Hate and Offensive Keyword Extraction from CodeMix Malayalam Social Media Text Using Contextual Embedding

Mariya Raphel, Premjith B, Sreelakshmi K
Amrita School of Artificial Intelligence,
Coimbatore,
Amrita Vishwa Vidyapeetham, India
b_premjith@cb.amrita.edu

Bharathi Raja Chakravarthi
School of Computer Science,
University Of Galway,
Galway, Ireland

Abstract

This paper focuses on identifying hate and offensive keywords from codemix Malayalam social media text. As part of this work, a dataset for hate and offensive keyword extraction for codemix Malayalam language was created. Two different methods were experimented to extract Hate and Offensive language (HOL) keywords from social media text. In the first method, intrinsic evaluation was performed on the dataset to identify the hate and offensive keywords. Three different approaches namely – unigram approach, bigram approach and trigram approach were performed to extract the HOL keywords, sequence of HOL words and the sequence that contribute HOL meaning even in the absence of a HOL word. Five different transformer models were used in each of the approaches for extracting the embeddings for the ngrams. Later, HOL keywords were extracted based on the similarity score obtained using the cosine similarity. Out of the five transformer models, the best results were obtained with multilingual BERT. In the second method, multilingual BERT transformer model was fine tuned with the dataset to develop a HOL keyword tagger model. This work is a new beginning for HOL keyword identification in Dravidian language – Malayalam.

1 Introduction

Social networking sites are the platforms where users can create their own profiles and communicate with other users regardless of any kind of limitations. The freedom to share any content on social media led to the rise of hate and offensive posts on online social media (OSN) (Bharathi and Agnusimmaculate Silvia, 2021; Bharathi and Varsha, 2022; Swaminathan et al., 2022). Hate and offensive posts pose a severe risk to victims' physical and mental health and lead to serious consequences (Chakravarthi, 2022a,b; Kumaresan et al., 2022). This emphasis the importance of automatically detecting hate and offensive content from social media (Sreelakshmi et al., 2021), (Chakravarthi et al., 2023).

The identification of words which make the text hate or offensive is even more critical because it helps to restrict users from posting as well as reading comments containing such words. Therefore, the automatic extraction of the keywords from a social media post has the equal significance of detecting hate content from a social media post. HOL keyword extraction models are available in some languages. However, such models are not yet implemented in Dravidian languages like Malayalam, Tamil, Kannada etc. This task is challenging in Dravidian languages because Dravidian languages are abundant in morphology and can generate numerous word forms by joining a sequence of morphemes to the root word (Chakravarthi et al., 2022a,b; Chakravarthi, 2023). Besides, the social media posts are codemixed and low-resource for Dravidian languages, which poses other challenges in developing an automatic keyword extraction model. Despite of the challenges, developing a HOL keyword extraction model for Dravidian languages is necessary due to its increased use in social media.

Developing a model on codemix data is really challenging. The unavailability of an annotated dataset for HOL keyword extraction on codemix data was the other main challenge. Therefore, we developed an annotated dataset where all the HOL words are labelled in each social media text. Further, we prepared a dictionary of hate and offensive words. Thus, through this work, we addressed the main challenge which hindered any research in HOL keyword extraction in Malayalam by developing the HOL keyword extraction dataset. Later, we performed an intrinsic evaluation on the dataset using five different multilingual sentence transform-

Third Workshop on Speech and Language Technologies for Dravidian Languages,
pages 10–18, Varna, Bulgaria, Sep 7, 2023.
https://doi.org/10.26615/978-954-452-085-4_002

ers.

This paper investigates the efficacy of different multilingual transformer-based embedding models for automatically extracting the keywords from Malayalam codemixed social media posts. We experimented three approaches namely, unigram approach, bigram approach and trigram approach for this. Unigram approach was meant for extracting HOL keywords. In addition, we used the transformer-based models to identify the multiword expressions that make a sentence which does not have a hate or offensive word, hate or offensive text. Here, we define the multiword expression as a sequence of two words (bi-gram) or three words (tri-gram). We considered the intrinsic evaluation scheme in all these approaches for detecting the keywords and multiword expressions from a social media comment. Likewise we developed a transformer based model (Vaswani et al., 2017) and performed various analysis to evaluate the efficacy of our model.

The major contributions of this paper are:

- A model for extracting keyword/multiword expression from social media posts in Malayalam codemix text.
- An annotated dataset for detecting hate and offensive keywords from social media posts in Malayalam codemix text.
- A comparison between the performance of different multilingual transformer models on identifying HOL keyword/multiword expression.
- A transformer-based model for HOL keyword identification

2 Related Works

Hate and offensive content is a pervasive and developing social trend because of the surge of social media technology usage. There are many works related to hate and offensive language identification.

In (Hande et al., 2021), the work was on identifying the offensive language in the low-resourced code-mixed Dravidian languages - Tamil, Kannada, and Malayalam. They constructed a dataset by transliterating all the code-mixed texts into the respective Dravidian language and then pseudo labels were generated for it. They used different pretrained language for extracting the embeddings and then it was given to recurrent neural networks. The best results were obtained when they used the ULM fit model. Their model gave an F1 score of 0.79 for Tamil-English, 0.96 for Malayalam-English and 0.73 for Kannada-English.

In (Sreelakshmi et al., 2021), the authors developed three deep neural architectures for offensive language identification in Dravidian languages. The first architecture was a hybrid model including a convolutional layer, a Bi-LSTM layer and a hidden layer. The second architecture contains a Bi-LSTM and the third architecture contains a Bi-RNN. They mainly focused on the code-mixed Tamil-English, Malayalam-English and Kannada-English for their work. On evaluation, the hybrid model gave them the best results with an F1 score of 0.64 for Tamil- English, 0.90 for Malayalam-English and 0.65 for Kannada-English.

Various approaches were used by different works for identifying hate and offensive language identification. However, we couldn't find any works on hate or offensive keyword extraction from Dravidian languages. Lack of HOL keyword annotated dataset is one of the main reason behind it. On the contrary, works on hate and offensive keyword extraction exists for languages except Dravidian.

In (Sarracén and Rosso, 2023), the work was on extracting Offensive keyword from English comments. OffensEval 2019 and OffensEval 2020 were the datasets used for this work. The offensive keyword extraction was done based on the attention mechanism of BERT and the eigenvector centrality using a graph representation. On testing, they obtained an F1 score of 0.5687 on Off20-OFF19 and 0.5798 on Off19-OFF20.

In (Pamungkas et al., 2022), the authors investigated the role of swear words in detecting the abusive language. They proposed the guidelines for tagging the HOL keywords. They developed a swear word abusive dataset for English language using twitter comments. They also performed certain intrinsic evaluations such as sequence labelling on their dataset. They obtained an f1 score of 0.75 for non-abusive swear word, 0.42 for abusive swear word and 0.99 for not a swear word.

In (Martinc et al., 2022), the authors proposed Transformer-based Neural Tagger for Keyword Identification (TNT-KID) to extract one or multiword phrase which represents the key aspects of a document. For this task, they collected a dataset of scientific abstracts and extracted keywords. According to their work, keyword tagging task was modeled as a binary classification task and predict if a word in the sequence is a keyword or not. The

model was trained and tested accordingly and they obtained an F1-score of 0.63.

Though the HOL keyword extraction was put forward by few works, it is not yet implemented in Dravidian languages. This was the major research gap that motivated us for our work. It is necessary to develop a HOL keyword extraction model for Dravidian languages, because, it is widely spoken in south India and commonly used in social media for posting comments. Hence our work focuses on implementing HOL keyword extraction in Dravidian language. As mentioned earlier, lack of dataset was the main challenge for this. Therefore, we created a HOL keyword dataset for code-mixed Malayalam language. Thus we tackled the prime cause for the research gap. Our work is a new beginning for HOL keyword extracton in Dravidian language.

3 DATASET

Since dataset for HOL keyword extraction were not existing for Malayalam language, creation of the dataset was our prime motive. Tweets and YouTube comments were the sources of data. We extended the existing 'HASCO' dataset (Chakravarthi et al., 2020) to create our new HOL keyword dataset. The HASOC dataset comprises of code-mixed malayalam comments labelled as 'Hate' or 'Not Hate'. We focused on the negative comments (labelled as 'Hate') for finding the HOL keywords. The dataset consisted of 8943 comments. Out of that 3092 comments were of HOL nature and remaining 5851comments were normal. On analysing the negative comments in perspective of HOL keywords, we could notice two types of negative comments. We categorised the comments into two. The first category consisted of negative comments with a HOL word(s). The second category contained negative comments which did not have a HOL word in it.

Keyword annotations process then was narrowed down to first category. The dataset creation (annotation) steps are illustrated in Fig. **??**.

The first step was to search for HOL keywords from the comments belonging to category one. For the ease of identifying negative words, we created a custom list of HOL keywords from swear word website.The HOL words in each comment was then identified by referring to this custom list. In order to label the identified HOL keywords, we followed the guidelines proposed by (Pamungkas et al., 2022). According to this, each offensive/hate keyword was tagged using and . If any comments contain multiple negative words, that comments can be replicated to mark those words. We tagged the keywords accordingly to create the final annotated dataset. The final list of HOL list contained 1082 words in it. The test set used in this work comprised of 756 comments.

4 Methodology

We followed two methodologies for extracting hate and offensive keyword from codemix Malayalam social media text. The first method involves an intrinsic evaluation whereas the second method follows a transformer based approach.

4.1 Data Preprocessing

Preprocessing was performed on the annotated dataset. This step focuses on conversion of letters into lower case, punctuation removal, emoji removal and username removal. Python's built-in package "`re`" was used for the removal of punctuations and username.

4.2 Method - 1:

Our first methodology involves the following steps as illustrated below in Fig. 1. The overall model has four main stages, namely, dataset creation, preprocessing, extracting embedding and HOL identification.

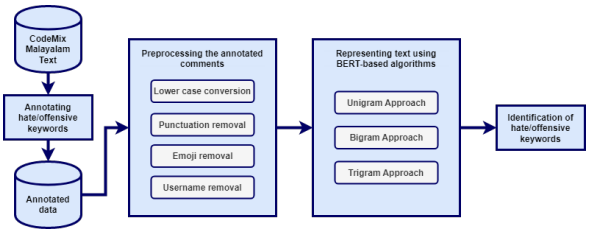

Figure 1: Proposed Architecture - 1

4.2.1 Generate Embeddings

Embedding helps to represent a word and its semantic information in a vector format. Words that are close in vector space are likely to have similar meanings. Therefore, in this work, we have generated embeddings for the n-grams to represent them in the vector space. The model will be able to segregate the HOL words and other normal words as the HOL words will be closer in the vector space. Various BERT based multilingual algorithms were used for representing the text and generating the

embeddings. Five different sentence transformers (Devlin et al., 2018) (Sanh et al., 2019) (Das et al., 2022) (Kakwani et al., 2020) were used for the same.

At this stage, three approaches were followed for generating the embeddings - Unigram, bigram and trigram

4.2.2 Unigram Approach

In this approach, we considered words or unigrams obtained by tokenizing the input text. Later, the word embeddings were generated using the embedding models mentioned above. This embedding can be matched against the embeddings of the hate and offensive dictionary words.

4.2.3 Bigram Approach

In this approach, a two-word sequence or bigrams were considered. Bigrams were obtained by using an overlapping window approach over the comments. The window size was two and the overlapping size was one. Embedding of the bigrams were then generated using the chosen embedding models. These embeddings were used to compare with the embeddings of the hate and offensive dictionary words in the later stages. Thus, during the final identification phase in this approach, a sequence of two words will be predicted.

4.2.4 Trigram Approach

In this approach, a three-word sequence or trigrams were considered. We used the the overlapping window approach to obtain trigram with minor modifications in window size. In order to generate trigrams, the window size was set to three and the overlapping size was one. Later, the word embeddings were generated using the embedding models. These embeddings were used to match against the embeddings of the hate and offensive dictionary words. According to this approach, a sequence of three words will be predicted during the identification phase.

The bigram and trigram approaches were done to extract the sequence of HOL words from the comments. Similarly, there may be comments which does not contain a hate word. However the whole sentence might contribute a HOL context. In order to tackle these two possibilities, bigram approach and trigram approach were introduced.

4.3 Identify Hate and Offensive Keywords

The hate and offensive keywords were identified using similarity score. Cosine similarity was used for this purpose. After generating the embeddings for both the list of hate words and the tokenised comments (ngrams), cosine similarity between each of the ngrams and the hate word was calculated. Based on this similarity score, top five words (ngrams) were extracted as the hate words. Let W denotes the word vector and H denotes the hate word vector, then the cosine similarity can be given as:

$$CosineSimilarity(W, H) = \frac{W.H}{||W||||H||} \quad (1)$$

4.4 Method - 2:

Our second methodology follows a transformer based approach. Fig. 2 illustrates the steps involved in this method.

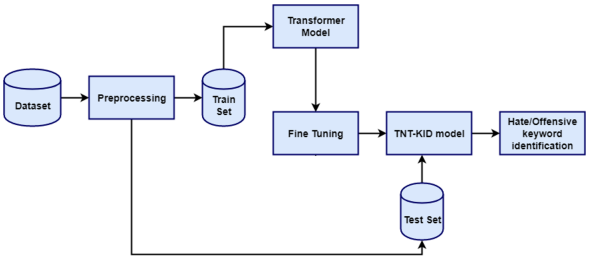

Figure 2: Proposed Architecture - 2

We started with our annotated HOL keyword dataset. The preprocessing step was same as in the method - 1. The dataset was then split into train set and test set. Train set was used developing and fine tuning our transformer model.

4.4.1 Model Development and Fine Tuning:

Pre-trained 'bert-base-multilingual-cased' transformer model was used as the base model in this method. The task of identifying HOL keywords was modelled as a binary class token classification problem. Therefore, our model have a task-specific output layer on top of the transformer model. Since it is a token classification task, the value at each index position in the output vector denotes whether the token at that index position in the comment vector is a hateword/offensive or normal.

We prepared two lists based on the train set. The first list contained all the comments of the train set. The second list contained the HOL words if any present in the corresponding sentence. The lists were named as 'sentences' and 'hatewords' respectively. Later both the lists were tokenised and the embeddings were generated. Comments in the 'sentences' list were padded to form a vector of length 256. Subsequently, label vectors of length

256 were prepared. The values in the label vectors were 1s and 0s. 1 represents HOL token and 0 represents a normal token.

'FULL_FINETUNING' was set to true in our model. Therefore, all the parameters of the pre-trained model were fine-tuned except the parameters like gama, bias etc. The weight_decay_rate was set to 0.01 for the non-normalization parameters and to 0.0 for the normalization parameters. Normalization parameters (e.g., gamma and beta) are typically used to scale and shift the outputs of a layer during training to improve performance. These parameters are not typically fine-tuned because they are usually set to some reasonable initial values and then fixed during training to prevent overfitting. This is because these parameters control the normalization of the activations across training examples, and overfitting on this normalization can lead to poor generalization performance on new data. By fixing these parameters during training, the model can learn better representations that are less dependent on the normalization parameters and thus more likely to generalize to new data.

5 Experiments and Results

In each approach of method-1, the comments were tokenized to form ngrams. Five different sentence transformers were used to generate the embedding of the ngram. Finally, cosine similarity was employed for finding the HOL keyword.

The models were trained with 3425 HOL comments and 6174 normal comments. Later, they were tested with 84 HOL comments. As this is a task where HOL words are detected, the no.of comments in the test set might not create any bias. The 84 test cases were prepared meticulously. The test cases included only those comments which contained HOL words that weren't listed on the custom list. This was done to know how efficiently the model could recognize unseen HOL words.

The performance of five different sentence transformers in extracting HOL words are compared in this section.

5.1 Unigram Approach

The results obtained on testing unigrams using different transformers are as follows:

5.1.1 BERT-base-multilingual-cased

Some sample results of this model for comments typed in English alone, Malayalam alone and including both are given in Fig. 3. For multilingual BERT it can be see that, the expected words are present in the predicted for first and second comments. But missed one word in last case. But still, since the expected words can be found in the predicted, this can be considered as a good performance.

Comment	Expected Words	Predicted Words
അയ്യേ നട്ടെല്ലില്ലാത്ത അവതാരക... അഭിമാനം എന്നൊന്നില്ലാത്ത അവതാരക...	അയ്യേ, നട്ടെല്ലില്ലാത്ത, അഭിമാനം എന്നൊന്നില്ലാത്ത	'നട്ടെല്ലില്ലാത്ത', 'എന്നൊന്നില്ലാത്ത', 'അയ്യേ', 'അവതാരക', 'അഭിമാനം'
ഇൻസൾട്ട് ചെയ്ത്താൽ അടുത്ത് ഗോൾഡൻ ബമ്പർ കിട്ടും .. സേച്ചി ഒന്നും തോന്നരുത് podii പിത്തക്കാടി	podii, പിത്തക്കാടി	'podii', 'പിത്തക്കാടി', 'തോന്നരുത്', 'ബമ്പർ', 'ഇൻസൾട്ട്'
poi chavadi ena. Oro durandangal. Online class kanum vegam vitto	poi chavadi, Oro durandangal	'chavadi', 'durandangal', 'kanum', 'vegam', 'ena'

Figure 3: Unigram - bert-base-multilingual-cased

5.1.2 distilbert-base-multilingual-cased

For multilingual distilbert, some of the expected words were present in the predicted list. On analysing the similarity scores it was seen that, the non hate words got a higher score than the hate words in the comments which include both English and Malayalam. However, this can also be regarded as a fair performance as the expected words were predicted.

5.1.3 Hate-speech-CNERG/indic-abusive-allInOne-MuRIL

This model follows a similar pattern as that of multilingual disitil bert. Some of the expected words were found in the predicted lists But for the comments typed in Malayalam alone, the non hate word has got a higher score than hate words.

5.1.4 ai4bharat/indic-bert

On inspecting the words predicted by this model, it was noted that few of the expected keywords were missed in the comments typed in English. Similarly non hate words got a higher score than the hate words in the comments typed in Malayalam.

5.1.5 Hate-speech-CNERG/malayalam-codemixed-abusive-MuRIL

In case of codemixed-abusive-MuRIL, a satisfactory result was not obtained for any of the comments.The model is not predicting well for English based comments. Therefore, for the comments typed in English alone and comments typed in English-malyalam, the predictions were not as desired. Even for the comments typed in Malayalam alone, non hate words had higher scores. In fact the scores are very similar or very high for the all

words . This can be one of the causes for not distinguishing hate and non hate words properly and for the incorrect predictions.

The prediction accuracy of each model on testing with unigram approach is given in Table 1

5.2 Bigram Approach

In this approach, sequences of two words were predicted. Performance of the sentence transformers in the bigram approach has a similar pattern as that of the unigrams approach. The comments that do not contain a hate word, but a negative context are presented below. The results obtained on testing bigrams using different transformers are as follows:

5.2.1 BERT-base-multilingual-cased

The predictions obtained by this model is given in Fig. 4. On inspecting the predictions, it can be found that the sequences predicted by multilingual BERT contributes a negative meaning for all the three comments.

Comment	Expected Words	Predicted Words
ഈ മുഖഭാഷം കണ്ട് ആരും അത്ഭുധപെടേണ്ട . സന്തോഷ് പണ്ഡിറ്റ് നേക്കാളും കിടുവ	സന്തോഷ് പണ്ഡിറ്റ് നേക്കാളും കിടുവ	'നേക്കാളും കിടുവ', 'അത്ഭുധപെടേണ്ട സന്തോഷ്', 'പണ്ഡിറ്റ് നേക്കാളും', 'ആരും അത്ഭുധപെടേണ്ട', 'മുഖഭാഷം കണ്ട്
No violence only peace. ഗോപാലൻ പണ്ടേ പീസിന്റെ ആളാണല്ലോ	പീസിന്റെ ആളാണല്ലോ	'പണ്ടേ പീസിന്റെ', 'പീസിന്റെ ആളാണല്ലോ', 'ഗോപാലൻ പണ്ടേ', 'peace ഗോപാലൻ', 'only peace'
Ikka fansinte 25 k dislike undallo... dislike adichittu karanjolu	adichittu karanjolu	'adichittu karanjolu', 'dislike undallo', 'undallo dislike', 'dislike adichittu', 'ikka fansinte'

Figure 4: Bigram - BERT-base-multilingual-cased

5.2.2 distilbert-base-multilingual-cased

Multilingual distilbert model was able to extract negative sequences from the comments typed in English alone and Malayalam alone. Whereas, for the comment typed in English-Malayalam s.a discrepancy was seen in the predicted sequences.

5.2.3 Hate-speech-CNERG/indic-abusive-allInOne-MuRIL

The performance of this model using bigram approach was not satisfactory. The similarity scores of the bigrams were higher and closer. Hence, the predictions obtained were not accurate.

5.2.4 ai4bharat/indic-bert

The performance of this model was similar to that of the previous model. Even in this model, the predictions were not satisfactory because the model predicted incorrect bigram sequences as HOL for the comments.

5.2.5 Hate-speech-CNERG/malayalam-codemixed-abusive-MuRIL

Wrong predictions were obtained for the comments typed in Malayalam alone and English alone. Comparatively, better predictions were obtained for the comment typed in English and Malayalam.

The prediction accuracy of each model on testing with bigram approach is given in Table 2 On analysing the performance of different sentence transformer models in the bigram approach, it can be concluded that the multilingual bert outperforms the other models.

5.3 Trigram Approach

In trigram approach, sequences of three words were predicted. Performance of the sentence transformers in trigram approach follows a similar pattern as that of the bigram approach. The results obtained on testing trigrams using different transformers are as follows:

5.3.1 BERT-base-multilingual-cased

The predictions obtained by this model is given in Fig. 5. On inspecting the predictions, it can be found that the correct sequences were predicted by multilingual bert and they contributed a negative meaning for all the three comments.

Comment	Expected Words	Predicted Words
ഈ മുഖഭാഷം കണ്ട് ആരും അത്ഭുധപെടേണ്ട . സന്തോഷ് പണ്ഡിറ്റ് നേക്കാളും കിടുവ	സന്തോഷ് പണ്ഡിറ്റ് നേക്കാളും കിടുവ	'ആരും അത്ഭുധപെടേണ്ട സന്തോഷ്', 'പണ്ഡിറ്റ് നേക്കാളും കിടുവ', 'കണ്ട് ആരും അത്ഭുധപെടേണ്ട', 'അത്ഭുധപെടേണ്ട സന്തോഷ് പണ്ഡിറ്റ്', 'സന്തോഷ് പണ്ഡിറ്റ് നേക്കാളും'
No violence only peace. ഗോപാലൻ പണ്ടേ പീസിന്റെ ആളാണല്ലോ	പണ്ടേ പീസിന്റെ ആളാണല്ലോ	'പണ്ടേ പീസിന്റെ ആളാണല്ലോ', 'ഗോപാലൻ പണ്ടേ പീസിന്റെ', 'peace ഗോപാലൻ പണ്ടേ', 'only peace ഗോപാലൻ', 'violence only peace'
Ikka fansinte 25 k dislike undallo... dislike adichittu karanjolu	dislike adichittu karanjolu	'dislike adichittu karanjolu', 'undallo dislike adichittu', 'k dislike undallo', 'dislike undallo dislike', 'ikka fansinte 25'

Figure 5: Trigram - BERT-base-multilingual-cased

5.3.2 distilbert-base-multilingual-cased

Multilingual distilbert gave a good result in trigram approach when compared with the bigram approach. It predicted the negative sequences from all the three types of comments.

5.3.3 Hate-speech-CNERG/indic-abusive-allInOne-MuRIL

The performance of this model in trigram approach follows the same pattern as that of its bigram approach. The results obtained with this model in the trigram approach is not satisfactory as the predictions obtained were inaccurate due to the higher similarity scores of all the trigrams.

Model	# of predicted hate words	Prediction accuracy
BERT-base-multilingual-cased	85	76.58%
distilbert-base-multilingual-cased	89	80.18%
ai4bharat/indic-bert	76	68.47%
Hate-speech-CNERG/indic-abusive-allInOne-MuRIL	73	65.76%
Hate-speech-CNERG/malayalam-codemixed-abusive-MuRIL	60	54.05%

Table 1: Results of Unigram-based approach

Model	# of predicted hate words	Prediction accuracy
BERT-base-multilingual-cased	100	86.20%
distilbert-base-multilingual-cased	97	83.62%
ai4bharat/indic-bert	88	75.86%
Hate-speech-CNERG/indic-abusive-allInOne-MuRIL	87	75%
Hate-speech-CNERG/malayalam-codemixed-abusive-MuRIL	71	61.20%

Table 2: Results of Bigram-based approach

5.3.4 ai4bharat/indic-bert

On inspecting the predictions obtained with this model, it was seen that the predictions obtained for the comments typed using English script and Malayalam script were better than the predictions generated for the comments typed using English-Malayalam.

5.3.5 Hate-speech-CNERG/malayalam-codemixed-abusive-MuRIL

The similarity scores of trigrams obtained with this model were too close and higher. The model couldn't discern the hate/offensive sequences. Hence, incorrect predictions were obtained in most of the cases.

The prediction accuracy of each model on testing with trigram approach is given in Table 3 On comparing the performance of different sentence transformer models in the trigram approach, it is evident that the multilingual bert model performs better than the other models.

6 Performance of Method-2

This section presents the results of method-2. The train set was divided for training and validation in the ratio 8:2. The model was trained and validated for 30 epochs using Adam and RMSprop optimizers with learning rate 3e-5. Later the model was tested on the test set comprising of 84 comments. The performance of the model on the comments typed in English, comments typed in malayalam and comments including both Malayalam and English were evaluated. Table 4 and Table 5 denote the results obtained for Adam and RMSprop optimizers respectively. A validation accuracy of 88.80% was obtained for the former model and 89.65% was attained for the latter. The best results were obtained with Adam optimizer.

7 Discussion

On analysing the dataset and results obtained in method-1, it was seen that in most of the cases the hate words had a similarity score >=0.7. On analysing the performance of transformer models, we could see that the best results were obtained with multilingual bert.

Multilingual bert is trained on a large corpus and it has a huge number.of parameters when compared with the other models. This can be one of the reasons behind the excellent performance of the model in all the three categories of comments (typed in English alone, malayalam alone and including both).

Multilingual bert is followed by Multilingual distilbert whose performance is a bit lower in the category of comments including both English and Malayalam. Multilingual distilbert is a distilled version of multilingual bert with lower number of parameters. This can be the cause for lower performance of multilingual distilbert when compared with the multilingual bert.

The indic-abusive-allInOne-MuRIL model has a lower performance in the category of comments typed in Malayalam alone when compared with the other two categories. But we can see that indic-abusive-allInOne-MuRIL and multilingual distilbert showed a similar pattern of fair performance and it is better than indic bert.

The indic-bert gave a medium performace in all the three categories of comments, but lower than

Model	# of predicted hate words	Prediction accuracy
BERT-base-multilingual-cased	88	84.62%
distilbert-base-multilingual-cased	87	83.65%
ai4bharat/indic-bert	86	82.69%
Hate-speech-CNERG/indic-abusive-allInOne-MuRIL	85	81.73%
Hate-speech-CNERG/malayalam-codemixed-abusive-MuRIL	71	68.27%

Table 3: Results of Trigram-based approach

Comment Type	# of predicted hate words	Prediction accuracy
English	24	70.58%
Malayalam	23	45.09%
Eng-Mal	10	52.63%

Table 4: Results of Model with Adam Optimizer

Comment Type	# of predicted hate words	Prediction accuracy
English	20	58.82%
Malayalam	23	45.09%
Eng-Mal	8	42.11%

Table 5: Results of Model with RMSprop Optimizer

the above 3 models. It can be due to the fewer number of parameters in this model. And finally, the malayalam-codemixed-abusive-MuRIL, gave a lower-than expected result. Even though this model is trained on Malayalam codemix abusive language, the performance was not as expected.

On comparing bigram and trigram approaches, bigram approach yields a better result than the latter.

Based on the results obtained in method-2, we could see that (the best performance of the model was evident on English based comments. Comparatively good performance was seen on Malayalam comments and English-Malayalam comments.) Though we evaluated the performance of the model with various optimizers, Adam optimizer was performing best on our dataset.

8 Conclusion and future work

In order to fulfil the gap of lack of annotated data for HOL keywords in Malayalam, a dataset was created for the same as part of this paper. Later, the hate and offensive keywords in the dataset were identified based on cosine similarity using unigram approach. Apart from this, hate and offensive sequences were extracted from the sentences even in the absence of a hate word using bigram and trigram approach. On comparing the performance of various sentence transformer models, "`bert-base-multilingual-cased`" turned out to be the best model for extracting hate keywords from code-mix Malayalam social media text.

Being the best model, the "`bert-base-multilingual-cased`" was utilized for developing the transformer model in the second method. Based on the results obtained in method-2, the best performance of the model was evident on English based comments. Comparatively good performance can be seen on Malayalam comments and English-Malayalam comments.

As a future work, the explainability concept(Peyrard et al., 2021)can be employed to improve the performance of the current model. Also, the performance of indic-abusive-allInOne-MuRIL and malayalam-codemixed-abusive-MuRIL models can be further investigated. Likewise, the effect of different dialects can be analysed to know its role in identifying HOL keywords.

Acknowledgments

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).

References

B Bharathi and A Agnusimmaculate Silvia. 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*, pages 313–318, Kyiv. Association for Computational Linguistics.

B Bharathi and Josephine Varsha. 2022. SSNCSE NLP@TamilNLP-ACL2022: Transformer based approach for detection of abusive comment for Tamil language. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 158–164, Dublin, Ireland. Association for Computational Linguistics.

Bharathi Raja Chakravarthi. 2022a. Hope speech detec-

tion in Youtube comments. *Social Network Analysis and Mining*, 12(1):75.

Bharathi Raja Chakravarthi. 2022b. Multilingual hope speech detection in English and Dravidian languages. *International Journal of Data Science and Analytics*, 14(4):389–406.

Bharathi Raja Chakravarthi. 2023. Detection of homophobia and transphobia in Youtube comments. *International Journal of Data Science and Analytics*, pages 1–20.

Bharathi Raja Chakravarthi, Adeep Hande, Rahul Ponnusamy, Prasanna Kumar Kumaresan, and Ruba Priyadharshini. 2022a. How can we detect homophobia and transphobia? experiments in a multilingual code-mixed setting for social media governance. *International Journal of Information Management Data Insights*, 2(2):100119.

Bharathi Raja Chakravarthi, Manoj Balaji Jagadeeshan, Vasanth Palanikumar, and Ruba Priyadharshini. 2023. Offensive language identification in dravidian languages using mpnet and cnn. *International Journal of Information Management Data Insights*, 3(1):100151.

Bharathi Raja Chakravarthi, Anand Kumar M, John P McCrae, B Premjith, KP Soman, and Thomas Mandl. 2020. Overview of the track on hasoc-offensive language identification-dravidiancodemix. In *FIRE (Working notes)*, pages 112–120.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Thenmozhi Durairaj, John McCrae, Paul Buitelaar, Prasanna Kumaresan, and Rahul Ponnusamy. 2022b. Overview of the shared task on homophobia and transphobia detection in social media comments. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 369–377, Dublin, Ireland. Association for Computational Linguistics.

Mithun Das, Somnath Banerjee, and Animesh Mukherjee. 2022. Data bootstrapping approaches to improve low resource abusive language detection for indic languages. In *Proceedings of the 33rd ACM Conference on Hypertext and Social Media*, pages 32–42.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Adeep Hande, Karthik Puranik, Konthala Yasaswini, Ruba Priyadharshini, Sajeetha Thavareesan, Anbukkarasi Sampath, Kogilavani Shanmugavadivel, Durairaj Thenmozhi, and Bharathi Raja Chakravarthi. 2021. Offensive language identification in low-resourced code-mixed dravidian languages using pseudo-labeling. *arXiv preprint arXiv:2108.12177*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Prasanna Kumar Kumaresan, Rahul Ponnusamy, Elizabeth Sherly, Sangeetha Sivanesan, and Bharathi Raja Chakravarthi. 2022. Transformer based hope speech comment classification in code-mixed text. In *International Conference on Speech and Language Technologies for Low-resource Languages*, pages 120–137. Springer.

Matej Martinc, Blaž Škrlj, and Senja Pollak. 2022. Tnt-kid: Transformer-based neural tagger for keyword identification. *Natural Language Engineering*, 28(4):409–448.

Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2022. Investigating the role of swear words in abusive language detection tasks. *Language Resources and Evaluation*, pages 1–34.

Maxime Peyrard, Beatriz Borges, Kristina Gligorić, and Robert West. 2021. Laughing heads: Can transformers detect what makes a sentence funny? *arXiv preprint arXiv:2105.09142*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Gretel Liz De la Peña Sarracén and Paolo Rosso. 2023. Offensive keyword extraction based on the attention mechanism of bert and the eigenvector centrality using a graph representation. *Personal and Ubiquitous Computing*, 27(1):45–57.

K Sreelakshmi, B Premjith, and Soman Kp. 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*, pages 249–254.

Krithika Swaminathan, Bharathi B, Gayathri G L, and Hrishik Sampath. 2022. SSNCSE_NLP@LT-EDI-ACL2022: Homophobia/transphobia detection in multiple languages using SVM classifiers and BERT-based transformers. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 239–244, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.