classes. Even though, we augmented the training dataset, the augmentation is simply replacing the synonyms and contextual words. Instead of augmentation, if we could really frame and add actual sentences for infrequent classes, the performance could be further improved. We also understood that it would be challenging for even humans to classify some comments into the eight classes. Nonetheless, it is vital to examine the errors produced by the models to evaluate the classifier's performance across the various classes. Figures 4 and 5 present confusion matrices that show the misclassification errors. We noticed that, amid the eight classes, XLM-Roberta (Large) obtained a relatively high True Positive Rate (TPR) for four classes namely Hope-speech (61.5%), Homophobia (62.5%), Misandry (70.1%) and Counter-speech (61.7%). And, for Transphobia, XLM-Roberta (Large) gave 0% TPR. Most of the models experienced misclassifications for Transphobia class. Even though, we have augmented the infrequent classes, these texts are not real Transphobia class, but just shuffled or rephrased comments. This has resulted in a large number of wrongly classified texts in the test dataset.

7. Conclusion And Future Work

On the internet, abusive language is a huge concern, and it can lead to serious societal issues, and online platforms strive to prevent social harm and provide a conducive atmosphere for their users by limiting the use of such language. A variety of strategies for automatically detecting abusive language have been developed by researchers in the field of NLP.

This study took the abusive texts provided by DravidianTechLang ACL 2022 shared task and makes a contribution to this task by examining a set of classification models for identifying various forms of abusive texts. Five pre-trained models namely mBERT, MuRIL (Base and Large case), and the XLM-Roberta (Base and Large case) have been employed as fine-tuners and adapter-based models, with fine-tuning requiring the model to be retrained in its entirety, and adapters requiring only small bottleneck layers to be integrated into the pre-trained models. We have demonstrated that these adapter models require less training parameters than fine-tuning models. The accuracy of the adapter-based Muril large case model was higher than that of the other models. From this study, we also understand that adapter-based models outperform finetuned models in terms of performance and parameter efficiency.

Effective text augmentation and oversampling of minority classes will be used in the future to overcome data imbalance issues. Furthermore, we intend to look into how adapters might be used for future research, such as adapter migration to another downstream task, multitask learning, and stacking numerous adapters. Dravidian languages other than Tamil and domain-specific embeddings can be added to fine-tune the model's performance.

Declarations

Ethics approval and consent to participate : Not Applicable

Consent for publication : Not Applicable

Availability of data and materials : The datasets analysed during the current study are available in the “Abusive Comment Detection in Tamil-ACL 2022” repository : https://competitions.codalab.org/competitions/36403

Competing interests : The authors declare that they have no competing interests

Funding : This work is not supported by any funding

Authors' contributions : Author Contributions: Data curation : S.M., K.S. ; Formal analysis: B.R.C, N.S.; Investigation : A.G., K.S. ; Methodology : S.M.,.N.S. ; Project administration: A.G.,D.R.,S.M. Validation: B.R.C.,K.S.; Visualization: V.P., D.R.; Writing— Original Draft Preparation : S.M., K.S.; Writing—review and editing : B.R.C., M.S.;

Acknowledgements:We also render our sincere thanks to the organizers of DravidianTechLang ACL 2022 shared task for having provided with datasets.

References

1. Butt S, Ashraf N, Sidorov G, Gelbukh AF. Sexism Identification using BERT and Data Augmentation-EXIST2021. InIberLEF@ SEPLN 2021 Sep. pp. 381-389.
2. Spertus E. Smokey: Automatic recognition of hostile messages. InAaai/iaai 1997 Jul 27, pp. 1058-1065.
3. Razavi AH, Inkpen D, Uritsky S, Matwin S. Offensive language detection using multi-level classification. InAdvances in Artificial Intelligence: 23rd Canadian Conference on Artificial Intelligence, Canadian AI 2010, Ottawa, Canada, May 31–June 2, 2010. Proceedings 23 2010, pp. 16-27, Springer Berlin Heidelberg.
4. Djuric N, Zhou J, Morris R, Grbovic M, Radosavljevic V, Bhamidipati N. Hate speech detection with comment embeddings. InProceedings of the 24th international conference on world wide web 2015 May 18, pp. 29-30.
5. Chatzakou D, Kourtellis N, Blackburn J, De Cristofaro E, Stringhini G, Vakali A. Mean birds: Detecting aggression and bullying on twitter. InProceedings of the 2017 ACM on web science conference 2017 Jun 25, pp. 13-22.
6. Chakravarthi BR, Anand Kumar M, McCrae JP, Premjith B, Soman KP, Mandl T. Overview of the track on HASOC-Offensive Language Identification-DravidianCodeMix. InFIRE (Working notes) 2020 Dec 16, pp. 112-120.
7. Suryawanshi S, Chakravarthi BR. Findings of the shared task on Troll Meme Classification in Tamil. InProceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages 2021 Apr, pp. 126-132

8. Amjad M, Zhila A, Sidorov G, Labunets A, Butt S, Amjad HI, Vitman O, Gelbukh A. Urduthreat@ fire2021: Shared track on abusive threat identification in urdu. InForum for Information Retrieval Evaluation 2021 Dec 13, pp. 9-11.
9. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.: 'Bert: Pre-training of deep bidirectional transformers for language understanding', arXiv preprint arXiv:1810.04805, 2018
10. Peters, M.E., Ruder, S., and Smith, N.A.: 'To tune or not to tune? adapting pretrained representations to diverse tasks', arXiv preprint arXiv:1903.05987, 2019
11. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V.: 'Unsupervised cross-lingual representation learning at scale', arXiv preprint arXiv:1911.02116, 2019
12. Dave, B., Bhat, S., and Majumder, P.: 'IRNLP_DAIICT@ DravidianLangTech-EACL2021: offensive language identification in Dravidian languages using TF-IDF char n-grams and MuRIL', in Editor (Ed.)^(Eds.): 'Book IRNLP_DAIICT@ DravidianLangTech-EACL2021: offensive language identification in Dravidian languages using TF-IDF char n-grams and MuRIL' (2021, edn.), pp. 266-269
13. Chakravarthi, B.R., Muralidaran, V., Priyadharshini, R., and McCrae, J.P.: 'Corpus creation for sentiment analysis in code-mixed Tamil-English text', arXiv preprint arXiv:2006.00206, 2020
14. Mahabadi, R.K., Ruder, S., Dehghani, M., and Henderson, J.: 'Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks', arXiv preprint arXiv:2106.04489, 2021
15. Semnani, S., Sadagopan, K.R., and Tlili, F.: 'BERT-A: Finetuning BERT with Adapters and Data Augmentation', Standford University, 2019
16. Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, De Laroussilhe Q, Gesmundo A, Attariyan M, Gelly S. Parameter-efficient transfer learning for NLP. InInternational Conference on Machine Learning 2019 May 24, pp. 2790-2799.
17. Chakravarthi BR, Priyadharshini R, Jose N, Mandl T, Kumaresan PK, Ponnusamy R, Hariharan RL, McCrae JP, Sherly E. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. InProceedings of the first workshop on speech and language technologies for Dravidian languages 2021 Apr, pp. 133-145.
18. Shanmugavadivel, K., Hegde, S.U., and Kumaresan, P.K.: 'Overview of Abusive Comment Detection in Tamil-ACL 2022', DravidianLangTech 2022, 2022, pp. 292
19. Madasamy, A.K., Hegde, A., Banerjee, S., Chakravarthi, B.R., Priyadarshini, R., Shashirekha, H.L., and McCrae, J.P.: 'Overview of the Shared Task on Machine Translation in Dravidian Languages', DravidianLangTech 2022, 2022, pp. 271
20. Modha S, Mandl T, Shahi GK, Madhu H, Satapara S, Ranasinghe T, Zampieri M. Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages and conversational hate speech. InForum for Information Retrieval Evaluation 2021 Dec 13. pp. 1-3.
21. Ashraf, N., Zubiaga, A., and Gelbukh, A.: 'Abusive language detection in youtube comments leveraging replies as conversational context', PeerJ Computer Science, 2021, 7, pp. e742

22. Lee, Y., Yoon, S., and Jung, K.: 'Comparative studies of detecting abusive language on twitter', arXiv preprint arXiv:1808.10245, 2018
23. Emon EA, Rahman S, Banarjee J, Das AK, Mittra T. A deep learning approach to detect abusive bengali text. In2019 7th International Conference on Smart Computing & Communications (ICSCC) 2019 Jun 28,
24. Aurpa, T.T., Sadik, R., and Ahmed, M.S.: 'Abusive Bangla comments detection on Facebook using transformer-based deep learning models', Social Network Analysis and Mining, 2022, 12, (1), pp. 1-14
25. Sharif, O., Hossain, E., and Hoque, M.M.: 'Nlp-cuet@ dravidianlangtech-eacl2021: Offensive language detection from multilingual code-mixed text using transformers', arXiv preprint arXiv:2103.00455, 2021
26. Hande, A., Priyadharshini, R., Sampath, A., Thamburaj, K.P., Chandran, P., and Chakravarthi, B.R.: 'Hope speech detection in under-resourced kannada language', arXiv preprint arXiv:2108.04616, 2021
27. Pitsilis, G.K., Ramampiaro, H., and Langseth, H.: 'Detecting offensive language in tweets using deep learning', arXiv preprint arXiv:1801.04433, 2018
28. Ziehe S, Pannach F, Krishnan A. GCDH@ LT-EDI-EACL2021: XLM-RoBERTa for hope speech detection in English, Malayalam, and Tamil. Inproceedings of the first workshop on language Technology for Equality, diversity and inclusion 2021 Apr, pp. 132-135
29. Glazkova, A., Kadantsev, M., and Glazkov, M.: 'Fine-tuning of pre-trained transformers for hate, offensive, and profane content detection in english and marathi', arXiv preprint arXiv:2110.12687, 2021
30. Steimel K, Dakota D, Chen Y, Kübler S. Investigating multilingual abusive language detection: A cautionary tale. InProceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019) 2019 Sep, pp. 1151-1160
31. El-Alami, F.-z., El Alaoui, S.O., and Nahnahi, N.E.: 'A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model', Journal of King Saud University-Computer and Information Sciences, 2021
32. Sundar, A., Ramakrishnan, A., Balaji, A., and Durairaj, T.: 'Hope Speech Detection for Dravidian Languages Using Cross-Lingual Embeddings with Stacked Encoder Architecture', SN Computer Science, 2022, 3, (1), pp. 1-15
33. Chakravarthi BR. HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion. InProceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media 2020 Dec, pp. 41-53
34. Chakravarthi, B.R., Priyadharshini, R., Ponnusamy, R., Kumaresan, P.K., Sampath, K., Thenmozhi, D., Thangasamy, S., Nallathambi, R., and McCrae, J.P.: 'Dataset for identification of homophobia and transophobia in multilingual YouTube comments', arXiv preprint arXiv:2109.00227, 2021
35. Jose N, Chakravarthi BR, Suryawanshi S, Sherly E, McCrae JP. A survey of current datasets for code-switching research. In2020 6th international conference on advanced computing and communication systems (ICACCS) 2020 , pp. 136-141.

36. Vinoth, D., and Prabhavathy, P.: 'Automated sarcasm detection and classification using hyperparameter tuned deep learning model for social networks', Expert Systems, pp. e13107
37. Osmani, A., Mohasefi, J.B., and Gharehchopogh, F.S.: 'Enriched latent Dirichlet allocation for sentiment analysis', Expert Systems, 2020, 37, (4), pp. e12527
38. Asghar, M.Z., Sattar, A., Khan, A., Ali, A., Masud Kundi, F., and Ahmad, S.: 'Creating sentiment lexicon for sentiment analysis in Urdu: The case of a resource-poor language', Expert Systems, 2019, 36, (3), pp. e12397
39. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I.: 'Attention is all you need', Advances in neural information processing systems, 2017, 30
40. Alammar, J.: 'The illustrated transformer', The Illustrated Transformer–Jay Alammar–Visualizing Machine Learning One Concept at a Time, 2018, 27
41. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R.: 'Albert: A lite bert for self-supervised learning of language representations', arXiv preprint arXiv:1909.11942, 2019
42. Pires, T., Schlinger, E., and Garrette, D.: 'How multilingual is multilingual BERT?', arXiv preprint arXiv:1906.01502, 2019
43. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V.: 'Roberta: A robustly optimized bert pretraining approach', arXiv preprint arXiv:1907.11692, 2019
44. Aßenmacher, M., and Heumann, C.: 'On the comparability of pre-trained language models', arXiv preprint arXiv:2001.00781, 2020
45. Hu J, Ruder S, Siddhant A, Neubig G, Firat O, Johnson M. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. InInternational Conference on Machine Learning 2020 Nov 21, pp. 4411-4421
46. Howard, J., and Ruder, S.: 'Universal language model fine-tuning for text classification', arXiv preprint arXiv:1801.06146, 2018
47. Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., and Gurevych, I.: 'AdapterFusion: Non-destructive task composition for transfer learning', arXiv preprint arXiv:2005.00247, 2020
48. Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., Cho, K., and Gurevych, I.: 'Adapterhub: A framework for adapting transformers', arXiv preprint arXiv:2007.07779, 2020
49. Kim, S., Shum, A., Susanj, N., and Hilgart, J.: 'Revisiting pretraining with adapters', in Editor (Ed.)^(Eds.): 'Book Revisiting pretraining with adapters' (2021, edn.), pp. 90-99
50. https://huggingface.co/docs/transformers/index
51. Priyadharshini R, Chakravarthi BR, Navaneethakrishnan SC, Durairaj T, Subramanian M, Shanmugavadivel K, Hegde SU, Kumaresan PK. Findings of the shared task on Abusive Comment Detection in Tamil. InProceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages. Association for Computational Linguistics 2022 May.
52. Prasanth SN, Raj RA, Adhithan P, Premjith B, Kp S. CEN-Tamil@ DravidianLangTech-ACL2022: Abusive Comment detection in Tamil using TF-IDF and Random Kitchen Sink Algorithm.

InProceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages 2022 May, pp. 70-74

53. Patankar, S., Gokhale, O., Litake, O., Mandke, A., and Kadam, D.: 'Optimize_Prime@ DravidianLangTech-ACL2022: Abusive Comment Detection in Tamil', arXiv preprint arXiv:2204.09675, 2022
54. Pahwa B. Bphigh@ tamilnlp-acl2022: Augmentation strategies for indic transformer-based abusive comment detection in tamil. InProceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages. Association for Computational Linguistics 2022.
55. Hossain A, Bishal M, Hossain E, Sharif O, Hoque MM. COMBATANT@ TamilNLP-ACL2022: Fine-grained Categorization of Abusive Comments using Logistic Regression. InProceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages 2022 May, pp. 221-228
56. Palanikumar V, Benhur S, Hande A, Chakravarthi BR. DE-ABUSE@ TamilNLP-ACL 2022: Transliteration as Data Augmentation for Abuse Detection in Tamil. InProceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages 2022 May, pp. 33-38
57. Rajalakshmi R, Duraphe A, Shibani A. DLRG@ DravidianLangTech-ACL2022: Abusive Comment Detection in Tamil using Multilingual Transformer Models. InProceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages 2022 May, pp. 207-213
58. Prasad G, Prasad J, Gunavathi C. GJG@ TamilNLP-ACL2022: Using Transformers for Abusive Comment Classification in Tamil. InProceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages 2022 May, pp. 93-99.
59. Balouchzahi F, Gowda A, Shashirekha H, Sidorov G. MUCIC@ TamilNLP-ACL2022: Abusive Comment Detection in Tamil Language using 1D Conv-LSTM. InProceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages 2022 May (pp. 64-69), pp. 64-69
60. Bharathi B, Varsha J. SSNCSE NLP@ TamilNLP-ACL2022: Transformer based approach for detection of abusive comment for Tamil language. InProceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages 2022 May, pp. 158-164

Figures

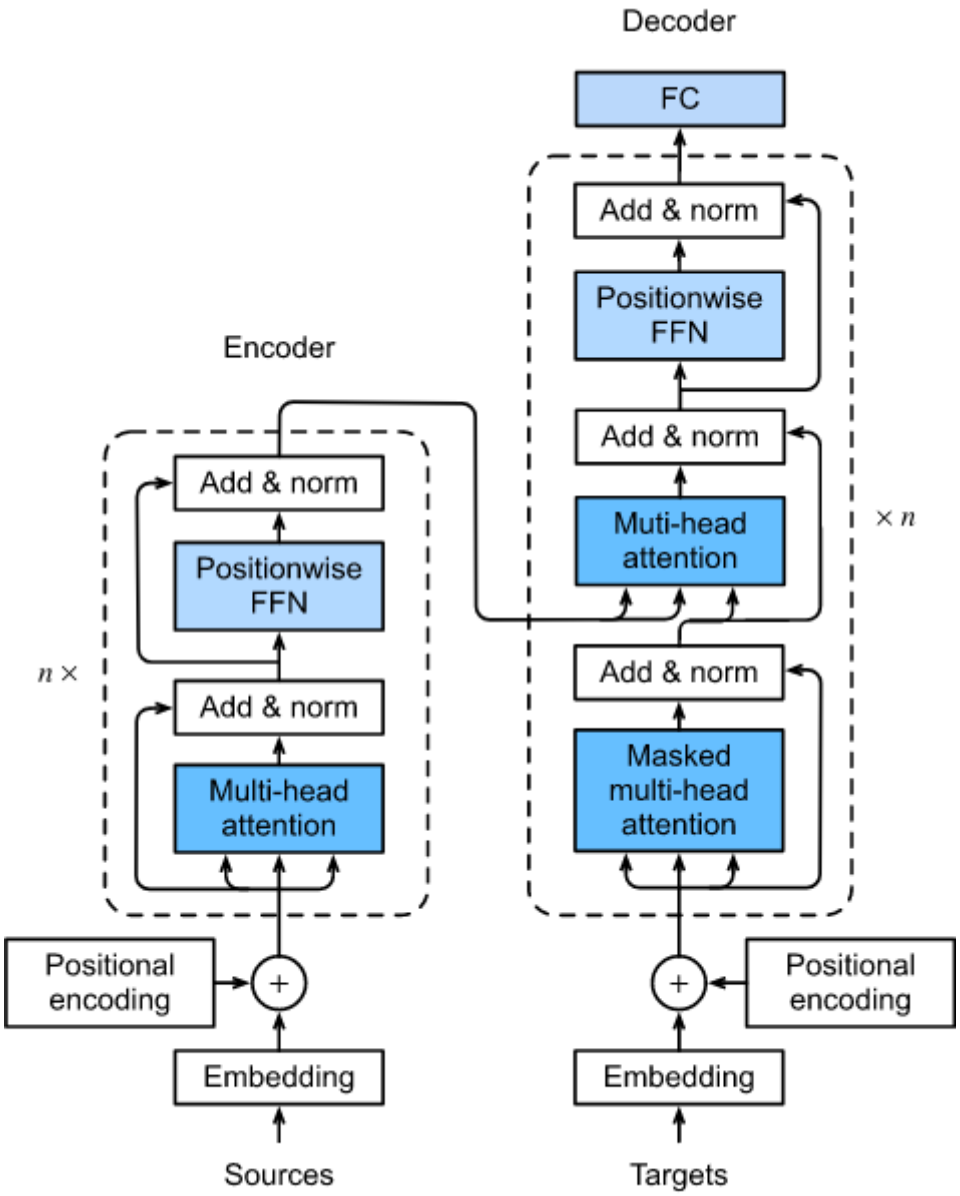


Figure 1

Transformer Architecture

(Adapted from: https://arxiv.org/pdf/1706.03762.pdf)

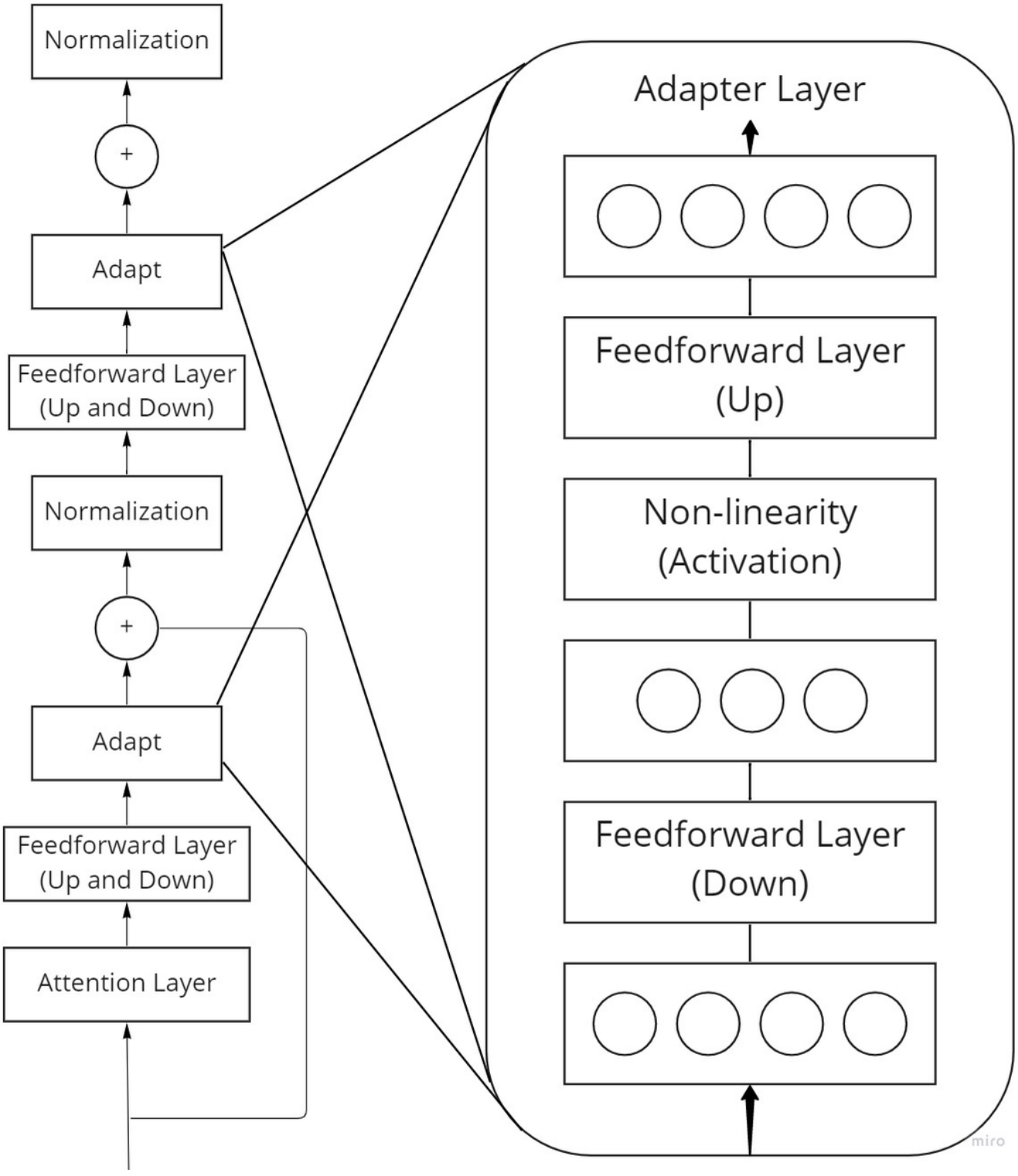


Figure 2

Integration of Adapter into Encoder part of a Transformer

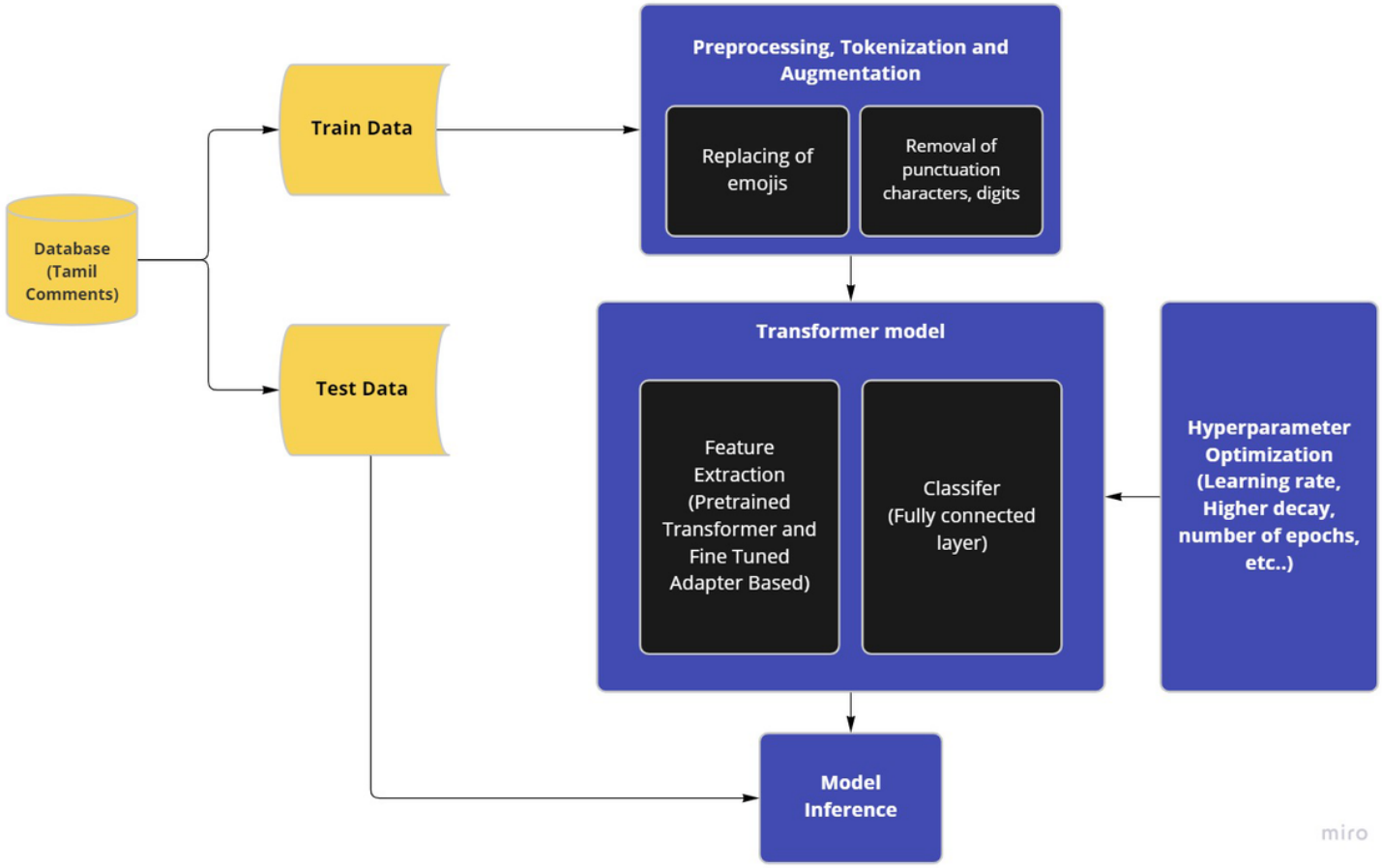


Figure 3

Proposed workflow

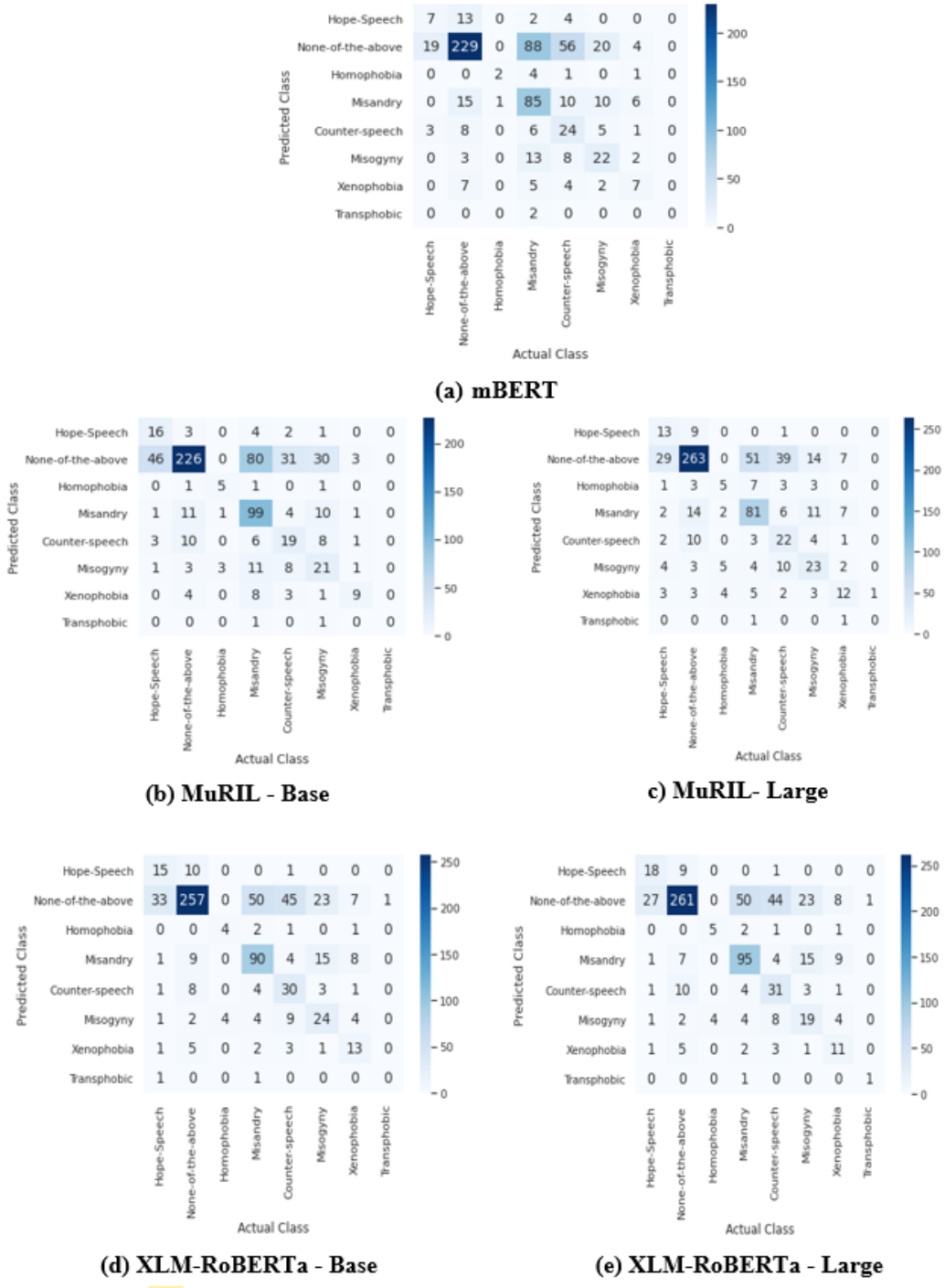
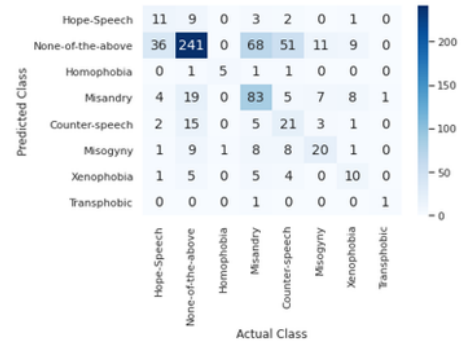
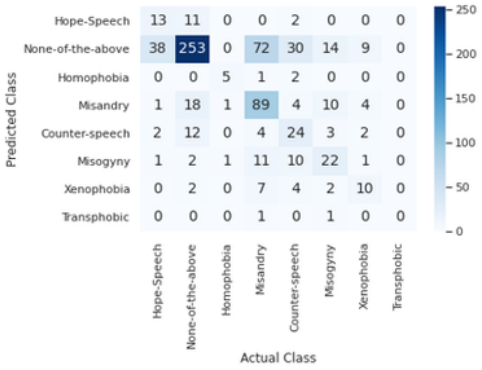


Figure 4

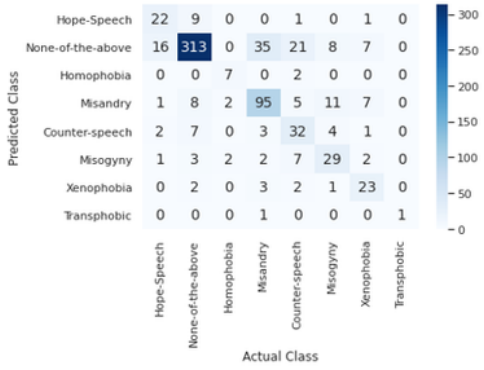
Confusion matrices for fine-tuned transformer models



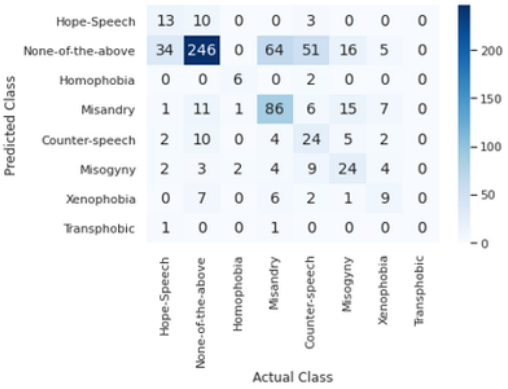
(a) mBERT



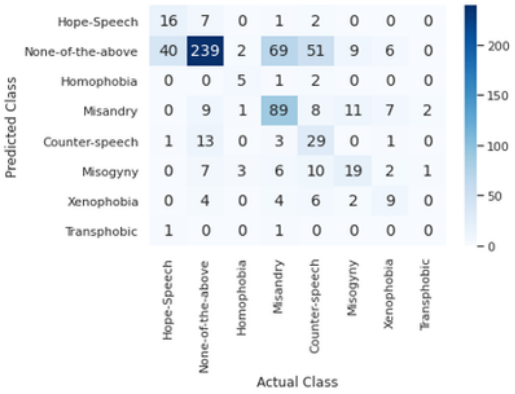
(b) MuRIL - Base



c) MuRIL- Large



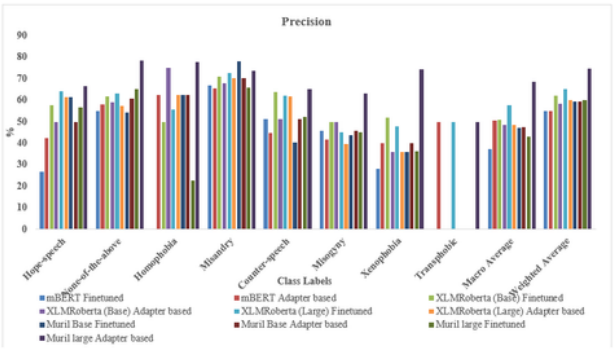
(d) XLM-RoBERTa - Base

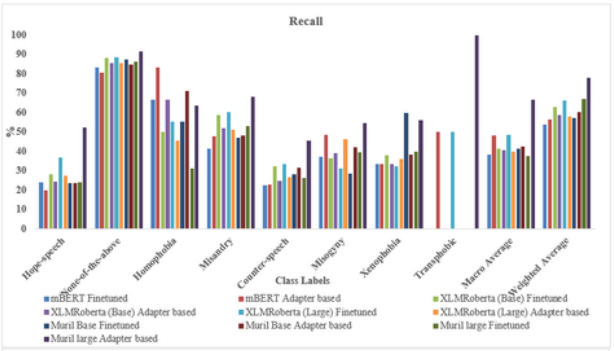


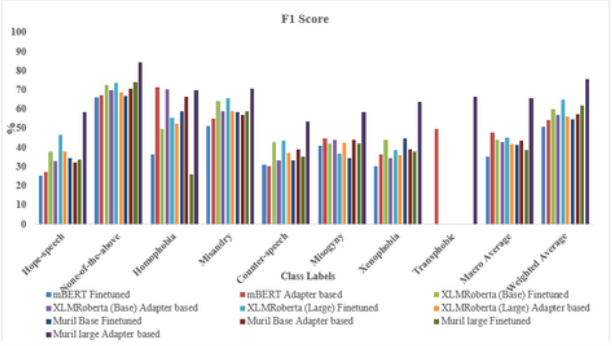
(e) XLM-RoBERTa - Large

Figure 5

Confusion matrices for adapter-based transformer models







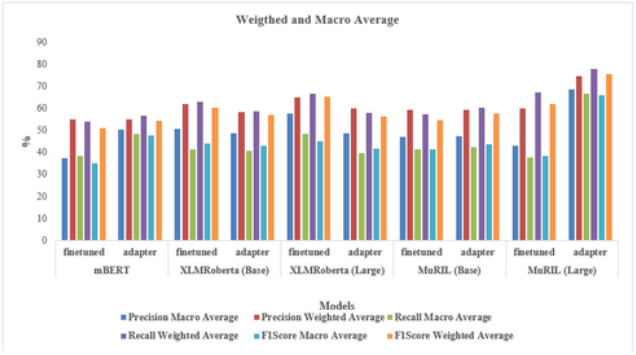


Figure 6

Finetuned vs Adapter based Transformer models