EDIOne@LT-EDI-EACL2021: Pre-trained Transformers with Convolutional Neural Networks for Hope Speech Detection.

Suman Dowlagar
LTRC
IIIT-Hyderabad
suman.dowlagar
@research.iiit.ac.in

Radhika Mamidi
LTRC
IIIT-Hyderabad
radhika.mamidi
@iiit.ac.in

Abstract

Hope is an essential aspect of mental health stability and recovery in every individual in this fast-changing world. Any tools and methods developed for detection, analysis, and generation of hope speech will be beneficial. In this paper, we propose a model on hope-speech detection to automatically detect web content that may play a positive role in diffusing hostility on social media. We perform the experiments by taking advantage of pre-processing and transfer-learning models. We observed that the pre-trained multilingual-BERT model with convolution neural networks gave the best results. Our model ranked 1^{st}, 3^{rd}, and 4^{th} ranks on English, Malayalam-English, and Tamil-English code-mixed datasets.

1 Introduction

Nowadays, people often use social media websites to share their views and thoughts. The thoughts might be positive or negative. Much work has been done towards identifying the negative thoughts, i.e., hate speech and offensive content identification on social media (Schmidt and Wiegand, 2017; Davidson et al., 2017). Research is now shifting to the field of analyzing positivity via hope speech detection on social media.

Hope is a positive state of mind, an expectation of positive outcomes concerning events and circumstances in one's life (Youssef and Luthans, 2007). Hope drives an individual to move forward. Hope can be a useful tool for each individual to maintain a stable and optimistic attitude towards life.

In a multilingual society, people usually express their thoughts by mixing two or more languages in a single utterance. This form of language contact is known as code-mixing (Di Sciullo et al., 1986). Code-mixed data is real-world unprocessed data that has non-standard variations of spelling and does not follow a grammatical structure (Bali et al., 2014). Any automated hope-speech detection tool will face challenges in this aspect. So the analysis of code-mixed hope speech detection is necessary to handle the real-time social media data.

The European Association of Computational Linguistics 2021 has organized a Language Technology workshop for Equality, Diversity, and Inclusion (Chakravarthi and Muralidaran, 2021) with a shared task to cultivate positivity and promote research on code-mixed hope speech data. The goal of this task is to identify whether a given comment contains hope-speech or not.

In this work, we address the issue of hope-speech detection on code-mixed Dravidian youtube comments. This paper presents a pre-trained multilingual BERT encoder with CNN as a classifier for the hope speech classification data.

The paper is organized as follows. Section 2 provides related work on hope speech detection. Section 3 provides information on the task and datasets Section 4 describes the proposed work. Section 5 presents the experimental setup and the performance of the model. Section 6 concludes our work.

2 Related Work

Hope speech detection is a novel topic with a significantly limited amount of research done in this field.

(Palakodety et al., 2019) propose a novel task to automatically detect hope speech on web content that may play a positive role in diffusing hostility on social media triggered by heightened political tensions during a conflict between two nuclear power nations.

(Chakravarthi, 2020) created a multilingual, hostility-diffusing hope speech dataset for equality, diversity, and inclusion. It is a new large-scale English, Tamil (code-switched) dataset, and Malay-

Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pages 86–91
April 19, 2021.

Dataset	#Train	#Dev	#Test	#Total
English	22762	2843	2846	28451
Malayalam-English	8564	1070	1071	10705
Tamil-English	16160	2018	2020	20198

Table 1: Data Statistics

alam (code-switched) YouTube comments. They have experimented on the dataset by using traditional machine learning classifiers.

3 Task Description

We need to identify the hope in the code-mixed English, Tamil-English, and Malayalam-English youtube comments for hope speech detection. For the English language, data was collected related to the following issues, including women in STEM, LGBTIQ issues, COVID-19 pandemic, Racism, and Black Lives Matters, United Kingdom (UK) versus China, United States of America (USA) versus China, and Australia versus China from YouTube video comments. For Tamil and Malayalam, we collected data from India on the recent topics regarding LGBTIQ issues, COVID-19, women in STEM, the Indo-China border dispute. Each comment or post is annotated with hope-speech or non-hope-speech and not-Tamil/not-Malayalam labels. The dataset is divided into train, development, and test sets for the given hope speech task. The details of the dataset are given in the table 1.

4 Our method

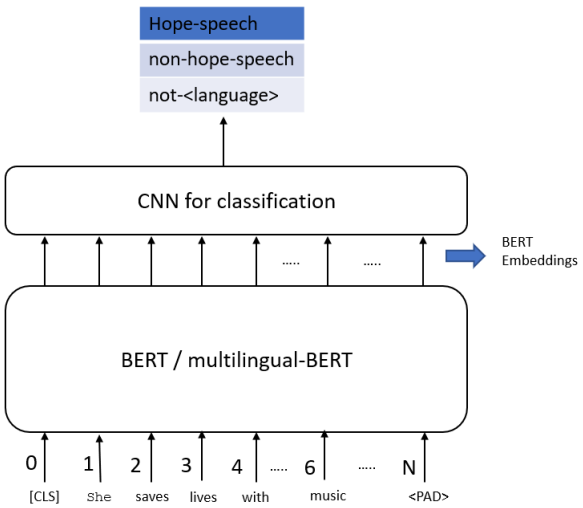


Figure 1: Pretrained BERT with CNN for Hope Speech Detection

In this section, we present the pre-processing and the use of pre-trained multilingual BERT embeddings with CNN classification model for hope speech identification.

4.1 Pre-processing

The data set is a code-mixed real-time dataset, and it has issues related to transliterated script with variations in the spelling, excessive use of emoticons, mentions, and hashtags in the corpus. Pre-processing will help to solve these issues to some extent. While pre-processing,

1. We removed all punctuations, emotions, hashtags
2. We back-transliterated the script to its native language using linguistic rules, transliteration, and language detection libraries.

4.2 Pre-trained BERT model for embeddings

Bi-directional Encoder Representations with Transformers (BERT) (Devlin et al., 2018) is a language model. BERT uses a transformer (Vaswani et al., 2017), multi-headed attention with a point-wise feed-forward network, to learn contextual relations between words (or sub-words) in a text. In its vanilla form, the transformer includes two separate mechanisms, an encoder that reads the text input and a decoder that produces a prediction for the task.

Using BERT, each sentence is tokenized using a sub-word level tokenizer and uses a layer of transformers to encode the word in a sentence into a vector of size 1x768, where 768 is the length of BERT embedding. The BERT uses [CLS] token to indicate the beginning of the sentence and [SEP] token to indicate the end of the sentence. In our approach, we used a small version of the pre-trained multilingual BERT model called *bert-base-multilingual-cased* obtained from the transformers library (Wolf et al., 2019).

4.3 Convolutional Neural Network (CNN)

CNN is a category of Neural Networks that uses convolution and pooling operations and performs

Classifier	Hope	Non-Hope	not-English	macro-F1	weighted-F1	Acc
SVM	0.53	0.95	0.00	0.49	0.91	0.91
SVM *processed*	0.53	0.95	0.00	0.49	0.91	0.91
mBERT	**0.54**	0.96	0.00	0.53	0.92	0.92
mBERT *processed*	**0.54**	0.96	0.00	0.53	0.92	0.92
Our approach	**0.54**	**0.97**	0.00	**0.54**	**0.94**	**0.94**

Table 2: Classification performance of Our approach w.r.t Baselines on English Data

Classifier	Hope	Non-Hope	not-Malayalam	macro-F1	weighted-F1	Acc
SVM	0.61	0.88	0.70	0.73	0.81	0.81
SVM *processed*	0.60	0.87	0.71	0.73	0.80	0.80
mBERT	**0.66**	0.91	0.83	0.80	0.86	0.86
mBERT *processed*	0.65	**0.92**	**1.00**	**0.84**	0.86	**0.87**
Our approach	**0.66**	**0.92**	**1.00**	**0.84**	**0.87**	**0.87**

Table 3: Classification performance of Our approach w.r.t Baselines on Malayalam Data

Classifier	Hope	Non-Hope	not-Tamil	macro-F1	weighted-F1	Acc
SVM	0.57	0.61	0.60	0.57	0.57	0.57
SVM *processed*	0.46	0.60	0.59	0.55	0.55	0.55
mBERT	0.57	0.56	0.58	0.57	0.56	0.56
mBERT *processed*	**0.51**	0.64	**0.60**	**0.59**	**0.60**	**0.59**
Our approach	**0.51**	**0.65**	**0.60**	**0.59**	**0.60**	**0.59**

Table 4: Classification performance of Our approach w.r.t Baselines on Tamil Data

parameter sharing. Compared to its predecessors, CNN's main advantage is that it automatically detects the important features using convolution operations.

In our paper, we used the convolutional model developed for sentence classification by (Kim, 2014). In this CNN, We take a word embedding obtained of size (n,d), where n is the number of tokens in a sentence, and d is the embedding dimensionality. We apply convolution operations on those embeddings with three kernels of sizes (2,d), (3,d), and (4,d). We consider these kernel sizes as it takes combinations of 2, 3, 4 tokens and extracts the feature representation. We use a ReLU activation function after each convolutional layer. Then we apply max-pooling over convolutions to down-sample the input representation and to avoid overfitting. Concatenate the kernels of different sizes. Then we pass those convolutions through a forward feed network for output representation and add a dropout layer to avoid overfitting.

5 Implementation

We give the input text to the pre-trained multilingual BERT model, and the pre-trained BERT model gives encoded information as output. Now we need to learn a classification model for the given encoded information. We pass the encoded information as embeddings to the above CNN classifier. The classifier applies convolutions, max-pooling, and finally, a feed-forward network for classification with dropout.

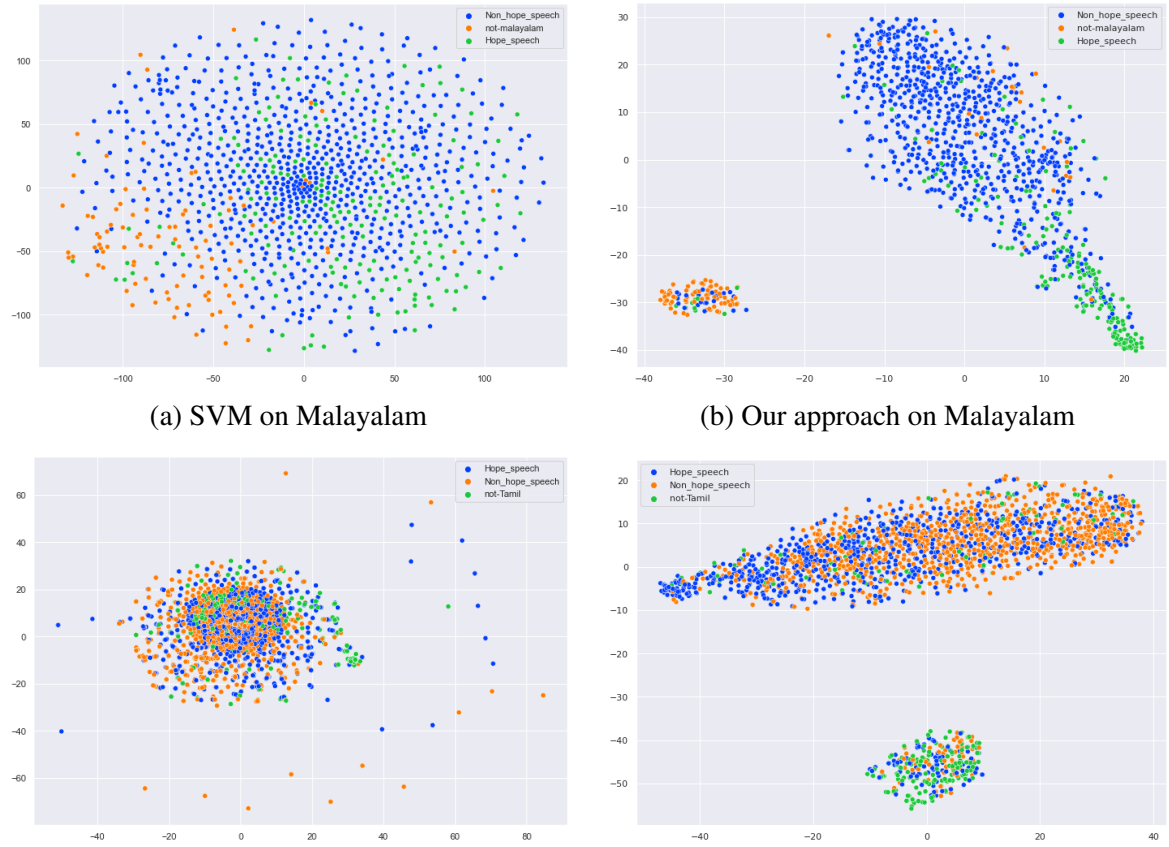
6 Experiments

The section presents the baselines, hyper-parameter settings, and analysis of observed results.

The baselines used for the proposed work is,

1. **SVM with TF-IDF and sub-word level tokenization:** This baseline uses term frequency and inverse document frequency-based vectorization (Ramos et al., 2003) for feature representation and the support vector machine (Cortes and Vapnik, 1995) to classify the data. The code-mixed data contains transliterated text with a non-standard representation of words. We used sub-word-level tokenization to extract the tokens to capture a better sentence representation.

2. **Pre-trained multilingual BERT (mBERT):**



(a) SVM on Malayalam

(b) Our approach on Malayalam

(c) SVM on Tamil

(d) Our approach on Tamil

Figure 2: Comparison of SVM classifier and our approach on Tamil Data

This baseline uses a pre-trained multilingual BERT model with a feed-forward network for classification.

Note: We consider two formats, *processed* and original data for comparing the baselines, where "processed" refers to applying the pre-processing method on the original data.

6.1 Hyperparameters and libraries

Bert is a non-regressive model and uses positional embeddings along with token embeddings, so BERT specifies the model to consider the default sentence length of 512. In our model, we kept the default sentence length as 128. The number of class labels for classification is given as three (hope_speech, non_hope_speech, and not_English // not_Tamil // not_Malayalam). The dropout is kept at 0.5. We train the model for 30 epochs with batch size set to 128 and the learning rate to 1e-5. We use Adam optimizer with the cross-entropy.

We used the ai4bharat-transliteration [1], a deep learning-based transliteration tool to back transliterate the identified Malayalam and Tamil words. We used the hugging face transformers library to download the pre-trained BERT for English data and the pre-trained multilingual BERT model for Malayalam and Tamil data to obtain the encoded representations of the tokens. The PyTorch library is used to model the convolutional neural networks. The TF-IDF feature representation, SVM classifiers, and the classification metrics are obtained from the scikit-learn library. The Sentencepiece (Kudo and Richardson, 2018) achieves the subword level tokenization.

6.2 Results

Tables 2, 3, and 4 present the f1-score and accuracy of the models on the English and Dravidian code-mixed datasets.

From the above results, it is clear that our approach of the multilingual pre-trained BERT model to extract the embeddings and CNN to sentence classification and transliteration based pre-processing to handle the code-mixed data works

[1] https://pypi.org/project/ai4bharat-transliteration/

best for the given datasets.

Preprocessing helped the BERT model to focus on the relevant information and back-transliteration helped the BERT to obtain the embeddings based on the native script.

In SVM "processed" classifier did not perform better after back transliteration because the TF-IDF feature representation used in the SVM classifier and TF-IDF works on statistical counts of words rather than extraction of embeddings based on the native script. When TF-IDF vectorization was performed on the original data, the feature vector was very big, i.e., there were many unique words in the data. The words present in both native and transliterated forms are mapped to a single word after back transliteration. This reduced the feature space of each word and affected the feature representation of the word.

We have plotted the TSNE distribution 2 on the Malayalam and Tamil data to better visualize our approach compared to the baseline SVM classifier on the original data. As the plots obtained from pre-trained BERT and our approach are visually similar, we compared the SVM and our approach for better visualization. The plots on the Malayalam data show the effectiveness of our approach when compared to the SVM model. We can see a better distinction of classes (*hope-speech, non-hope-speech, and not-Malayalam*) compared to the SVM classifier.

Tamil data shows a very slight distinction between hope-speech and non-hope speech data but a clear distinction of classes for not-Tamil data. The words in the *not-Tamil* label are not back-transliterated into the Tamil language. It ensured the script of such words being different from the Tamil script, followed by different embedding representations of the words, and improved accuracy for the not-Tamil class label.

7 Conclusion

In this paper, we addressed hope speech identification in English and Dravidian code-mixed languages. We used pre-trained multilingual bidirectional encoder representations to obtain the word embeddings, and we used convolutional neural networks for classification. We compared the method with other baselines. The results showed that using back-transliteration helped the model capture the pre-trained word embeddings based on the native script. CNN helped the model extract feature representations better than feed-forward networks, which increased the model's performance. In the future, we will work on cross-lingual meta word embeddings to handle the multilingual scenario in the code-mixed datasets.

References

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "i am borrowing ya mixing?" An Analysis of English-Hindi Code Mixing in Facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126.

Bharathi Raja Chakravarthi. 2020. HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.

Bharathi Raja Chakravarthi and Vigneshwaran Muralidaran. 2021. Findings of the shared task on Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

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*.

Anne-Marie Di Sciullo, Pieter Muysken, and Rajendra Singh. 1986. Government and code-mixing. *Journal of linguistics*, 22(1):1–24.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Shriphani Palakodety, Ashiqur R KhudaBukhsh, and Jaime G Carbonell. 2019. Hope speech detection: A computational analysis of the voice of peace. *arXiv preprint arXiv:1909.12940*.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International workshop on natural language processing for social media*, pages 1–10.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

Carolyn M Youssef and Fred Luthans. 2007. Positive organizational behavior in the workplace: The impact of hope, optimism, and resilience. *Journal of management*, 33(5):774–800.