Highlights

Offensive Language Identification in Low-resourced Code-mixed Dravidian languages using Pseudo-labeling

Adeep Hande,Karthik Puranik,Konthala Yasaswini,Ruba Priyadharshini,Sajeetha Thavareesan,Anbukkarasi Sampath,Kogilavani Shanmugavadivel,Durairaj Thenmozhi,Bharathi Raja Chakravarthi

- Our work intends to identify offensive language on user-generated comments from social media applications such as YouTube in the code-mixed Dravidian languages of Kannada, Malayalam, and Tamil.
- Proposing an approach to increase the training data by generating labels using pseudo-labeling and Fine-tuning several pretrained language models.
- We experiment with multilingual languages models separately on the primary dataset, the transliterated dataset, and the newly constructed combination of both the datasets to examine if an increase in training data would improve the overall performance of the language models.
- We have shown that our approach yields the best-weighted F1-Scores on all three languages concerning its counterparts

Offensive Language Identification in Low-resourced Code-mixed Dravidian languages using Pseudo-labeling

Adeep Hande[a,*], Karthik Puranik[a,*], Konthala Yasaswini[a], Ruba Priyadharshini[b], Sajeetha Thavareesan[c], Anbukkarasi Sampath[d], Kogilavani Shanmugavadivel[d], Durairaj Thenmozhi[e] and Bharathi Raja Chakravarthi[f,**]

[a]*Indian Institute of Information Technology Tiruchirappalli, Tamil Nadu, India*
[b]*ULTRA Arts and Science College, Madurai, Tamil Nadu, India*
[c]*Eastern University, Sri Lanka*
[d]*Kongu Engineering College, Erode, Tamil Nadu, India*
[e]*Sri Sivasubramaniya Nadar College of Engineering, Tamil Nadu, India*
[f]*Insight SFI Research Centre for Data Analytics, National University of Ireland Galway, Galway, Ireland*

ARTICLE INFO

Keywords:
Offensive language Identification
Dravidian languages
Code-mixing
Pseudo-labeling

ABSTRACT

Social media has effectively become the prime hub of communication and digital marketing. As these platforms enable the free manifestation of thoughts and facts in text, images and video, there is an extensive need to screen them to protect individuals and groups from offensive content targeted at them. Our work intends to classify code-mixed social media comments/posts in the Dravidian languages of Tamil, Kannada, and Malayalam. We intend to improve offensive language identification by generating pseudo-labels on the dataset. A custom dataset is constructed by transliterating all the code-mixed texts into the respective Dravidian language, either Kannada, Malayalam, or Tamil and then generating pseudo-labels for the transliterated dataset. The two datasets are combined using the generated pseudo-labels to create a custom dataset called CM-TRA. As Dravidian languages are under-resourced, our approach increases the amount of training data for the language models. We fine-tune several recent pretrained language models on the newly constructed dataset. We extract the pretrained language embeddings and pass them onto recurrent neural networks. We observe that fine-tuning ULMFiT on the custom dataset yields the best results on the code-mixed test sets of all three languages. Our approach yields the best results among the benchmarked models on Tamil-English, achieving a weighted F1-Score of 0.7934 while scoring competitive weighted F1-Scores of 0.9624 and 0.7306 on the code-mixed test sets of Malayalam-English and Kannada-English, respectively. The data and codes for the approaches discussed in our work have been released[1].

1. Introduction

Social media has become a popular contrivance of communication in the 21st century and is the "democratisation of information" by converting people into publishers from the conventional readers (Nasir Ansari et al., 2018). 53.6% of the world's population use social media (Chaffey, 2021) which comprises a vivid structure between users from various backgrounds (Kapoor et al., 2018). With its free expressing environment, it witnesses much content, including images, videos and comments from various age groups belonging to diverse regions, languages and interests. While the basic idea of social media remains to be communication and entertainment, users are seen using rude and defamatory language to express their views. Users might not appreciate such comments or posts and might be influential on teenagers. Offensive posts targeted on a group or an individual can lead to frustration, depression and distress (Kawate and Patil,

[1] https://github.com/adeepH/Dravidian-OLI

*Equal Contribution
**Corresponding Author
adeeph18c@iiitt.ac.in (A. Hande); karthikp18c@iiitt.ac.in (K. Puranik); konthalay18c@iiitt.ac.in (K. Yasaswini); rubapriyadharshini.a@gmail.com (R. Priyadharshini); sajeethas@esn.ac.lk (S. Thavareesan); anbu.1318@gmail.com (A. Sampath); kogilavani.sv@gmail.com (K. Shanmugavadivel); theni_d@ssn.edu.in (D. Thenmozhi); bharathi.raja@insight-centre.org (B.R. Chakravarthi)
ORCID(s): 0000-0002-2003-4836 (A. Hande); 0000-0002-5536-2258 (K. Puranik); 0000-0001-5845-4759 (K. Yasaswini); 0000-0003-2323-1701 (R. Priyadharshini); 0000-0002-6252-5393 (S. Thavareesan); 0000-0003-0226-8150 (A. Sampath); 0000-0002-0715-143X (K. Shanmugavadivel); 0000-0003-0681-6628 (D. Thenmozhi); 0000-0002-4575-7934 (B.R. Chakravarthi)

2017; Puranik et al., 2021). Researchers recognised the need to detect and remove offensive content from social media platforms for a long period. However, there were a few challenges faced in this field. Though automating this process with the help of supervised machine learning models gave better accuracy than human moderators (Zampieri et al., 2020), the latter were preferred as they could justify their decision in removing the comment/post from the platform (Risch et al., 2020). Secondly, most of the comments and posts made were in code-mixed under-resourced languages (Chakravarthi et al., 2020a). There was an absence of enough datasets and tools to produce state-of-the-art results to be implemented in these platforms. Our paper presents some unique approaches to give excellent F1 scores for code-mixed Dravidian languages, mainly Tamil, Malayalam, and Kannada.

Social media creates a whole new opportunity in the field of research. Non-English speakers tend to use phonetic typing, Roman scripts, transliteration, code-mixing and mixing several languages instead of Unicode[1]. Code-mixed sentences for Dravidian languages can be Inter-sentential which consists of pure Dravidian languages written in Latin script, Code-switching at a morphological level when it is written in both Latin and the Dravidian language and an Intra-sentential mix of English and the Dravidian language written in Latin script (Yasaswini et al., 2021). The foremost step in analysing code-mixed or code-switched data includes language tagging, which, if not accurate, can affect the results of other tasks. Language tagging has evolved over the years but is not yet satisfactory for analysing code-mixed data (Mandal and Singh, 2018). The past years have seen enormous encouragement and research in code-mixing of under-resourced languages due to over-fitting. One of the main issues of dealing with code-mixed languages are the lack of annotated datasets and languages models being pretrained on code-mixed texts. The lack of code-mixed data resulted in constriction of data crisis, which affected the performance of various tasks. Transliteration of code-mixed data can increase the size of the input dataset. Transliteration refers to converting a word from one language to another while protecting the semantic meaning of the utterance and obeying the syntactic structure in the target language. The pronunciation of the source word is maintained as much as possible. While trying to get the critical features from a text or translating from one language to another, some language pairs like English/Spanish might not encounter any issue as *Por favor* is written as *Por favor* in English as they share the same Latin script. However, performing such tasks on Dravidian languages might pose problems, and transliteration can solve them to an extent (Knight and Graehl, 1997). Supervised learning on small datasets or languages with limited resources can be complex. Thus, pseudo-labeling (Lee, 2013) can be employed to increase the performance considerably. In pseudo-labeling, the model is trained on labeled data to predict labels for a batch of unlabeled data. The predictions are then fed into the model as pseudo labelled data.

1.1. Research Questions

In this paper, we attempt to address the following research questions:

1. *What architectures can be employed for effective cross-lingual knowledge transfer among code-mixed languages for offensive language identification?*
   We evaluate several recent approaches for offensive language identification, primarily focusing on cross-lingual transfer due to the persistence of code-mixed instances in the dataset. We have also used several state-of-the-art pretrained multilingual language models for three languages, Kannada, Malayalam, and Tamil.
2. *How do we break the curse of the lack of data for under-resourced Dravidian languages?*
   To overcome the barrier of lack of data, we revisit pseudo-labeling. We transliterate the dataset to the respective Dravidian language for our multilingual dataset and generate labels by using approaches such as pseudo-labeling. We combine the two datasets to form a larger dataset.

1.2. Contribution

1. We propose an approach to improve offensive language identification by emphasising more on constructing a bigger dataset, generating the pseudo-labels on the transliterated dataset, and combining the latter with the former to have extensive amounts of data for training.
2. We experiment with multilingual languages models separately on the primary dataset, the transliterated dataset, and the newly constructed combination of both the datasets to examine if an increase in training data would improve the overall performance of the language models. We observe that this approach yields the best-weighted F1-Scores on all three languages concerning its counterparts.
3. We have shown that our method works for three under-resourced languages, namely Kannada, Malayalam and Tamil, in a code-mixed setting. We also have compared our approaches to all other models that have been benchmarked on the datasets.

[1] https://amitavadas.com/Code-Mixing.html

The rest of our work is organised as follows. Section 2 talks about the related work on offensive language identification, while Section 3 entails a discussion about the Dravidian languages and their histories. Section 4 introduces the dataset used for the task at hand. Section 5 discusses the several models and approaches to test their fidelity on the original code-mixed dataset, pseudo-labels procured for the transliterated data and the combination of the both. Section 6 comprises a detailed analysis concerning the behaviour and results of the pretrained models when fine-tuned on code-mixed and transliterated data, and the results are compared with other approaches (Chakravarthi et al., 2021a) from the shared task conducted by DravidianLangTech-2021[2] at EACL 2021. We also perform the error analysis on the Kannada and Tamil predictions. Finally, Section 7 concludes our work and talks about potential directions for future work on Offensive Language Identification in Dravidian languages.

2. Related Work

Recent years have witnessed a surge in the usage of offensive language on social media platforms. This surge has resulted in several corporations and organisations developing automated systems that filter offensive language on their platforms. In light of recent research on Dravidian languages, especially Dravidian code mixed text, a shared task for sentiment analysis of YouTube comments in Dravidian code mixed text has been presented (Mandl et al., 2020). The escalation of offensive language in social media has also resulted in some researchers conducting multi-disciplinary studies on the impact of offensive language on the mental health of its users (Chen et al., 2012a; Xu and Zhu, 2010). Early approaches to detect Offensive Language was very reliant on feature engineering applied to traditional machine learning classifiers (Dadvar et al., 2013). More recent approaches to offensive language detection were hinging on Recurrent Neural Networks (RNNs) such as LSTMs, GRUs (Pitsilis et al., 2018) while employing word embeddings to RNNs to surpass the performance of traditional machine learning classifiers (Bisht et al., 2020). However, the success of transformer architecture led to its adaptation as the building blocks of encoder-decoder models, replacing the conventional Recurrent Neural Networks (Vaswani et al., 2017). The success of pretraining has resulted in a pragmatic shift towards transfer learning for offensive language detection (Ruder et al., 2019; Liu et al., 2019a). Offensive language detection is essentially treated as a sentence classification task (Risch et al., 2020). Researchers have proposed an automated flame detection system that extracts features at a conceptual level to detect offensive language (Razavi et al., 2010). The terms hate speech, and offensive language is often misunderstood to mean the same thing, as hate speech is directed towards an entity. In contrast, offensive language is considered to be abusive and derogatory (Chaudhari et al., 2019).

Several researchers have worked on developing systems to detect the offensive language in English, Turkish, Greek, Danish, and Arabic (Zampieri et al., 2020). (Felbo et al., 2017) introduced the 'deepmoji' approach to elevate offensive language detection for English social media texts. This approach mainly depends on pretrain a neural network model for offensive language classification by utilizing emojis as weakly supervised training labels. In the work of (Chen et al., 2012b), the authors suggested a Lexical Syntactic Feature architecture to strike a balance between identifying offensive content and possible offensive users in social media, arguing that, although current approaches consider messages as separate instances, the emphasis should be on the content's source. A topic-based mixture model integrated into the framework of semi-supervised training that takes advantage of a large amount of un-annotated Twitter data to detect offensive tweets was employed (Xiang and Zhou, 2014). The authors of (Xiang et al., 2012) concentrated on Twitter and proposed a semi-supervised method along with statistical subject modelling for detection of offensive content.

Toxic material is on the increase as digital knowledge becomes more widely available. As a result, detecting this form of language is extremely important. To overcome this issue, a combination of a state-of-the-art pretrained language model (CharacterBERT) and a conventional bag-of-words technique was used(Karimi et al., 2021). Modern data science tools translate raw text into key features from which preprocessing or learning algorithms can make predictions for evaluating offensive communications, enabling for the identification and classification of toxic online comments (Noever, 2018). Before the emergence of toxic text analysis, (Warner and Hirschberg, 2012a) modelled hate speech as a word sense disambiguation issue in which SVM was used to classify data. To detect toxicity in social media messages, a technique based on a simple and powerful presumption: *A post is at least as toxic as its most toxic span* (Xiang et al., 2021) was presented to boost the interpretability of transformers-based models like BERT and ELECTRA. The BiLSTM-CRF model was combined with Toxic Bert Classification to train the detection model (Luu and Nguyen, 2021) for recognizing toxic words from articles. The authors of (Brassard-Gourdeau and Khoury, 2019) started by developing a sentiment detection tool that used a variety of lexicons and features such as word frequencies and negations,

[2] https://dravidianlangtech.github.io/2021/

which proved that there is a strong correlation between sentiment and toxicity using this method. Later, they integrated sentiment data into a toxicity detection neural network and showed improved detection accuracy.

Hate speech is a form of offensive language that employs stereotypes to convey a hateful philosophy (Warner and Hirschberg, 2012b). *Hate speech detection* is a closed-loop mechanism in which people know what is going on and deliberately avoid being detected. One of the challenges of automatic hate speech detection systems is the shifting of attitudes towards topics and historical context (MacAvaney et al., 2019). In the context of automated detection studies, offensive and abusive language are both used as overarching words for harmful content. Offensive language has a broader reach, and hope speech falls under each of these categories (Hande et al., 2021b; Yin and Zubiaga, 2021). The strong relationship between hate speech and actual hate crimes highlight the significance of identifying and moderating hate speech. Early detection of users who spread hate speech may lead to outreach efforts aimed at preventing the transition from speech to action (Waseem and Hovy, 2016). The authors of (Malmasi and Zampieri, 2017) used standard linguistic features and a linear SVM classifier to establish a baseline for discriminating between hate speech and profanity. *Dialect* and *race priming* was introduced, and conclusive proof that special consideration should be given to the conflicting effects of dialect in hate speech identification to prevent unintentional racial bias was found by the authors of (Sap et al., 2019). The authors of (Klamma et al., 2006) developed a pattern mining tool, PALADIN, to detect anti-social behaviours of users. PALADIN has demonstrated a method for information exploration, focusing on the disturbance in digital social networks using patterns. Work in (Burnap and Williams, 2014) describes a supervised machine learning text classifier that was trained and tested to differentiate between hateful and antagonistic responses based on race, ethnicity, and religion. By comparing classification results obtained from training on expert and amateur annotations, the authors of (Waseem, 2016) investigated the impact of annotator knowledge of hate speech on classification models. Numerous challenges relate to detecting negative online behaviour, including identification that goes beyond simply recognizing offensive terms. Unsvåg and Gambäck studied the potential and effects of including user features in hate speech classification, with an emphasis on Twitter, to close this distance.

To tackle Offensive language identification in Dravidian languages, several manually annotated datasets were constructed for Tamil (Chakravarthi et al., 2020b), Malayalam (Chakravarthi et al., 2020a), and Kannada (Hande et al., 2020) for sentiment analysis and offensive language identification. (Hande et al., 2020) scraped the datasets from the comments section of the YouTube videos. (Hande et al., 2021a) benchmarked multi-task learning of supplemental tasks in under-resourced Dravidian languages. In multilingual countries like India, where the speakers are polyglots, it is evident that one would find the presence of code-mixed sentences, as the videos were scraped from social media. Chakravarthi et al. carried a shared task on Offensive language identification in Dravidian languages of Tamil, Malayalam, and Tamil for the user-generated comments. Chakravarthi et al. used DravidianCodeMix[3], a multilingual code-mixed dataset manually annotated for sentiment analysis and offensive language identification (Chakravarthi et al., 2021c). The dataset consists of around 44,000 comments in code-mixed Tamil-English, 20,000 comments in Malayalam-English, and 7,700 comments in Kannada-English. The authors set the baselines using primitive machine learning algorithms for the datasets (Chakravarthi et al., 2021c), having a baseline weighted F1-Score of 0.65, 0.75, and 0.66 for the languages, in the said order. Jayanthi and Gupta devised an approach of task-adaptive pretraining multilingual BERT for offensive language identification that achieved the best-weighted F1-Score of 0.75 for Kannada. Saha et al. pretrain XLM-RoBERTa from scratch, and developed the ensemble (Hande et al., 2021c) of three models; Convolutional Neural Network (CNN), fine-tuned XLM-RoBERTa, and the custom pretrained XLM-R, scoring a weighted F1-Score of 0.78 and 0.97 in Tamil and Malayalam respectively, achieving the best results.

Majority of the works on offensive language identification focus more on the aspect of the model improvements. In this paper, we propose an approach that aims to address the lack of annotated data for low-resourced Dravidian languages, by transliterating the existing code-mixed dataset CM-TRA to their respective languages, and generating labels on them using pseudo-labeling. This approach results in effective cross-lingual transfer when fine-tuning multilingual language models. We fine-tune recent state-of-the-art pretrained language models that are very effective for cross-lingual transfer (Kalyan et al., 2021). We conduct sufficient experiments on all three different datasets (DravidianCodeMix, Transliterated-DravidianCodeMix, and Custom dataset CM-TRA), and our experimental results indicate that most of the fine-tuned language models fare relatively better than its results on the former datasets, with ULMFiT yielding the best results in all three languages.

[3] https://zenodo.org/record/4750858/

3. Dravidian Languages

The Dravidian languages comprise about 80 types, and are spoken in and around South Asia, mainly in southern and central India and countries like Singapore and Sri Lanka (Krishnamurti, 2003; Chakravarthi et al., 2021b). These languages flourished from the Dravidian civilization of the Indus Valley civilization around 4500 years ago(Chen and Kong, 2021a). The first signs of Dravidian languages are affirmed as Tamil-Brahmi scripts on the walls of caves in Madurai and Tirunelveli districts of Tamil Nadu, India, in the 2nd century BCE. While Tamil remains the oldest language in India, Telugu, Kannada, Malayalam and other Dravidian languages are prominently spoken by over 21% of India's population[4]. Tamil, the official language of the Indian state of Tamil Nadu, the Union Territory Puducherry and the nations of Sri Lanka and Singapore, is known to be one of the few most extended surviving languages in the world (Stein, 1977). The oldest literature among the Dravidian languages, the Sangam literature, was discovered in Tamil over 2000 years ago (Abraham, 2003). Linguists claim that Tamil is derived from Proto-Dravidian as there have been shreds of evidence of Tamil written in Brahmi script inscribed on rocks and caves around the 2nd century BC. Even today, 55% of the inscriptions discovered are found in Tamil language[5] with records also discovered in Sri Lanka, Egypt and Thailand. Popular to the contrary beliefs, The Tamil writing system has its writing systems and is not Abugida, Abjad, nor Alphabet system. Tamil has vowels, pure consonants, uyirmey and Aytam. (strictly speaking, kuRRiyalukaram and KuRRiyalikaram). Unlike any system globally, Tamil is the only language that defines the length (duration) of a 'letter', it Tamil it is called ezuttu. Furthermore, it does not recognize any pure consonant. In Abugida, it is "obligatory: to end in Vowel. No Aytam either. Tamil system should be referred to as Tamil System where ezuttu is defined (unlike any other system, even aksara does not specify the length of any rules as to how many consonants can cluster to combine with a vowel). Kannada, another Dravidian language, is spoken mainly in Karnataka and the southwestern regions of India. Written in Kadamba script, Kannada was used by prominent dynasties like Chalukya, Rashtrakuta, Vijayanagara and Hoysala. With a history of over 2500 years, Kannada is said to have subtle influences from Sanskrit, Prakrit and Pali (Steever, 2018). Kannada is historically classified into Old Kannada (450–1200 CE), Middle Kannada (1200–1700 CE), and Modern Kannada (1700 CE–present) (Rice, 1982; Narasimhachar, 1988) and is presently spoken by over 56.9 million people. Malayalam, the official language in the Indian state of Kerala and the Union Territories Lakshadweep and Puducherry (Mahé), is said to be derived from Tamil directly and separated at around 9th century CE or from the Proto-Dravidian from which Tamil has also originated. Vatteluttu script is currently used to write Malayalam is descended from Grantha script and is similar to the Tigalari script (Sekhar, 1951; Asher, 2013; George, 1972). Various literary works from the period of 9th to 11th centuries have been found, (Bright, 1999) of which *Ramacharitam* is said to be the earliest. Currently spoken by over 35 million people, it is one of the major Dravidian languages.

4. Dataset

We use the offensive language data from DravidianCodeMix (Chakravarthi et al., 2021c, 2020b,a; Hande et al., 2020). The data comprises of various code-mixed comments on movie trailers on YouTube in Tamil, Malayalam and Kannada languages. The dataset is divided into training, development, and test sets with similar distribution for the three languages. Each set contains five different labels for the Malayalam dataset, while Tamil and Kannada datasets have six different types of labels, including the "Offensive Targeted Insult Other" label. We can observe an enormous class imbalance in the dataset, with "Not offensive" occupying a major share and "Offensive Targeted Insult Other" having a negligible amount of sample data for all three languages. The distribution of DravidianCodemix is tabulated in Table 1. The class-wise distribution of the training and test set are tabulated in Table 2 and Table 3. The six different classes include(Chakravarthi et al., 2021a):

- **Not-Offensive (NO)**: Comments/post which is not impolite, rude and does not have obscenity, swearing, or profanity.
- **Offensive-Targeted-Insult-Individual (OTI)**:Comments/ posts which are offensive and targeted at a particular person.
- **Offensive-Targeted-Insult-Group (OTG)**: Comments/ posts which are offensive and targeted at a group of individuals or community.

[4] https://en.Wikipedia.org/wiki/Dravidian_languages

[5] https://web.archive.org/web/20060518064346/http://www.hindu.com/2005/11/22/stories/2005112215970400.htm

- **Offensive-Targeted-Insult-Other (OTO)**: Comments/ posts which are offensive but doesn't belong to any of the above two labels.
- **Offensive-Untargeted (OU)**: Comments/ posts which are offensive but not targeting anyone.
- **Other Language (OL)**: Comments/ posts are not in the intended language.

Table 1
Train-Development-Test Data Distribution

Split	Kannada	Malayalam	Tamil
Training	6,217	16,010	35,129
Development	777	1,999	4,388
Test	778	2,001	4392
Total	7,772	20,010	43,909

Table 2
Class-wise distribution of the training set.

Languages/Classes	NO	OL	OTI	OTG	OTO	OU	Total
Tamil	25,415 (72.3%)	1,454 (4.1%)	2,343 (6.7%)	2,557 (7.3%)	454 (1.3%)	2,906 (8.3%)	35,129
Malayalam	14,153 (88.4%)	1,287 (8.0%)	239 (1.5%)	140 (0.9%)	- (0%)	191 (1.2%)	16,010
Kannada	3,544 (57%)	1,522 (24.5%)	487 (7.8%)	329 (5.3%)	123 (2.0%)	212 (3.4%)	6,217

Table 3
Class-wise distribution of the test set

Languages/Classes	NO	OL	OTI	OTG	OTO	OU	Total
Tamil	3,190 (72.6%)	165 (3.8%)	315 (7.2%)	288 (6.6%)	71 (1.6%)	368 (8.4%)	4,392
Malayalam	1,765 (88.2%)	157 (7.9%)	27 (1.4%)	23 (1.3%)	- (0%)	29 (1.2%)	2,001
Kannada	417 (54.3%)	185 (24.1%)	75 (9.8%)	44 (5.7%)	14 (1.8%)	33 (4.3%)	768

4.1. Code Mixing

Multilingual speakers have a general trend of using distinct utterances from different languages, referred to as code-mixing. Code-mixing refers to the idea that a speaker switches from one language or variety to another in a text or a discussion. It is a prevalent phenomenon in a multilingual community (Jose et al., 2020). Moreover, users like to blend various languages in their online platform interactions. There are different types of code-mixing in a language. We have given the examples of code-mixing in our corpora in Figure 1, Figure 2, and Figure 3.

Code switching Type	EXAMPLE	Translation
No-code-mixing: Only Kannada (written in Kannada Script only)	ತುಂಬಾ ವರ್ಷದ ಹಿಂದೆ ಕೇಳಿದ್ದೆ. ಈಗ ಸಿಕ್ಕಿದ್ದು ಸಕತ್ ಖುಷಿ ಆಗಿದೆ	I had heard these years back; I am very ecstatic to find them now.
Inter-sentential code-mixing: Mix of English and Kannada (Kannada, written in Kannada script only)	Sister ಹಾಗೆಲ್ಲ ಮಾಡಲ್ಲ ನಾವು ಯಾರಾದರೂ ತಪ್ಪು ಮಾಡ್ತಿದ್ರೇರೆ ಅಂದಾಗ ಅವರನ್ನ roast ಮಾಡ್ತಿವಿ ಅಷ್ಟೇ. ಅದು entertainment ಗೆ ಅಷ್ಟೇ. ತುಂಬಾ ಧನ್ಯದಗಳು ಹೀಗೆ support ಮಾಡ್ತೀರಿ	We don't do that here sister, we only roast people if they commit some mistakes. It is solely for entertainment purposes, not to hurt others' feelings. Keep supporting us.
Only Kannada (written in Latin Script)	Namma deshanu china thara aitu andre badatanane erala sir	If our country becomes like China, poverty ceases to exist, sir.
Code-switching at a morphological level: (written in both Kannada and Latin script	ಈ ದರಿದ್ರ ಬಡ್ಡಿ*ತವ tiktok ನೋಡಿಯೆ ನಮ್ಮ ಮೋದಿಯವರು ಬ್ಯಾನ್ ಮಾಡಿದ್ದು ಗುರು	After having a look at this dreadful app tiktok, Modi banned it mate.
Intra-sentential mix of English and Kannada (written in Latin Script only)	Estella matadonu elli matadodkintta border ge hogi matado maraya	If your only intention is to babble, go do the same near the borders mate.
Inter-sentential and intrasentential mix. (Kannada, written in both Latin and Kannada script)	ನಿಜವಾಗಿಯೂ ಅದ್ಭುತ heartly heltidini... plz avrigella namma nimmellara support beku	Truly remarkable, saying it from the bottom of my heart, please, all of us need to support them.

Figure 1: Example of code-mixing in the Kannada Dataset

Code-Switching Type	Example	Translation
No-code-mixing: Only Malayalam (Written in Malayalam Script only)	ഏറെ പ്രതീക്ഷ ഉള്ള ചിത്രം. കാരണം ഇതിന്റെ സംവിധായകൻ ആണ്.. പ്രിത്വി	A very promising film. Because he is the director of this. Prithvi
Inter-sentential code-mixing: Mix of English and Malayalam (Malayalam written in Malayalam script only)	I am really overwhelmed by the excitement of people of other languages... മലയാള സിനിമക്ക് അംഗീകാരം..	I am really overwhelmed by the excitement of people of other languages...Recognition for Malayalam movies.
Only Malayalam (Written in Latin script)	Ivde ippo varunnavarellam padam kandu varunnavarayirikum.	Whoever comes here now will be after watching the movie.
Code-switching in morphological level (written in both Malayalam and Latin script)	BGM മാത്രം കേൾക്കാൻ വേണ്ടി repeat അടിച്ചു കണ്ടു... Adipoli...	Watched this in repeat mode just to hear it's BGM... Awesome...
Intra-sentential mix of English and Malayalam (written in Latin Script only)	Take off nu shesham mattoru hit koodi.	Yet another hit after the movie Take off.
Inter-sentential and Intra-sentential mix. (Malayalam written in both Latin and Malayalam script)	Great theme and making! Orupadu per negative പറഞ്ഞിട്ടും എനിക്ക് nice ayitanu തോന്നിയത്..	Great theme and making! Though it had many negative reviews, I felt it was a nice one.

Figure 2: Example of code-mixing in the Malayalam Dataset

5. Methodology

5.1. Transformers

Transformers is a revolutionary architecture in the field of Natural Language Processing introduced in the paper Vaswani et al. (2017). sequence-to-sequence tasks can be solved while managing long-range dependencies. Sequence-aligned RNN's or convolutions are not used as Transformers mainly depends on self-attention. The basic idea is to address the input and output with an attention mechanism and deliberately avoid recurrences. It executes this with the help of encoders and decoders (Hande et al., 2021d). Attention takes in the input sequence and determines all the essential parts of the sequence.

The self-attention model lets the inputs interact with one another to find the representation of the sequence. For example, take the sentence "The car didn't cross the bridge as it was out of fuel". Humans might find it easy to interpret the sentence and map "it" with the car. However, that might not be the case with standard models. However, self-attention successfully maps them together. To calculate self-attention, we require query vector, key vector and

Code switching Type	EXAMPLE	Translation
No-code-mixing: Only Tamil (written in Tamil Script only)	நம்ம ஒரு ஏரியில் குதித்து விளையாடுவம்ல அது எங்க? அது மேல்தான் நடக்குரொம். செம்ம	We played in a lake right. Where it is; We are walking over it only. Marvellous.
Inter-sentential code-mixing: Mix of English and Tamil (Tamil, written in Tamil script only)	நல்ல தலைவி மாரி காட்டி - background ல ஏதோ plan போட்ருக்க மாரி தெரிதே.	Showing like a good leader but looks like there is a plan in the background.
Only Tamil (written in Latin Script)	Oruthar mela Viswasam kaatrathukkaga..innoruthara yen Asinga Paduthuringa??	Why are you humiliating other to show the gratitude on someone?
Code-switching at a morphological level: (written in both Tamil and Latin script	இப்பல்லாம் Trailer le வார நெறய சீன் படத்துல வரமாட்டேங்குது. அந்த வரிசையில் இதுவும்.	Now a days scene which appears at trailer is not there in movies. This is one such thing.
Intra-sentential mix of English and Tamil (written in Latin Script only)	Padatha paakura nammala psycho maathiruvaainga pola	They will make the movie watchers a psycho.
Inter-sentential and intrasentential mix. (Tamil, written in both Latin and Tamil script)	நாடோடிகள் பார்ட் 1 போல இந்த படமும் ஹிட் அடிக்கிறது kashtam தான் பா sorry	It is difficult for this movie to be hit like Naadodi Part 1. Sorry.

Figure 3: Example of code-mixing in the Tamil Dataset

value vector.

- Query vector (Q): the current word
  $Q = X * W_Q$.
- Key vector (K): indexing method for value vector
  $K = X * W_K$.
- Value vector (V): data present in input word
  $V = X * W_V$.

Where, W_Q, W_K, and W_V are the weight projections of Query, Key, and Value vectors respectively. Self-attention is parallelly and independently for each word in the Transformer's architecture. Once the outputs are concatenated and linearly transformed, we get the self-attention output for the required word. The process of calculating self-attention several times is referred to as multi-head attention.

$$MultiHead(Q, K, V) = concat(head_1, head_2, ..., head_n)W_O$$
$$where, head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

The equation represents the formula for multi-head attention.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

The equation 1 represents the formula for Self-attention for input matrices (Q, K, V).

Furthermore, a thoroughly linked feed-forward network is present in the encoder and decoder layers, added independently to each of them. Two linear transformations with Rectified Linear Unit (ReLU) activation is comprised in between them. Though the linear transformations remain the same, the parameters differ from layer to layer. The equation of the Feed-Forward Network (FFN) is as follows:

$$FFN(x) = \max(0, x * W_1, b_1)W_2 + b_2 \tag{2}$$

These linear transformation layers with the softmax function convert the output to predict the probabilities of the next token. The learned embeddings are employed to convert the input and output tokens into vectors with d_{model} dimensions.

Without RNNs and CNNs, the model should be made aware of the sequence of tokens and the position in a sequence. Positional encoding is added at the bottom of encoder stacks and decoder stacks of the input embeddings, and with dimension, d_{model} which is the same as the embeddings, the two can be summed with ease. We compute the output vector's word embeddings and feed it into a bidirectional LSTM layer (BiLSTM) as shown in Figure 4. The pooler embedding T_E as input into the block. The model has three gates, namely, input gate I_t, output gate O_t and forget gate F_t.The method of updating memory cell C_i and current latency values H_t is decided by three gates.For each node in LSTM, mathematical relationships between these gates are computed as follows:

- $I_t = \sigma(w_i.[h_{t-1}, T_E] + b_i)$
- $F_t = \sigma(w_f.[h_{t-1}, T_E] + b_f)$
- $O_t = \sigma(w_o.[h_{t-1}, T_E] + b_o)$
- $C_i = \tanh(w_c.[h_{t-1}, T_E] + b_c)$

5.1.1. IndicBERT

IndicBERT (Kakwani et al., 2020) is a multilingual ALBERT model that has been specifically trained on 12 major Indian languages. ALBERT was chosen as the base model because it has fewer parameters and is thus making it simpler to deploy and use in application areas. A single model was trained for multiple Indian languages to take advantage of their similarity. IndicBERT can support some of the under-represented languages. It is trained on IndicCorp and evaluated on IndicGLUE. One of the most extensive publicly accessible corpora for Indian languages is IndicCorp.

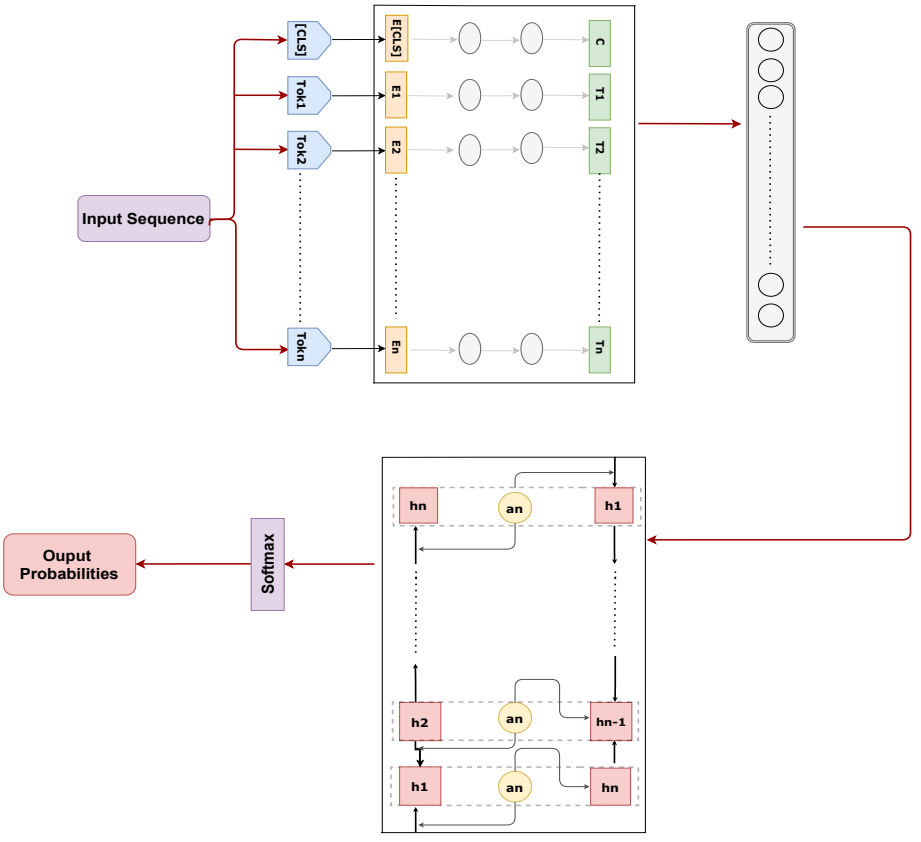


Figure 4: The architecture of feeding BERT-based Embeddings to a BiLSTM network.

It is created by exploring and filtering thousands of online outlets, mainly new books and magazines. To tokenize the sentences in each language, a sentencepiece tokenizer is trained with IndicCorp. This tokenized corpus trains a

Hyper-parameters	Characteristics
Number of LSTM Units	256
Loss	Cross Entropy
Epoch	5
Batch size	[16, 32, 64]
Optimiser	AdamW (Loshchilov and Hutter, 2019)
Dropout	0.4
Learning rate	2e-5
Max length	128

Table 4
Various hyper-parameters used for our experiments

multilingual ALBERT using the traditional masked language model (MLM). To give low-resource languages a more significant representation, exponentially smoothed data weighting is implemented across languages. A vocabulary of 200k to fit various scripts and massive vocabularies of Indic languages is selected. The Indic General Language Understanding Evaluation Benchmark, IndicGLUE, is a robust Natural Language Understanding (NLU) benchmark introduced to extensively evaluate language models on multiple Indian languages. It consists of 6 tasks. After pretraining, IndicBERT is fine-tuned on the specific task in IndicGLUE using the corresponding training sets. For Sentiment Analysis, the last layer's representation of the [CLS] token is fed to a linear classifier with a softmax layer to determine a probability distribution over the categories. It is important to note that IndicBERT is much smaller than two of the most potent performing multilingual models: mBERT and XLM-R base, despite being trained on larger Indic language corpora.

5.1.2. DistilBERT

Operating large models in peripheral computational training or inference resources remains difficult, as Transfer Learning from large-scale pre-trained models becomes more extensive in Natural Language Processing (NLP). Although models like BERT and XLM-RoBERTa have millions of parameters, increase performance substantially, training much greater models results in compelling performance on downstream areas. This shift presents a multitude of challenges. DistilBERT (Sanh et al., 2020) is a light transformer model trained by distilling BERT base. It is a method for pretraining a smaller language representation model. It has 40% lesser parameters than *bert-base-uncased* and runs 60%faster at the same time maintaining 97% of its language understanding capabilities. Using the DistilBERT model pretrained with knowledge distillation resulted in a similar performance on numerous downstream tasks.

Knowledge distillation is a compression method in which a smaller model - the student - is trained to emulate the behaviour of a larger model - the teacher or an ensemble of models. The student has the same architecture as BERT, but the token-type embeddings and the pooler are removed from the architecture, and the number of layers is reduced by a factor of 2. The student is initialised by taking one layer out of two from the teacher through the advantage of the shared dimensionality between their networks. The student is trained with the triple loss, which is a linear combination of the distilled loss L_{CE}, the supervised training loss, which is the masked language modelling loss L_{MLM} and a cosine embedding loss (L_{cos}) as it matches the student and teacher hidden states vectors. The triple loss is mathematically defined as :

$$L_{ce} = \sum_i t_i * log(s_i) \tag{3}$$

where t_i (resp. s_i) is a probability estimated by the teacher (resp. the student). This training loss leads to an upscale training signal by leveraging the complete teacher distribution.

5.1.3. ULMFiT

A language model (LM) determines the probability distribution over a sequence of words. When fine-tuned with a classifier, LMs encountered catastrophic forgetting and overfitted to small datasets. Due to the length of the vocabulary, statistical LMs suffer the effects of data sparsity. A novel approach, Universal Language Model Fine-tuning (ULMFiT) (Howard and Ruder, 2018) is introduced that solves these challenges and promotes efficient inductive transfer learning

for any NLP task. ULMFit pretrains a language model(LM) on a broad general-domain corpus and fine-tunes it on the target task using standard methods. It employs a single architecture and training process. No custom feature engineering or preprocessing is required in this model. It involves three stages:

1. **General-domain LM pretraining:**
   The language model(LM) is pretrained on WikiText (Merity et al., 2016) comprising of 28,595 preprocessed Wikipedia articles 103 million words to attain the same level of accuracy as computer vision models trained on ImageNet corpus. Although this stage is the most expensive, it can capture the general language properties. After that, the pre-trained model may be used for further NLP applications.
2. **Target task LM fine-tuning:**
   The target task dataset will have different distribution regardless of how varied the general-domain corpus is for pretraining the model. In this stage, the LM is fine-tuned on the target task dataset to learn its distinctions using discriminative fine-tuning and slanted triangular learning rates. Rather than utilising the same learning rate in the entire model, the authors propose to use different learning rates for each layer. It initially selects the learning rate of the last layer by fine-tuning only that layer and then applies the following formula to the lower layers -

   $$\eta^{I-1} = \frac{\eta^I}{2.6}, \text{where } \eta^I \text{ is the learning rate of the I-th layer.}$$

   Howard and Ruder introduced slanted triangular learning rate (STLR) to make the model parameters adapt to the task-specific text features. In this stage, the learning rate first increases linearly and then decays linearly.
3. **Target task classifier fine-tuning:** In the final stage of ULMFiT, the model is trained with two extra linear blocks. Each block uses batch normalization and dropout. The intermediate layer is activated with ReLU, while the final linear layer is activated with Softmax. The most important aspect of the transfer learning approach is fine-tuning the target classifier. Hence, Gradual unfreezing is used to fine-tune rather than training all layers at once, leading to catastrophic forgetting.

5.1.4. MuRIL

MuRIL or Multilingual Representation for Indian Languages (Khanuja et al., 2021) was introduced to boost Indian Natural Language Understanding in 17 Indian languages. Introduced by Google Research India, it uses BERT (Devlin et al., 2019; Hegde et al., 2021) architecture but differs from mBERT as it is trained on translation and transliteration pairs. It uses the IndicNLP-Transliteration library to transliterate the Wikipedia dataset and also the Dakshina [6] dataset to train the model on transliterated data. This approach promises better accuracy on transliterated data. MuRIL trained it with a Masked Language Modeling (MLM) to give a maximum of 80 predictions with 4096 as the batch size and a maximum sequence length of 512.

5.2. XLM-RoBERTa

XLM-Roberta (Conneau et al., 2020), a model by the Facebook AI team, is a transformer-based Masked Language Model (MLM) from over 100 languages, including the low resourced languages released as an update to the previous XLM-100 model (Lample and Conneau, 2019). It follows the training routine as the RoBERTa model (Liu et al., 2019b) which adds it to its name. It distinguishes itself from the other models with its more training data up to 2.5TB of the new at the time of release CommonCrawl data[7] and more languages. The model discusses various topics like the constraints of the multilingual MLMs and that the multilingual model provides better results than monolingual models with state-of-the-art outcomes for cross-lingual classification, question answering, and various other tasks. We have used *xlm-roberta-large* from HuggingFace[8] to predict the pseudo labels for the transliterated data due to its robust results and the fact that it was the largest multilingual language model available at the time of this paper.

$$FFN(x) = max(0, x * W_1 + b_1)W_2 + b_2 \tag{4}$$

[6] https://github.com/google-research-datasets/dakshina
[7] https://commoncrawl.org/
[8] https://huggingface.co/

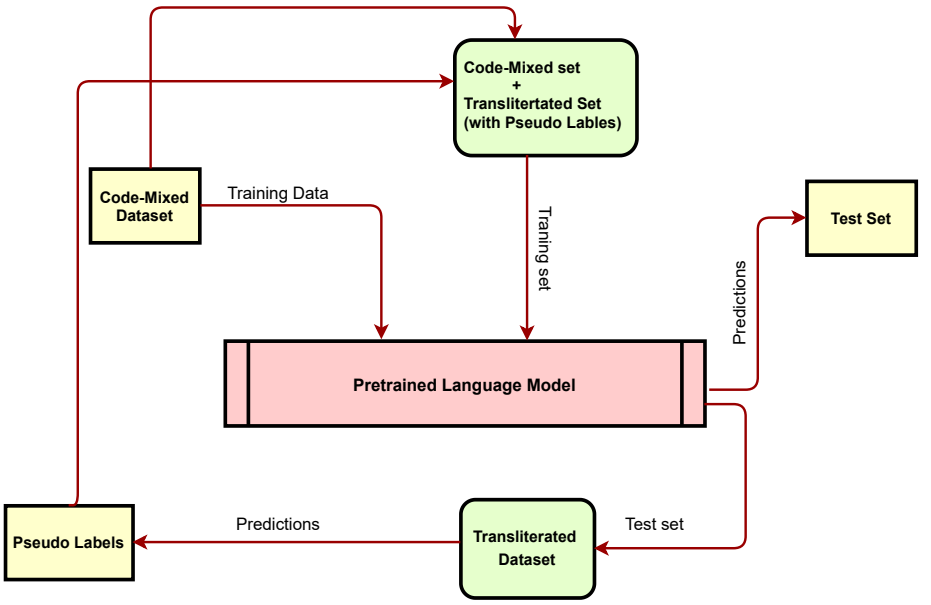


Figure 5: Generating Pseudo Labels on the transliterated dataset.

5.3. Proposed Approach

We transliterate the dataset with the help of the AI4Bharat transliteration application, Indic Deep-Xlit Engine[9]. This deep transliteration engine can translate from Roman script to significant Indian languages and contain under-represented and resourced languages. Its architecture features a seq2seq model with RNNs (Recurrent neural network) and encoder-decoders. The model efficiently learns all the embeddings and weights, and the decoder acquires *topk* predictions which are ranked, and the most likely words are predicted (Bahdanau et al., 2016). The training datasets were transliterated for Malayalam, Tamil and Kannada. Various experiments were conducted using the transliterated dataset. Firstly, the models featured in Yasaswini et al. (2021) were fed with the transliterated data to compare the F1 scores obtained. Later, the transliterated dataset was used to perform pseudo labelling. The encoder in the Indic Deep-Xlit engine is typically a bidirectional LSTM model. The input characters are transformed into an embedding in every timestep which it learns during the training. At the end of the sequence, we also get a compressed context vector from the encoder. This context vector is used by the decoder and with the help of the weighted encoder output vectors, predicts the output. The output at each timestep is the probability distribution of the target language over the characters. Finally, the decoder uses the mechanism called active beam search to get the topk predictions of every input based on the joint probability which are later reranked. This reranking happens with the vocabulary of the target and the most probable words are found.

While the machine learning algorithms and models to produce state-of-the-art results evolve, the limitation of datasets and the uneven imbalance in the classes still poses a problem to researchers. The field of image processing deals with this problem with classical data augmentation techniques like rotation, cropping, zooming and new modus operandi, including GANs and Style transfer (Mikołajczyk and Grochowski, 2018). Pseudo labeling is one technique that can be used to handle this problem in Natural Language Processing. Pseudo-labeling is a semi-supervised learning technique in which a model is trained on a set of labeled data and used to predict the unlabeled test data. The test data, along with their supposed labels, are merged with the training data for additional training (Aroyehun and Gelbukh, 2018). For our cause, we increase the training size of the dataset by fine-tuning a pretrained language model on the actual training set. We use XLM-R, the largest multilingual language model available at the time of experimentation. In addition to being the largest multilingual language model, XLM-R is effective for cross-lingual transfer, which is very significant when we are dealing with code-mixed sentences. After fine-tuning the model, we treat the transliaterated dataset as the test set, and generate pseudo-labels on the same. The new dataset is called **CM-TRA**. We combine the normal training set and dataset with pseudolabels. The test data has been left untouched, and the labels have been

[9]https://github.com/AI4Bharat/IndianNLP-Transliteration

Algorithm 1: Proposed Approach

Input : Code-mixed dataset *CM*

Output : Class Label L, generated for the transliterated dataset using Pseudo-labeling

Let $C = c_1, c_2, .., c_n$ *where* $C_i \in (kn - en/ml - en/ta - en)$ *be the code-mixed sentences*

Transliteration : Code-mixed dataset CM to Kannada/Malayalam/Tamil

Let $T = t_1, t_2, ..., t_n$ *where* $T_i \in (kn/ml/ta)$

for $T_i \in T$ **do**

  Extract cross-lingual embeddings $E[T_i]$ *from XLM-RoBERTa-large, where* $E_{size}[T_i] = 1024$

  $Q = E[T_i] * W^Q$

  $K = E[T_i] * W^K$

  $V = E[T_i] * W^V$

  where W^Q, W^K, W^V are the weight projections for Q, K, and V

  for $i = 0$ **to** 16 **do**

    ; // num(heads) = 16

    $Attention(Q_i, K_i, V_i) = softmax(\frac{QK^T}{\sqrt{d_k}})V$

  end

  $MultiHead(Q, K, V) = Concat(Head_1, Head_2, ..., head_{16})W_O$

  where, $Head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$

end

Loss Function : $L_{CE} = \sum_i^C t_i \log(s_i)$

; // t_i and s_i are the ground truths for every class i in C

Obtain the embeddings from the pooler output, T_E *and feed it as input to an LSTM block*

for $j = 1$ **to** D **do**

  ; // D = Number of memory blocks in LSTM

  $I_t = \sigma(w_i.[h_{t-1}, T_E] + b_i)$ // I_t = Input Gate

  $F_t = \sigma(w_f.[h_{t-1}, T_E] + b_f)$ // F_t = Forget Gate

  $O_t = \sigma(w_o.[h_{t-1}, T_E] + b_o)$ // O_t = Output Gate

  $C_i = \tanh(w_c.[h_{t-1}, T_E] + b_c)$ // C_i = Updating the memory cell, C_i

end

Obtain the final output of the LSTM Block

$T = t_1, t_2, ..., t_n$ *where* $T_i \in (kn/ml/ta)$ *is used as the test set*

Output O is fed into a softmax layer to obtain probabilities (P) of all classes // $N(Class) = 6, 6, 5(kn, ta, ml)$

$Softmax(z) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$ // softmax over K classes

Output Class, $L = argmax(P)$ // Obtaining Pseudo-labels on the transliterated test set

Return : Label L (Pseudo-labels)

Combining $T = t_1, t_2, ..., t_n$ *and* $C = c_1, c_2, .., c_n$ *into a single dataset*

Dataset : CM-TRA dataset

predicted for the transliterated training dataset. Furthermore, the transliterated and the unprocessed training dataset have been collectively used to train the models. We have fine-tuned several multilingual language models on the three datasets as shown in Table 5. We have elaborated on the approach in Algorithm 1.

6. Results

In this section, we report the precision (P), recall (R), F1 scores (F1) of our transformer-based models to identify offensive comments/posts and further classify them offensive-targeted-insult-individual, offensive-targeted-insult-group, offensive-targeted-insult-other, offensive untargeted and not-in-intended-language. The TP, FP, FN, TN values of a class are defined as follows -

1. *TP (True Positive)* - the value of the actual class is positive, and the value of the predicted class is also positive.

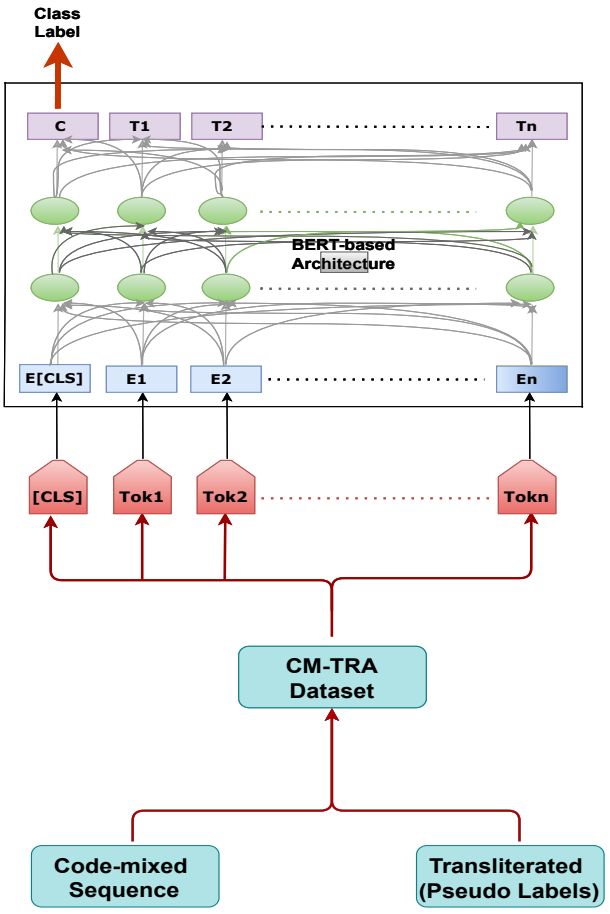


Figure 6: Fine-tuning BERT-based models on the CM-TRA dataset.

2. *FP (False Positive)* - the value of the actual class is negative, but the value of the predicted class is positive.
3. *FN (False Negative)* - the value of the actual class is positive, but the value of the predicted class is negative.
4. *TN (True Negative)* - the value of the actual class is negative, and the value of the predicted class is negative.

$$Precision(P) = \frac{TP}{TP + FP} \tag{5}$$

$$Recall(R) = \frac{TP}{TP + FN} \tag{6}$$

$$F1\ score(F1) = \frac{2*R*P}{R + P} \tag{7}$$

$$P_{\text{weighted}} = \sum_{i=1}^{L} (P \text{ of } i \times \text{ Weight of } i) \tag{8}$$

$$R_{\text{weighted}} = \sum_{i=1}^{L} (R \text{ of } i \times \text{ Weight of } i) \tag{9}$$

Table 5
Weighted Precision, Weighted Recall, and Weighted F1-scores of offensive language detection models on the three datasets

Model	Code-Mixed Dataset								
	Malayalam			Tamil			Kannada		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
mBERT	0.9195	0.9410	0.9301	0.7461	**0.7664**	0.7556	0.6863	0.7082	0.6936
XLM-R	0.9206	0.9380	0.9288	0.5275	0.7263	0.6112	0.6449	**0.7326**	0.6851
DistilmBERT	0.9411	0.9520	0.9465	0.7368	0.7632	0.7489	0.6789	0.7249	**0.7010**
MURiL	0.7780	0.8821	0.8268	0.5275	0.7263	0.6112	0.3012	0.5488	0.3890
IndicBERT	0.9572	**0.9600**	0.9568	0.7150	0.7454	0.7287	0.6714	0.6992	0.6809
ULMFiT	**0.9643**	0.9580	**0.9603**	**0.8220**	0.7650	**0.7895**	**0.7186**	0.6864	0.7000
	Transliterated Dataset								
	Malayalam			Tamil			Kannada		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
mBERT	0.9023	0.9398	0.9202	0.7063	0.7648	0.7286	0.6779	**0.7389**	0.7002
XLM-R	0.8902	0.9265	0.9080	0.5538	0.7320	0.6290	0.6369	0.7198	0.6750
DistilmBERT	0.9089	0.9370	0.9199	0.7248	0.7571	0.7390	0.6789	0.7249	0.7010
MuRIL	0.9039	0.9405	0.9218	0.5275	0.7263	0.6112	0.6432	0.7249	0.6815
IndicBERT	0.9305	0.9445	0.9373	0.7194	0.7354	0.7263	0.6433	0.6722	0.6558
ULMFiT	**0.9521**	**0.9505**	**0.9508**	**0.8033**	**0.7682**	**0.7842**	**0.7304**	0.6979	**0.7115**
	CM-TRA								
	Malayalam			Tamil			Kannada		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
mBERT	0.9468	0.9535	0.9478	0.6865	0.7432	0.7026	0.6048	0.6517	0.6188
XLM-R	0.9370	0.9410	0.9366	0.7284	0.7609	0.7427	0.6997	0.7455	0.7029
DistilmBERT	0.9582	0.9575	0.9537	0.7414	0.7516	0.7461	0.7008	0.7198	0.7037
MuRIL	0.7780	0.8821	0.8268	0.7081	0.7511	0.7045	0.6407	0.7249	0.6801
IndicBERT	0.9306	0.9465	0.9380	0.6867	0.7516	0.7057	0.5937	0.6671	0.6235
ULMFiT	**0.9649**	**0.9610**	**0.9624**	**0.8203**	**0.7719**	**0.7934**	**0.7576**	**0.7104**	**0.7306**

$$F1_{\text{weighted}} = \sum_{i=1}^{L} (F1 \text{ of } i \times \text{ Weight of } i) \tag{10}$$

For future reference, we refer to the dataset comprising the custom code-mixed and transliterated set as CM-TRA. Table 5 shows the weighted average F1-scores of various transformer-based models trained and evaluated on code-mixed, transliterated and CM-TRA datasets of Malayalam, Tamil, and Kannada.

XLM-Roberta Large (Conneau et al., 2020), the largest multilingual model available at the time of our work, was used to predict the labels for the transliterated data. Later, we used the combined data to train various top-performing models. We expected this approach to produce a considerable increase in the F1 scores, as shown in Figure 5. We have discussed our proposed approach in Algorithm 1.

However, test set we are using is the holdout set of (Chakravarthi et al., 2021c), we do not employ any cross-validation techniques in our experiments. As observed in Table 2 and Table 3 representing the class-wise distribution of the training and test sets. To address the issue of class imbalance, we use class weighting. The inverse the weights of each class and pass it as tensor while computing the loss during training (Hande et al., 2021a). While this approach had improved the recall of the classes with low samples, it drastically decreases the performance of the large sampled classes, effectively reducing the overall weighted F1-Scores. Hence, we refrain from using class weighting for computing the

loss during training, and revert back to the traditional computation of loss where all classes are treated with equal importance.

We have experimented with models like multilingual BERT, XLM-RoBERTa, DistilmBERT, MURiL, IndicBERT and ULMFiT. We observe that the ULMFiT model is the better performing model on code-mixed datasets of Malayalam, Tamil with F1-scores 0.9603, 0.7895 respectively, and DistilmBERT is the better performing model on Kannada of F1-score 0.7000. MURiL is the model that gave poor results on Tamil, Malayalam and Kannada code-mixed datasets with F1-scores of 0.6112, 0.8268 and 0.3890, respectively, compared with other models. One of the reasons for poor performance could be class imbalances and code-mixed and writing in non-native languages. The remaining models gave a relatively good performance on the three language code-mixed datasets. The process of fine-tuning on the custom dataset CM-TRA has been computed as shown in Figure 6.

Approach	Precision	Recall	F1-Score	Rank
CM-TRA-ULMFiT	**0.82**	0.77	**0.79**	**1**
(Saha et al., 2021)	0.78	0.78	0.78	2
(Kedia and Nandy, 2021)	0.75	**0.79**	0.77	3
(Zhao and Tao, 2021)	0.75	0.77	0.76	4
(Jayanthi and Gupta, 2021)	0.75	**0.79**	0.76	4
(Sharif et al., 2021)	0.75	0.78	0.76	4
(Li, 2021)	0.74	0.77	0.75	7
(Huang and Bai, 2021)	0.73	0.78	0.75	7
(Balouchzahi et al., 2021)	0.74	0.77	0.75	7
(Tula et al., 2021)	0.74	0.77	0.75	7
(Ghanghor et al., 2021)	0.74	0.77	0.75	7
(Dowlagar and Mamidi, 2021)	0.74	0.75	0.75	7
(Chen and Kong, 2021b)	0.74	0.75	0.74	13
(Vasantharajan and Thayasivam, 2021)	0.71	0.76	0.73	14
(B and A, 2021)	0.74	0.73	0.73	14
(Garain et al., 2021)	0.71	0.74	0.72	16
(Yasaswini et al., 2021)	0.70	0.73	0.71	17
(Dave et al., 2021)	0.72	0.77	0.71	17
(Renjit and Idicula, 2021)	0.67	0.71	0.69	19
(K et al., 2021)	0.64	0.62	0.62	20
(Andrew, 2021)	0.54	0.73	0.61	21

Table 6
Comparisons of the existing models that were developed for the Tamil dataset with other existing models on the dataset. Ranks are based on the descending order of the weighted F1-Scores.

There is a significant improvement in the MURiL model on Malayalam and Kannada transliterated datasets of F1-scores 0.9218 and 0.6815, respectively, while there is no improvement of the model on the Tamil transliterated dataset. The other models showed a marginal decrement in the performance on Malayalam transliterated dataset. In the Tamil transliterated dataset, except the XLM-RoBERTa model, the remaining models performed relatively poor performance, and the XLM-RoBERTa model showed an improvement with an F1-score 0.6290. The multilingual BERT model and ULMFiT model showed a slight increase in the model performance on Kannada transliterated. Surprisingly, on transliterated datasets, ULMFiT still gave better results when compared to the other models. The exceptional fine-tuning methods of the ULMFiT model may also result in giving better performance over other models.

To our expectations, the performance of the models is noticed to be ameliorated on CM-TRA Tamil, Malayalam and Kannada datasets. In comparison to the performance of the models on transliterated datasets, DistilmBERT and XLM-RoBERTa models showed a significant enhancement of the performance on the three languages. It is observed that the IndicBERT model showed no improvement on the datasets. Overall, the ULMFiT model gave promising results on CM-TRA datasets of all the three languages with F1-scores of 0.9624 (Malayalam), 0.7934 (Tamil) and 0.7306 (Kannada).

One of the essential things we can observe in the results is the slight decrease in the F1 score for the transliterated data compared to the code-mixed dataset. This drop can be because all the pretrained models are primarily trained on large English datasets and fewer sentences belonging to a particular language. So, when fine-tuned on only the Dravidian

Approach	Precision	Recall	F1-Score	Rank
(Jayanthi and Gupta, 2021)	0.73	**0.78**	**0.75**	1
(Saha et al., 2021)	**0.76**	0.76	0.74	2
CM-TRA-ULMFiT	**0.76**	0.71	0.73	**3**
(Kedia and Nandy, 2021)	0.71	0.74	0.72	4
(Li, 2021)	0.70	0.75	0.72	4
(Ghanghor et al., 2021)	0.70	0.75	0.72	4
(Sharif et al., 2021)	0.70	0.74	0.71	7
(Vasantharajan and Thayasivam, 2021)	0.69	0.72	0.70	8
(Tula et al., 2021)	0.69	0.72	0.70	8
(B and A, 2021)	0.71	0.74	0.70	8
(Balouchzahi et al., 2021)	0.68	0.72	0.69	11
(Zhao and Tao, 2021)	0.65	0.74	0.69	11
(Garain et al., 2021)	0.62	0.71	0.66	13
(Dowlagar and Mamidi, 2021)	0.66	0.65	0.65	14
(Chen and Kong, 2021a)	0.64	0.67	0.64	15
(Huang and Bai, 2021)	0.65	0.69	0.64	15
(Dave et al., 2021)	0.69	0.69	0.64	15
(Andrew, 2021)	0.66	0.67	0.63	18
(Renjit and Idicula, 2021)	0.62	0.63	0.62	19
(K et al., 2021)	0.65	0.54	0.58	20
(Yasaswini et al., 2021)	0.46	0.48	0.47	21
(Que, 2021)	0.60	0.30	0.33	22

Table 7
Comparisons of the existing models developed for the code-mixed Kannada dataset.

Approach	Precision	Recall	F1-Score	Rank
(Saha et al., 2021)	**0.97**	**0.97**	**0.97**	1
(Balouchzahi et al., 2021)	0.97	0.97	0.97	1
(Kedia and Nandy, 2021)	0.97	0.97	0.97	1
(Tula et al., 2021)	0.97	0.97	0.97	1
CM-TRA-ULMFIT	0.96	0.96	0.96	**5**
(Vasantharajan and Thayasivam, 2021)	0.96	0.96	0.96	6
(Dowlagar and Mamidi, 2021)	0.96	0.96	0.96	6
(Jayanthi and Gupta, 2021)	0.97	0.97	0.96	6
(Renjit and Idicula, 2021)	0.95	0.95	0.95	9
(Ghanghor et al., 2021)	0.94	0.95	0.95	9
(B and A, 2021)	0.95	0.96	0.95	9
(Dave et al., 2021)	0.96	0.96	0.95	9
(Li, 2021)	0.93	0.94	0.94	13
(Andrew, 2021)	0.94	0.94	0.93	13
(Chen and Kong, 2021a)	0.92	0.94	0.93	13
(Yang, 2021)	0.91	0.94	0.93	13
(Sharif et al., 2021)	0.92	0.94	0.93	13
(Zhao and Tao, 2021)	0.91	0.94	0.92	18
(Huang and Bai, 2021)	0.89	0.93	0.91	19
(Yasaswini et al., 2021)	0.84	0.87	0.86	20
(K et al., 2021)	0.90	0.82	0.85	21
(Nair and Fernandes, 2021)	0.89	0.84	0.85	21
(Garain et al., 2021)	0.77	0.43	0.54	23

Table 8
Comparisons of the existing models developed for the code-mixed Malayalam dataset.

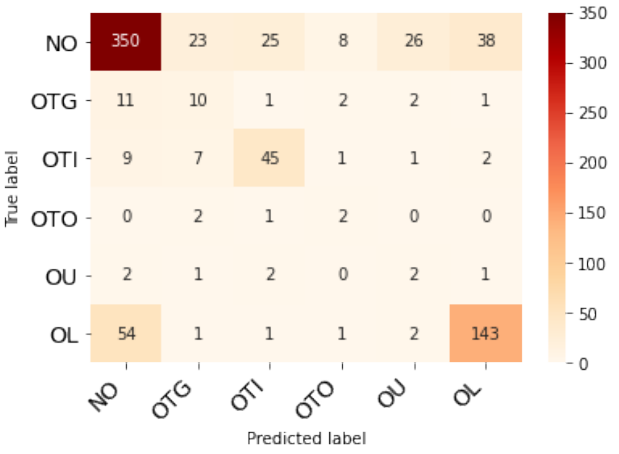


Figure 7: Heatmap of the confusion matrix for the test set of code-mixed English-Kannada

	Kannada				Malayalam			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
Not Offensive	0.8216	0.7447	0.7812	417	0.9875	0.9710	0.9792	1,765
Other Language	0.7730	0.7079	0.7390	185	0.8408	0.9496	0.8919	157
Offensive Targeted Individual	0.6000	0.6923	0.6429	75	0.5926	0.8421	0.6957	27
Offensive Targeted Group	0.2273	0.3704	0.2817	44	0.5217	0.6667	0.5854	23
Offensive Targeted Others	0.1429	0.4000	0.2105	14	-	-	-	-
Offensive Untargeted	0.0606	0.2500	0.0976	33	0.6897	0.6667	0.6667	29
Accuracy			0.7104	768			0.9610	2,001
Macro Average	0.4376	0.5275	0.4588	768	0.7265	0.8192	0.7660	2,001
Weighted Average	0.7576	0.7104	0.7306	768	0.9649	0.9610	0.9624	2,001

Table 9
Classification report of CM-TRA-ULMFiT on the Kannada and Malayalam test sets.

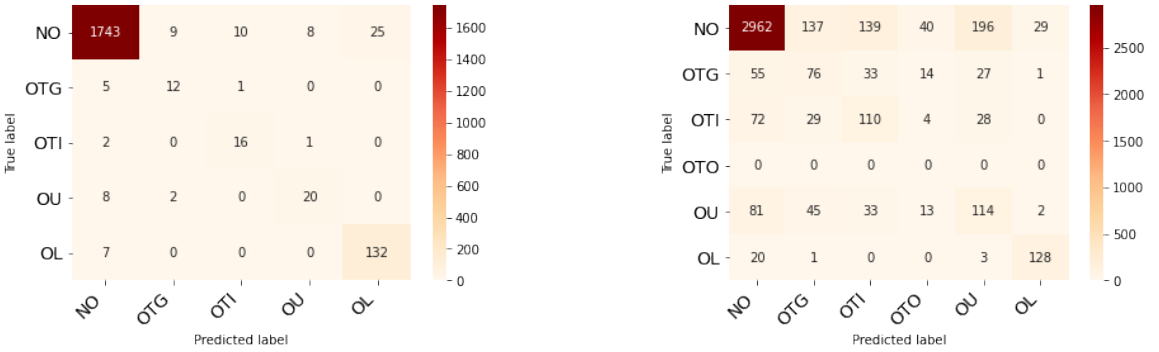
language, the model might not detect the other languages in the test dataset well. Our code-mixed dataset shows the predominant use of English with the Dravidian languages. The presence of English sentences in the code-mixed data seems to be the determining point for this decrease. The other factor responsible for this can be the inconsistency of transliterating the data using the Ai4Bharat transliteration application. The pseudo labels obtained from these data may not be accurate, causing the models to get misled when trained on transliterated data.

However, as expected, the brainchild of our research, the models, when trained on the CM-TRA dataset, gave exceptional results. With the combination of the code-mixed and transliterated data, the model successfully learned the code-mixed and language aspects. One of the trends observed is that the BERT based models, including mBERT, DistilmBERT, IndicBERT, show declining F1 scores for the CM-TRA dataset, especially for the languages of Kannada and Tamil. This decline could indicate the types of Tamil and Kannada datasets used during pretraining for the models compared to the code-mixed data scraped from the internet. IndicBERT, which is extensively pretrained in Indian languages, shows a rapid decline in the results. XLM-RoBERTa, though it possesses BERT architecture, is trained on transliterated datasets, which probably is the reason for not showing this decreasing trend. However, due to superior pretraining strategies, ULMFiT fared well on the dataset and procured the best F1 scores for all three languages on the CM-TRA dataset. The classification reports for the best performing models are tabulated in Table 9 and Table 10.

The results obtained for this task has succeeded in surpassing many of the state-of-the-art models on the dataset Chakravarthi et al. (2021a). For Tamil, our model secured the first position with an F1-score of 0.79 by surpassing Saha et al. (2021) who scored an F1-score of 0.78 as observed in Table 6. The model ranked third in Kannada after Jayanthi

	Tamil			
	Precision	Recall	F1-Score	Support
Not Offensive	0.9285	0.8456	0.8851	3,190
Other Language	0.8000	0.8421	0.8205	165
Offensive Targeted Individual	0.3492	0.4527	0.3943	315
Offensive Targeted Group	0.2639	0.3689	0.3077	288
Offensive Targeted Others	0.0	0.0	0.0	71
Offensive Untargeted	0.3098	0.3958	0.3476	368
Accuracy			0.7719	4,392
Macro Average	0.4419	0.4842	0.4592	4,392
Weighted Average	0.8203	0.7719	0.7934	4,392

Table 10
Classification report of CM-TRA-ULMFiT on the Tamil test sets.



(a) Heatmap of the confusion matrix for the test set of code-mixed Malayalam-English.

(b) Heatmap of the confusion matrix for the test set of code-mixed Tamil-English.

Figure 8: My flowers.

and Gupta (2021) and Saha et al. (2021) with an F1-score of 0.73 while the first ranker scored 0.75 as shown in Table 7. For Malayalam, the model came second with an F1-score of 0.9649 while the first rank is shared by 4 teams Saha et al. (2021), Balouchzahi et al. (2021), Kedia and Nandy (2021) and Tula et al. (2021) with 0.97 as seen in Table 8. Saha et al. (2021) used a genetic algorithm technique for ensembling different transformer models. Jayanthi and Gupta (2021) also adopted an ensembling method using mBERT and XLM-RoBERTa with a masked language modelling objective. The confusion matrices of the best performing models are displayed in Figure 7, Figure 8a, and Figure 8b.

6.1. Error Analysis

6.1.1. Tamil

In this section, we discuss the misclassification errors of the model for the classification task. Some of the examples with English translations are discussed for each target class in Fig. 9. Here, the most six misclassified comments are discussed. 1,002 comments out of 4,392 in the testing set are wrongly predicted, and 3,390 sentences are correctly predicted. Most of the wrong predictions are that the offensiveness is not projected explicitly in terms of words in such sentences. However, the emotions in those sentences depict the offensiveness. For example, consider the following sentence.

Text: *Epudi da oru ANJAN oru AYAN oru NJK. Mari irukumo*

Translation: *How it would be? May be Anjan,Ayan or NJK?*

In this sentence, no offensive word is used. However, the sarcasm in the text is offensive, and it would be understood by the people who know the Tamil Cinema and who watched Anjan, Ayan and NJK movies already (All are Tamil movie titles). Another problem in some classifications is that the sentences are made with positive words, yet it gives offensive meaning implicitly. The sentence is misclassified as not offensive, though it comes under the category of

Label	Example	Predicted
Not-Offensive	Wig paatha Vijay use pannamari irukku விஜய் உபயோகித்த விக் போல் உள்ளது **The wig looks like it is used by Vijay.**	Offensive_Targeted_Insult_Individual
Offensive-Targeted-Insult-Individual	Idha vida Rajiniya Troll panna mudiyave mudiyadhu da saamy இதை விட ரஜினியை ட்ரோல் பண்ண முடியாதுடா சாமி **God, Rajni can't be trolled more than this.**	Not_offensive
Offensive-Targeted-Insult-Group	Guys Teaser la suriya Anna name podalai guys Enna nadakuthu கைஸ் டீசர்ல சூர்யா அண்ணன் பேர் போடல கைஸ் என்ன நடக்குது **Surya's name is not in the teaser. What's happening?**	Not_offensive
Offensive-Targeted-Insult-Other	Rajin political entry dailog 1996/2016 10 years one dailog naaku baag nachindi ரஜினி பொலிடிகல் என்ட்ரி டயலாக் 1996/2016 பத்து வருஷம் ஒரு டயலாக் **Political entry dialog of Rajnikanth between 1996-2016.I liked it.**	Not_offensive
Offensive-Untargeted	Epudi da oru ANJAN oru AYAN oru NJK. mari irukumo எப்புடி டா ஒரு அஞ்சான், ஒரு அயன், ஒரு NGK மாதிரி இருக்குமோ **How it would be? May be Anjan,Ayan or NJK?**	Not_offensive
Not-in-intended-language	Wow. Ithu 96 yaara yamatha paakkura வாவ், இது 96 யாரை ஏமாத்த பாக்கற Don't Cheat. This is 96.	Offensive_Untargeted

Figure 9: Wrong Predictions in the Tamil dataset

'Offensive-Untargeted'. Consider another example,

Text: *Rajini political entry dailog 1996/2016 10 years one dailog naaku baag nachindi*

Translation: *Political entry dialog of Rajnikanth between 1996-2016. I liked it.*

This sentence also does not have any offensive words. Though it has positive words such as 'I liked it', One can only understand the sarcasm if one knows Tamil movies and the actor Rajnikanth's dialogues in his movies. Hence, the given sentence is not predicted as offensive even though it comes under Offensive-Targeted-Insult-Other.

6.1.2. Kannada

The classifications in the Kannada language saw the poorest F1-scores of the three languages. The test dataset consisted of 778 sentences, out of which 552 sentences were classified well, while 226 sentences failed to be grouped into their respective class. The confusion matrix can be viewed in Figure 7. We can see that many sentences are mismatched for various reasons that the model cannot detect. Consider the following sentences,

Text: *Finally sonu gowda b day dhinane tiktok ban aythu*

Translation: *Finally, TikTok got banned on the birthday of Sonu Gowda!* The sentence might sound like a fact. It does not contain any offensive words or anything. However, people who know Sonu Gowda (a TikTok star) would understand the sarcasm behind the comment on how TikTok was banned in India just on her birthday. Therefore, the model does not predict it as Offensive-Targeted-Insult-Individual, the class to which it should belong.

Text: *Found 806 rashmika mangannas....*

Translation: *Found 806 Rashmika monkeys*

This sentence is very confusing due to the use of puns by the commenter. Rashmika Mandanna is a well known Indian actress, and "manganna" which sounds like "Mandanna" means monkey in Kannada. This replaced alphabet can be

Label	Example	Predicted
Not-Offensive	ಚಂದನ್ ಶೆಟ್ಟಿ ಟ್ರೊಲ್ ವಿಡಿಯೋ ನೀವು ನೋಡಿ ನೆಗಡ್ಡೆ ಇದ್ರೆ ನಾವ್ ಗ್ಯಾರಂಟಿ ಚಂದನ್ ಶೆಟ್ಟಿ ಟ್ರೊಲ್ ವಿಡಿಯೋ ನೀವು ನೋಡಿ ನೆಗಡ್ಡೆ ಇದ್ರೆ ನಾವ್ ಗ್ಯಾರಂಟಿ I guarantee you that you are going to laugh after watching Chandan Shetty troll video	Offensive_Targeted_Insult_Individual
Offensive-Targeted-Insult-Individual	Finally, sonu gowda b day dhinane tiktok ban aythu ಫೈನಲ್ಲಿ ಸೋನು ಗೌಡ ಬರ್ತ್ಡೇ ದಿನನೇ ಟಿಕ್ ಟಾಕ್ ಬ್ಯಾನ್ ಆಯಿತು Finally, TikTok got banned on the birthday of Sonu Gowda	Not_offensive
Offensive-Targeted-Insult-Group	Found 806 rashmika mangannas.... ಫೌಂಡ್ ಆಂಒ ರಶ್ಮಿಕಾ ಮಂಗಣ್ಣಾಸ್ Found 806 Rashmika monkeys	Not_offensive
Offensive-Targeted-Insult-Other	Guru ee desha uddhara agalla bedu bhai indian youth waste bedu ಗುರು ಈ ದೇಶ ಉದ್ಧಾರ ಆಗಲ್ಲ ಬಿಡು ಭಾಯ್ ಇಂಡಿಯನ್ ಯೂಥ್ ವೇಸ್ಟ್ ಬಿಡು Brother this country won't develop as Indian youth are waste.	Offensive-Targeted-Insult-Group
Offensive-Untargeted	ningyak like madbeko ನಿನಿಗೆ ಯಾಕೆ ಲೈಕ್ ಮಾಡ್ಬೇಕೊ Why should I "like" you?	Offensive-Targeted-Insult-Individual
Not-in-intended-language	togari tippa supar ತೊಗರಿ ತಿಪ್ಪ ಸೂಪರ್ "Thogari Tippa" super	Not-offensive

Figure 10: Wrong Predictions in the Kannada dataset

considered a spelling error, and the sentence could be classified into other classes like "not-Kannada" or "Not-offensive". Hence, the model failed to categorize it into "Offensive-Targeted-Insult-Group", which is the suitable class.

Text: *togari tippa supar*

Translation: *"Thogari Tippa" super*

"Thogari Tippa" is the title of a Kannada movie and is derived from Kannada. The model fails to detect that "Thogari Tippa" is a movie and classifies it into "Not-offensive". However, the sentence belongs to the group, "not-Kannada".

6.2. Malayalam

The performance of all the models on the code-mixed Malayalam dataset was very exceptional, as most of the models achieved weighted F1-Scores greater than 0.90. The best performing model misclassified relatively lesser examples in contrast to its performance on other datasets, misclassifying 78 samples among the 2,001 samples in the test set.

Text: *Ella oollapadathinteyum stiram cheruva. 8 onilayil padam pottum.*

Translation: *Exact ingredient of all flop movies. This movie is going to be a failure.*

Though this comment does not have any offensive words, this sentence as a whole is meant to insult the director or any person behind that movie. Based on just a trailer, the author sees the film with prejudice and gives negative reactions about the movie itself. Hence the sentence is wrongly classified as 'Not Offensive' but it belongs to the 'Offensive Targeted Individual' category. Let us see another example,

Text: *Valla panikkum pokkude nalla prayam indallo mammotty*

Translation: *You are too old Mammootty. Can't you do some real work?*

This comment also does not contain any offensive words but this explicitly insults a prominent actor of Malayalam

Label	Example	Predicted
Not Offensive	Oh my God ഇത ചീറും എന്റെ ദൈവമേ, ഇത് ചീറും. Oh my God, this will be a blast!	Other Language
Offensive Targeted Individual	Ella oollapadathinteyum stiram cheruva. 8 nilayil padam pottum. എല്ലാ ഊളപ്പടത്തിന്റെയും സ്തിരം ചേരുവ. 8 നിലയിൽ പടം പൊട്ടും. Exact ingredient of all flop movies. This movie is going to be a failure.	Not Offensive
Offensive Targeted Group	Nattalillatha lalappan. നട്ടലില്ലാത്ത ലാലപ്പൻ. No spine for Mohanlal.	Not Offensive
Offensive Untargeted	Pavam sanjeev pillaiku enthengilum credit kodukeda nayinte makkale. പാവം സഞ്ജീവ് പിള്ളേയ്ക്ക് എന്തെങ്കിലും ക്രെഡിറ്റ് കൊടുക്കെടാ നായിന്റെ മക്കളെ. Please give some credits to Sanjeev pillai also you assholes.	Not Offensive
Other Language	similar to Kirik party (kannada movie). കിർക് പാർട്ടി (കന്നഡ പടം) പോലെ. similar to Kirik party (kannada movie).	Not Offensive

Figure 11: Wrong Predictions in the Malayalam dataset

movies. Mammootty is considered one of the legends among Indian actors yet this comment deliberately humiliates him. So this comment with no doubt should fall into the 'Offensive Targeted Individual' category but wrongly predicted as 'Not Offensive' as there are no offensive words in the comment.

7. Conclusion

The increasing amount of offensive language persisting on social media and relatively fewer approaches that address code-mixing in Dravidian languages have pushed us to improve our approaches in offensive language identification. This paper revisited offensive language identification by prioritising the dataset rather than the models to increase the overall weighted F1-scores on the test set. We intend to address the lack of data by generating pseudo-labels on the dataset, which was transliterated to the respective Dravidian languages, as Dravidian languages are primarily under-resourced. This approach aims at improving cross-lingual transfer by having a multilingual training dataset. In this paper, we have presented an approach to identify the offensive language in social media for multilingual countries that comprise code-mixed sentences. We observe that when we fine-tune ULMFiT on our newly constructed custom dataset, it yields the best performance in code-mixed Tamil while almost achieving very competitive results on several other benchmarked models on the respective languages. For future work, we intend to combine the three languages and develop a multilingual offensive language identification system for code-mixed Dravidian languages.

CRediT authorship contribution statement

Adeep Hande: Conceptualization, Investigation, Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Karthik Puranik:** Investigation, Methodology, Formal Analysis, Software, Writing - Original draft, Writing- editing. **Konthala Yasaswini:** Investigation, Methodology, Visualization, Writing - Original draft. **Ruba Priyadharshini:** Investigation, Supervision, Data curation, Writing- reviewing & editing. **Sajeetha Thavareesan:** Investigation, Supervision, Writing - reviewing & editing. **Anbukkarasi Sampath:** Formal Analysis, Investigation, Supervision, Writing - original draft, Writing - reviewing & editing. **Kogilavani Shanmugavadivel:** Supervision, Writing - reviewing & editing. **Durairaj Thenmozhi:** Supervision, Writing - reviewing & editing. **Bharathi Raja Chakravarthi:** Conceptualization, Formal Analysis, Investigation, Supervision, Data curation, Writing - original draft, Writing - reviewing & editing.

Acknowledgements

The author Bharathi Raja Chakravarthi was supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2 (Insight_2), co-funded by the European Regional Development Fund and Irish Research Council grant IRCLA/2017/129 (CARDAMOM-Comparative Deep Models of Language for Minority and Historical Languages).

References

Abraham, S., 2003. Chera, chola, pandya: Using archaeological evidence to identify the tamil kingdoms of early historic south india. Asian Perspectives 42. doi:10.1353/asi.2003.0031.

Andrew, J.J., 2021. JudithJeyafreedaAndrew@DravidianLangTech-EACL2021:offensive language detection for Dravidian code-mixed YouTube comments, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 169–174. URL: https://aclanthology.org/2021.dravidianlangtech-1.22.

Aroyehun, S.T., Gelbukh, A., 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), Association for Computational Linguistics, Santa Fe, New Mexico, USA. pp. 90–97. URL: https://www.aclweb.org/anthology/W18-4411.

Asher, R., 2013. Malayalam. Routledge.

B, B., A, A.S., 2021. SSNCSE_NLP@DravidianLangTech-EACL2021: Offensive language identification on multilingual code mixing text, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 313–318. URL: https://aclanthology.org/2021.dravidianlangtech-1.45.

Bahdanau, D., Cho, K., Bengio, Y., 2016. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473.

Balouchzahi, F., B K, A., Shashirekha, H.L., 2021. MUCS@DravidianLangTech-EACL2021:COOLI-code-mixing offensive language identification, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 323–329. URL: https://aclanthology.org/2021.dravidianlangtech-1.47.

Bisht, A., Singh, A., Bhadauria, H.S., Virmani, J., Kriti, 2020. Detection of Hate Speech and Offensive Language in Twitter Data Using LSTM Model. Springer Singapore, Singapore. pp. 243–264. URL: https://doi.org/10.1007/978-981-15-2740-1_17, doi:10.1007/978-981-15-2740-1_17.

Brassard-Gourdeau, E., Khoury, R., 2019. Subversive toxicity detection using sentiment information, in: Proceedings of the Third Workshop on Abusive Language Online, pp. 1–10.

Bright, W., 1999. R. e. asher & t. c. kumari, malayalam . (descriptive grammars.) london & new york: Routledge, 1997. pp. xxvi, 491. Language in Society 28, 482–483.

Burnap, P., Williams, M.L., 2014. Hate speech, machine classification and statistical modelling of information flows on twitter: Interpretation and communication for policy decision making .

Chaffey, D., 2021. Global social media statistics research summary. https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/. [Online; accessed 23-July-2021].

Chakravarthi, B.R., Jose, N., Suryawanshi, S., Sherly, E., McCrae, J.P., 2020a. A sentiment analysis dataset for code-mixed Malayalam-English, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France. pp. 177–184. URL: https://www.aclweb.org/anthology/2020.sltu-1.25.

Chakravarthi, B.R., Muralidaran, V., Priyadharshini, R., McCrae, J.P., 2020b. Corpus creation for sentiment analysis in code-mixed Tamil-English text, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France. pp. 202–210. URL: https://www.aclweb.org/anthology/2020.sltu-1.28.

Chakravarthi, B.R., Priyadharshini, R., Jose, N., Kumar M, A., Mandl, T., Kumaresan, P.K., Ponnusamy, R., R L, H., McCrae, J.P., Sherly, E., 2021a. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 133–145. URL: https://www.aclweb.org/anthology/2021.dravidianlangtech-1.17.

Chakravarthi, B.R., Priyadharshini, R., Jose, N., Kumar M, A., Mandl, T., Kumaresan, P.K., Ponnusamy, R., R L, H., McCrae, J.P., Sherly, E., 2021b. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 133–145. URL: https://www.aclweb.org/anthology/2021.dravidianlangtech-1.17.

Chakravarthi, B.R., Priyadharshini, R., Muralidaran, V., Jose, N., Suryawanshi, S., Sherly, E., McCrae, J.P., 2021c. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. arXiv:2106.09460.

Chaudhari, S., Chaudhari, P., Fegade, P., Kulkarni, A., Patil, R.P., 2019. Hate speech and abusive language suspect identification and report generation. International Journal of Scientific & Technology Research 8, 1609–1611. URL: https://www.ijstr.org/final-print/nov2019/Hate-Speech-And-Abusive-Language-Suspect-Identification-And-Report-Generation.pdf.

Chen, S., Kong, B., 2021a. cs@DravidianLangTech-EACL2021: Offensive language identification based on multilingual BERT model, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 230–235. URL: https://www.aclweb.org/anthology/2021.dravidianlangtech-1.31.

Chen, S., Kong, B., 2021b. cs@DravidianLangTech-EACL2021: Offensive language identification based on multilingual BERT model, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 230–235. URL: https://aclanthology.org/2021.dravidianlangtech-1.31.

Chen, Y., Zhou, Y., Zhu, S., Xu, H., 2012a. Detecting offensive language in social media to protect adolescent online safety, in: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, IEEE. pp. 71–80.

Chen, Y., Zhou, Y., Zhu, S., Xu, H., 2012b. Detecting offensive language in social media to protect adolescent online safety, in: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, pp. 71–80. doi:10.1109/SocialCom-PASSAT.2012.55.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V., 2020. Unsupervised cross-lingual representation learning at scale. arXiv:1911.02116.

Dadvar, M., Trieschnigg, D., Ordelman, R., de Jong, F., 2013. Improving cyberbullying detection with user context, in: European Conference on Information Retrieval, Springer. pp. 693–696.

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, Association for Computational Linguistics, Kyiv. pp. 266–269. URL: https://aclanthology.org/2021.dravidianlangtech-1.37.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota. pp. 4171–4186. URL: https://aclanthology.org/N19-1423, doi:10.18653/v1/N19-1423.

Dowlagar, S., Mamidi, R., 2021. Graph convolutional networks with multi-headed attention for code-mixed sentiment analysis, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 65–72. URL: https://aclanthology.org/2021.dravidianlangtech-1.8.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., Lehmann, S., 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark. pp. 1615–1625. URL: https://www.aclweb.org/anthology/D17-1169, doi:10.18653/v1/D17-1169.

Garain, A., Mandal, A., Naskar, S.K., 2021. JUNLP@DravidianLangTech-EACL2021: Offensive language identification in Dravidian langauges, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 319–322. URL: https://aclanthology.org/2021.dravidianlangtech-1.46.

George, K.M., 1972. Western influence on Malayalam language and literature. Sahitya Akademi.

Ghanghor, N., Krishnamurthy, P., Thavareesan, S., Priyadharshini, R., Chakravarthi, B.R., 2021. IIITK@DravidianLangTech-EACL2021: Offensive language identification and meme classification in Tamil, Malayalam and Kannada, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 222–229. URL: https://aclanthology.org/2021.dravidianlangtech-1.30.

Hande, A., Hegde, S.U., Priyadharshini, R., Ponnusamy, R., Kumaresan, P.K., Thavareesan, S., Chakravarthi, B.R., 2021a. Benchmarking multi-task learning for sentiment analysis and offensive language identification in under-resourced dravidian languages. arXiv:2108.03867.

Hande, A., Priyadharshini, R., Chakravarthi, B.R., 2020. KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection, in: Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media, Association for Computational Linguistics, Barcelona, Spain (Online). pp. 54–63. URL: https://www.aclweb.org/anthology/2020.peoples-1.6.

Hande, A., Priyadharshini, R., Sampath, A., Thamburaj, K.P., Chandran, P., Chakravarthi, B.R., 2021b. Hope speech detection in under-resourced kannada language. arXiv:2108.04616.

Hande, A., Puranik, K., Priyadharshini, R., Chakravarthi, B.R., 2021c. Domain identification of scientific articles using transfer learning and ensembles, in: Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2021 Workshops, WSPA, MLMEIN, SDPRA, DARAI, and AI4EPT, Delhi, India, May 11, 2021 Proceedings 25, Springer International Publishing. pp. 88–97.

Hande, A., Puranik, K., Priyadharshini, R., Thavareesan, S., Chakravarthi, B.R., 2021d. Evaluating pretrained transformer-based models for covid-19 fake news detection, in: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pp. 766–772. doi:10.1109/ICCMC51019.2021.9418446.

Hegde, S.U., Hande, A., Priyadharshini, R., Thavareesan, S., Chakravarthi, B.R., 2021. Uvce-iiitt@ dravidianlangtech-eacl2021: Tamil troll meme classification: You need to pay more attention. arXiv preprint arXiv:2104.09081 .

Howard, J., Ruder, S., 2018. Universal language model fine-tuning for text classification. arXiv:1801.06146.

Huang, B., Bai, Y., 2021. HUB@DravidianLangTech-EACL2021: Identify and classify offensive text in multilingual code mixing in social media, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 203–209. URL: https://aclanthology.org/2021.dravidianlangtech-1.27.

Jayanthi, S.M., Gupta, A., 2021. SJ_AJ@DravidianLangTech-EACL2021: Task-adaptive pre-training of multilingual BERT models for offensive language identification, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 307–312. URL: https://www.aclweb.org/anthology/2021.dravidianlangtech-1.44.

Jose, N., Chakravarthi, B.R., Suryawanshi, S., Sherly, E., McCrae, J.P., 2020. A survey of current datasets for code-switching research, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS).

K, S., B, P., Kp, S., 2021. Amrita_CEN_NLP@DravidianLangTech-EACL2021: Deep learning-based offensive language identification in Malayalam, Tamil and Kannada, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 249–254. URL: https://aclanthology.org/2021.dravidianlangtech-1.34.

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

Kalyan, K.S., Rajasekharan, A., Sangeetha, S., 2021. Ammus: A survey of transformer-based pretrained models in natural language processing. arXiv preprint arXiv:2108.05542 .

Kapoor, K., Tamilmani, K., Rana, N., Patil, P., Dwivedi, Y., Nerur, S., 2018. Advances in social media research: Past, present and future. Information Systems Frontiers 20. doi:10.1007/s10796-017-9810-y.

Karimi, A., Rossi, L., Prati, A., 2021. Uniparma at semeval-2021 task 5: Toxic spans detection using characterbert and bag-of-words model. arXiv:2103.09645.

Kawate, S., Patil, K., 2017. Analysis of foul language usage in social media text conversation. International Journal of Social Media and Interactive Learning Environments 5, 227. doi:10.1504/IJSMILE.2017.10008890.

Kedia, K., Nandy, A., 2021. indicnlp@kgp at DravidianLangTech-EACL2021: Offensive language identification in Dravidian languages, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 330–335. URL: https://aclanthology.org/2021.dravidianlangtech-1.48.

Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., Margam, D.K., Aggarwal, P., Nagipogu, R.T., Dave, S., Gupta, S., Gali, S.C.B., Subramanian, V., Talukdar, P., 2021. Muril: Multilingual representations for indian languages. arXiv:2103.10730.

Klamma, R., Spaniol, M., Denev, D., 2006. Paladin: A pattern based approach to knowledge discovery in digital social networks.

Knight, K., Graehl, J., 1997. Machine transliteration. CoRR cmp-lg/9704003. URL: http://arxiv.org/abs/cmp-lg/9704003.

Krishnamurti, B., 2003. The dravidian languages. The Dravidian Languages , 1–545doi:10.1017/CBO9780511486876.

Lample, G., Conneau, A., 2019. Cross-lingual language model pretraining. arXiv:1901.07291.

Lee, D.H., 2013. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. ICML 2013 Workshop : Challenges in Representation Learning (WREPL) .

Li, Z., 2021. Codewithzichao@DravidianLangTech-EACL2021: Exploring multilingual transformers for offensive language identification on code mixing text, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 164–168. URL: https://aclanthology.org/2021.dravidianlangtech-1.21.

Liu, P., Li, W., Zou, L., 2019a. Nuli at semeval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers, in: Proceedings of the 13th international workshop on semantic evaluation, pp. 87–91.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization. arXiv:1711.05101.

Luu, S.T., Nguyen, N.L.T., 2021. Uit-ise-nlp at semeval-2021 task 5: Toxic spans detection with bilstm-crf and toxic bert comment classification. arXiv:2104.10100.

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

Malmasi, S., Zampieri, M., 2017. Detecting hate speech in social media. arXiv:1712.06427.

Mandal, S., Singh, A.K., 2018. Language identification in code-mixed data using multichannel neural networks and context capture, in: Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text, Association for Computational Linguistics, Brussels, Belgium. pp. 116–120. URL: https://www.aclweb.org/anthology/W18-6116, doi:10.18653/v1/W18-6116.

Mandl, T., Modha, S., Kumar M, A., Chakravarthi, B.R., 2020. Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german, in: Forum for Information Retrieval Evaluation, Association for Computing Machinery, New York, NY, USA. p. 29–32. URL: https://doi.org/10.1145/3441501.3441517, doi:10.1145/3441501.3441517.

Merity, S., Xiong, C., Bradbury, J., Socher, R., 2016. Pointer sentinel mixture models. arXiv:1609.07843.

Mikołajczyk, A., Grochowski, M., 2018. Data augmentation for improving deep learning in image classification problem, in: 2018 International Interdisciplinary PhD Workshop (IIPhDW), pp. 117–122. doi:10.1109/IIPHDW.2018.8388338.

Nair, S., Fernandes, D., 2021. professionals@DravidianLangTech-EACL2021: Malayalam offensive language identification - a minimalistic approach, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 175–179. URL: https://aclanthology.org/2021.dravidianlangtech-1.23.

Narasimhachar, R., 1988. History of Kannada Literature: readership lectures. Asian Educational Services.

Nasir Ansari, J., Khatoon, A., Bharadwaj, S., 2018. Social media users in india: A futuristic approach.

Noever, D., 2018. Machine learning suites for online toxicity detection. arXiv:1810.01869.

Pitsilis, G.K., Ramampiaro, H., Langseth, H., 2018. Effective hate-speech detection in twitter data using recurrent neural networks. Applied Intelligence 48, 4730–4742. URL: https://doi.org/10.1007%2Fs10489-018-1242-y, doi:10.1007/s10489-018-1242-y.

Puranik, K., Hande, A., Priyadharshini, R., Thavareesan, S., Chakravarthi, B.R., 2021. Iiitt@ lt-edi-eacl2021-hope speech detection: There is always

hope in transformers. arXiv preprint arXiv:2104.09066 .

Que, Q., 2021. Simon @ DravidianLangTech-EACL2021: Detecting offensive content in Kannada language, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 160–163. URL: https://aclanthology.org/2021.dravidianlangtech-1.20.

Razavi, A.H., Inkpen, D., Uritsky, S., Matwin, S., 2010. Offensive language detection using multi-level classification, in: Canadian Conference on Artificial Intelligence, Springer. pp. 16–27.

Renjit, S., Idicula, S.M., 2021. CUSATNLP@DravidianLangTech-EACL2021:language agnostic classification of offensive content in tweets, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 236–242. URL: https://aclanthology.org/2021.dravidianlangtech-1.32.

Rice, E.P., 1982. A History of Kannada Literature. Asian educational services.

Risch, J., Ruff, R., Krestel, R., 2020. Offensive language detection explained, in: Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, European Language Resources Association (ELRA), Marseille, France. pp. 137–143. URL: https://www.aclweb.org/anthology/2020.trac-1.22.

Ruder, S., Peters, M.E., Swayamdipta, S., Wolf, T., 2019. Transfer learning in natural language processing, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, pp. 15–18.

Saha, D., Paharia, N., Chakraborty, D., Saha, P., Mukherjee, A., 2021. Hate-alert@DravidianLangTech-EACL2021: Ensembling strategies for transformer-based offensive language detection, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 270–276. URL: https://aclanthology.org/2021.dravidianlangtech-1.38.

Sanh, V., Debut, L., Chaumond, J., Wolf, T., 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv:1910.01108.

Sap, M., Card, D., Gabriel, S., Choi, Y., Smith, N.A., 2019. The risk of racial bias in hate speech detection, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy. pp. 1668–1678. URL: https://www.aclweb.org/anthology/P19-1163, doi:10.18653/v1/P19-1163.

Sekhar, A.C., 1951. Evolution of malayalam. Bulletin of the Deccan College Research Institute 12, 1–216.

Sharif, O., Hossain, E., Hoque, M.M., 2021. NLP-CUET@DravidianLangTech-EACL2021: Offensive language detection from multilingual code-mixed text using transformers, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 255–261. URL: https://aclanthology.org/2021.dravidianlangtech-1.35.

Steever, S.B., 2018. Tamil and the dravidian languages, in: The world's major languages. Routledge, pp. 653–671.

Stein, B., 1977. Circulation and the historical geography of tamil country. The Journal of Asian Studies 37, 7–26. doi:10.2307/2053325.

Tula, D., Potluri, P., Ms, S., Doddapaneni, S., Sahu, P., Sukumaran, R., Patwa, P., 2021. Bitions@DravidianLangTech-EACL2021: Ensemble of multilingual language models with pseudo labeling for offence detection in Dravidian languages, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 291–299. URL: https://aclanthology.org/2021.dravidianlangtech-1.42.

Unsvåg, E.F., Gambäck, B., 2018. The effects of user features on twitter hate speech detection, in: Proceedings of the 2nd workshop on abusive language online (ALW2), pp. 75–85.

Vasantharajan, C., Thayasivam, U., 2021. Hypers@DravidianLangTech-EACL2021: Offensive language identification in Dravidian code-mixed YouTube comments and posts, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 195–202. URL: https://aclanthology.org/2021.dravidianlangtech-1.26.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: NIPS.

Warner, W., Hirschberg, J., 2012a. Detecting hate speech on the world wide web, in: Proceedings of the Second Workshop on Language in Social Media, Association for Computational Linguistics, Montréal, Canada. pp. 19–26. URL: https://www.aclweb.org/anthology/W12-2103.

Warner, W., Hirschberg, J., 2012b. Detecting hate speech on the world wide web, in: Proceedings of the second workshop on language in social media, pp. 19–26.

Waseem, Z., 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter, in: Proceedings of the first workshop on NLP and computational social science, pp. 138–142.

Waseem, Z., Hovy, D., 2016. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter, in: Proceedings of the NAACL Student Research Workshop, Association for Computational Linguistics, San Diego, California. pp. 88–93. URL: https://www.aclweb.org/anthology/N16-2013, doi:10.18653/v1/N16-2013.

Xiang, B., Zhou, L., 2014. Improving Twitter sentiment analysis with topic-based mixture modeling and semi-supervised training, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Baltimore, Maryland. pp. 434–439. URL: https://www.aclweb.org/anthology/P14-2071, doi:10.3115/v1/P14-2071.

Xiang, G., Fan, B., Wang, L., Hong, J., Rose, C., 2012. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, New York, NY, USA. p. 1980–1984. URL: https://doi.org/10.1145/2396761.2398556, doi:10.1145/2396761.2398556.

Xiang, T., MacAvaney, S., Yang, E., Goharian, N., 2021. Toxccin: Toxic content classification with interpretability. arXiv:2103.01328.

Xu, Z., Zhu, S., 2010. Filtering offensive language in online communities using grammatical relations, in: Proceedings of the Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference, pp. 1–10.

Yang, M., 2021. Maoqin @ DravidianLangTech-EACL2021: The application of transformer-based model, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 281–286. URL: https://aclanthology.org/2021.dravidianlangtech-1.40.

Yasaswini, K., Puranik, K., Hande, A., Priyadharshini, R., Thavareesan, S., Chakravarthi, B.R., 2021. IIITT@DravidianLangTech-EACL2021: Transfer learning for offensive language detection in Dravidian languages, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 187–194. URL: https://www.aclweb.org/

anthology/2021.dravidianlangtech-1.25.

Yin, W., Zubiaga, A., 2021. Towards generalisable hate speech detection: a review on obstacles and solutions. arXiv:2102.08886.

Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z., Çöltekin, Ç., 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020), in: Proceedings of the Fourteenth Workshop on Semantic Evaluation, International Committee for Computational Linguistics, Barcelona (online). pp. 1425–1447. URL: https://www.aclweb.org/anthology/2020.semeval-1.188.

Zhao, Y., Tao, X., 2021. ZYJ123@DravidianLangTech-EACL2021: Offensive language identification based on XLM-RoBERTa with DPCNN, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv. pp. 216–221. URL: https://aclanthology.org/2021.dravidianlangtech-1.29.