IIIT_DWD@LT-EDI-EACL2021: Hope Speech Detection in YouTube multilingual comments

Sunil Saumya[1] and **Ankit Kumar Mishra**[2]
[1]Indian Institute of Information Technology Dharwad Karnataka, India
[2]Magadh University, Bodh Gaya, Bihar, India
sunil.saumya@iiitdwd.ac.in, ankitmishra.in.com@gmail.com,

Abstract

Language as a significant part of communication should be inclusive of equality and diversity. The internet user's language has a huge influence on peer users all over the world. People express their views through language on virtual platforms like Facebook, Twitter, YouTube, etc. People admire the success of others, pray for their well-being, and encourage in their failure. Such inspirational comments are hope speech comments. At the same time, a group of users promotes discrimination based on gender, race, sexual orientation, persons with a disability, and other minorities. The current paper aims to identify hope speech comments which are very important to move on in life. Various machine learning and deep learning-based models (such as support vector machine, logistics regression, convolutional neural network, recurrent neural network) are employed to identify the hope speech in the given YouTube comments. The YouTube comments are available in English, Tamil, and Malayalam languages and are part of the task "*EACL-2021:Hope Speech Detection for Equality, Diversity, and Inclusion*".

1 Introduction

In the current era, there are two lives we live; a real one and a virtual one. In a real case, we interact, communicate and exchange our emotions physically. On the other hand in virtual cases, we interact with others through a computer-based simulated environment, social media is one such example. In both cases, to exchange information a language is required such as English, Hindi, Spanish, German, and so on. Through these languages, we share our moments such as happiness, joy, anger, sadness, appreciation for success, motivation on failure, and others which everyone needs in their hard time. These comments convey the well-being of someone and are termed as "hope speech". The other categories of comments that discourage, abuse, demotivate based on gender, race, sexual orientation, persons with a disability, and other minorities are termed as "not hope speech". In the physical scenario, the reachability and effect of these comments mitigate with region and time. On the other hand, on virtual platforms (especially social media) their effects are long-lasting across the geographical region.

Hope speech reflects the idea that you will find paths to the desired goals and be encouraged to use them. A few examples of hope speech on social media platforms are; i) product-leverage reviews, ii) election campaign comments, iii) disaster or crisis-informed decision-making comments, and so on. Since there is a large number of comments and suggestions on a specific topic, the overall difference in the sentiment of the topic is created (Lee et al., 2017). Many times hope speech of a product leads to its great success, for example, introducing a number of helpful votes received by product reviews on Amazon leads to additional $2.7 billion revenue[1]. Similarly, a number of likes, shares, and hope speech comments received by YouTube video can make it the most viewed video and so on. Also, if the video content or quality is not up to the mark, the hopeful suggestions received by other users help to boost the content or consistency of the video. On the other hand, the not hope speech comments can lower the subject's value and demotivates the person. Considering the higher impact of hope and not hope comments in social media contexts, it is therefore important to identify them for an informed decision.

The current paper identifies the hope speech comments on the YouTube platform. The dataset used in this study is a part of "*EACL-2021: Hope Speech Detection for Equality, Diversity, and Inclusion*". The comments are in three different languages English, Tamil, and Malayalam. The problem proposed is a classification problem in three groups, where all comments are marked with hope

[1] https://articles.uie.com/magicbehindamazon/

Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pages 107–113
April 19, 2021. ©2020 Association for Computational Linguistics

or not hope or not in the intended language. For classification of comments, several classification models based on conventional machine learning (e.g. support vector machine, logistic regression), deep learning (e.g. convolutional neural network (CNN), long short term memory network (LSTM)), and hybrid learning (parallel CNN-LSTM network, three parallel Bi-LSTM network) are developed. The results from the experiments showed that a model based on the CNN-LSTM network having 2 parallel layers worked best for the English dataset, with Bi-LSTM having three parallel layers performed best for Tamil and Malayalam [2].

The rest of the article is organized as follows; Section 2 discusses the related works for monolingual dataset classification. Section 3 describes the given task and data dimension. This is followed by the methodology of the current paper in Section 4. The results are explained in Section 5. Finally, Section 6 concludes the paper by highlighting some future scope.

2 Related work

According to the best of our knowledge, this is the first task on hope speech detection. Therefore, there are no previous works in this category. However, the current work is a monolingual classification problem, one of the highly researched problems in the processing of natural languages. In this section, we identify a few state-of-art papers for monolingual classification.

There are many applications of monolingual contexts such as sentiment classification, hate speech identification, fake content identification, and others. (Pak and Paroubek, 2010) performed sentiment analysis of English text from Twitter. They use a multi-nominal naive Bayes classifier to classify each tweet into positive, negative, and neutral. The features used were n-grams and POS-tags. They found that POS-tags are a strong predictor of sentiments. (Saumya and Singh, 2018) used the sentiment analysis approach to identify spam reviews on the e-commerce platform. Three classifiers random forest, gradient boosting, and support vector machine was trained on Amazon reviews. The best performance was reported by random forest with an F1-score of 0.91. (Davidson et al., 2017) proposed a hate speech classifier for English corpus (25000 tweets). (Pitsilis et al., 2018) trained a recurrent neural network on 16K tweet for hate speech identification. A comparative analysis of machine learning, deep learning, and transfer models is presented by (Plaza-del Arco et al., 2021) for the hate speech Spanish dataset. Similarly, for fake news detection on German data (Vogel and Jiang, 2019) used convolutional network and support vector machine classifiers on 4500 news articles. A fake news hostility detection dataset was proposed by (Bhardwaj et al., 2020) in the Hindi language. They trained several machine learning models like logistic regression, support vector machine, random forest, and multilayer perceptron network with m-BERT embedding for classifying hostility contents.

In line with the above literature, the current paper employs several machine learning and deep learning models for the classification of monolingual hope speech comments into three classes. The detailed methodology of the proposed model is explained in the next section.

3 Task and data description

The Hope Speech Content Identification in English, Tamil, and Malayalam language is organized by *EACL-2021* in a track on *Hope Speech Detection for Equality, Diversity, and Inclusion*. The objective of this task is to determine whether or not a comment contains a speech of hope. Every comment is labeled with either hope speech, not hope speech, and not in intended language.

The competition dataset was released in a phased manner. Initially, training and development datasets were released for each English, Tamil, and Malayalam corpus, resulting in six different data files. Every file had two fields; a comment and its label. In the given corpus of English, Tamil and Malayalam, the average comment length was one except for a few cases where it was more than one. The description of both training and development set for all three corpora is shown in Table 1. As shown in Table 1, the current system was trained on 22762, 16160, and 8564 samples and validated on 2843, 2018, and 1070 samples for English, Tamil, and Malayalam respectively. Later, the test dataset was released by organizers on which the final ranking of submitted models was released. The dataset details can also be found in (Chakravarthi, 2020; Chakravarthi and Muralidaran, 2021).

[2] The developed model code can be seen in the github repository: https://github.com/ankitmishra2232/Hope-Speech-Identification

Language	**Hope speech**		**Not hope speech**		**Not in intended language**		**Total**	
	Train	Dev	Train	Dev	Train	Dev	Train	Dev
English	1962	272	20778	2569	22	2	22762	2843
Tamil	6327	757	7872	998	1961	263	16160	2018
Malayalam	1668	190	6205	784	691	96	8564	1070

Table 1: Train and Dev datasets description

4 Methodology

The current paper proposed multi-label classification methods for the identification of Hope speech contents. Several conventional machine learning, deep learning, and hybrid learning approaches were employed to achieve the objective. A detailed description of all proposed methods is discussed in the subsection.

4.1 Data preprocessing

The dataset was preprocessed before feeding it to the machine learning models. The steps used for preprocessing were common for all three languages English, Tamil, and Malayalam. The Preprocessing was applied to the comment field of training and development data. Initially, Roman scripts present in texts were converted into lowercase. Then, all the punctuations, emoji, extra spaces, numbers, and stopwords were removed from the texts. The cleaned texts were then tokenized and encoded into the sequence of token indexes.

4.2 Classification models

The current paper employed several classification models for hope speech detection in YouTube comments. This section first explains the different conventional classifiers, followed by deep learning classifiers and finally hybrid classifiers used in the study.

4.2.1 Conventional machine learning classifier

Three traditional machine learning-based models were developed to categorize YouTube comments, including the Support Vector Machine (SVM), Logistic Regression (LR), and the Random Forest (RF). The input to these classifiers were Tf-idf vectors created from English, Tamil, and Malayalam comments. To acquire the vector, comments were first tokenized using the library *WhiteSpace Tokenizer*. The tokenized data were then stemmed using *Porter stemmer* library. Finally, the stemmed data were vectorized using a *Tf-idf vectorizer*. The models on the Tf-idf vector had very high training times.

4.2.2 Deep learning classifier

In the category of deep networks, the current study utilized convolutional neural network (CNN) and variants of recurrent neural networks namely long short term memory (LSTM) and bidirectional long short term memory (Bi-LSTM) in a single and multiple layers setting. To feed an equal length input to deep networks, the tokenized comments were first encoded and then padded with maximum comment length 10, 20, and 20 for English, Tamil, and Malayalam dataset respectively, because we checked the average words count for the datasets and used as maximum comment length.

The input to deep models was a one-hot vector where every word in a language was represented in its corresponding vocabulary dimensions. For example, in the English corpus vocabulary size was 20373, consequently, every word in English data was represented in a vector of 20373 dimensions (1×20373). The one-hot input representation was a high dimensional sparse vector, having a single'1'(that represented the index of the token), and all zeros. To convert it into a low dimensional dense valued vector, an embedding layer was used that represented every word in a 300 dimensional (1×300) dense vector. While representing one-hot vector into embedded dense vector several pre-trained weights from Word2Vec, GloVe, and random embeddings were used. The obtained embedded vectors were fed to the deep networks (such as single and multiple layers of CNN, LSTM, and Bi-LSTM) for further steps like contextual feature extraction followed by classification.

4.2.3 Hybrid Network

Further, several hybrid settings of deep neural networks were developed such as parallel CNN-CNN,

CNN-LSTM, CNN-Bi-LSTM, LSTM-LSTM, and so on. The architecture of the best two hybrid models is shown in Figure 1 and 2. The input to these networks was embedding vectors same as explained above in Section 4.2.2.

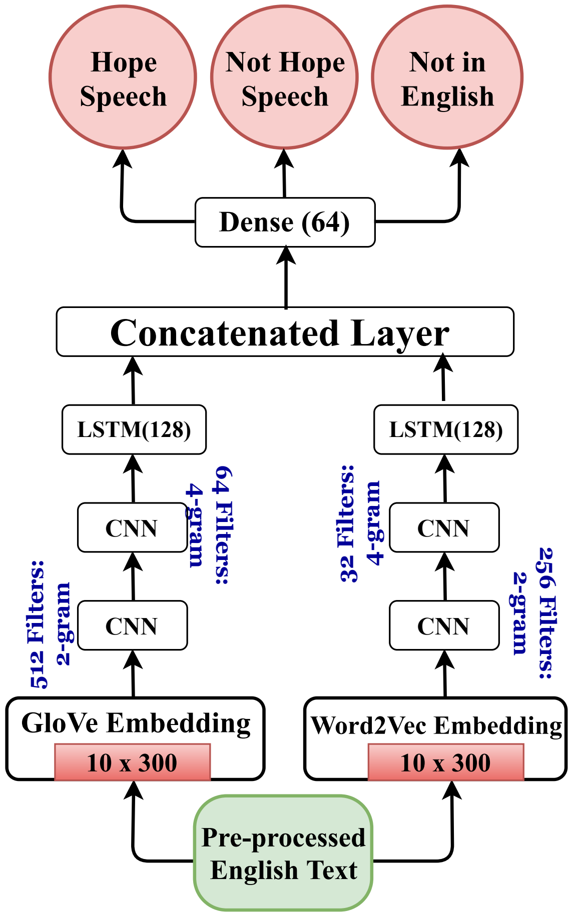

Figure 1: 2 Parallel CNN-Bi-LSTM model for English hope speech classification

The model shown in Figure 1 reported the best accuracy for the English dataset. As it is shown, the number of words fed from the pre-processed text was 10. Every word (having dimension 1 × 20373) was then represented in 300 dimensional embedded vector (1×300) using GloVe and Word2Vec embeddings. The obtained embedded vectors were fed to parallel 2 layered CNN and 1 layered LSTM model (2 parallel CNN-LSTM). The features (or output) extracted from LSTM layers were concatenated in a single vector. This vector was passed as input to a fully connected dense layer to classify every comment into three categories. The model was trained for 100 epochs with the "Adam" optimizer and "binary cross-entropy" loss function.

Alternatively, for Tamil and Malayalam datasets a three parallel Bi-LSTM model performed best (as shown in Figure 2). The input to the first, second, and third Bi-LSTM was character embedding, word embedding, and word embedding respectively. From the corpus of Malayalam, a unique 124 characters were extracted. For every comment in the Malayalam set, a maximum of 70 characters were given as input to the model. To the second and third Bi-LSTM, the input was 300-dimensional word embeddings obtained from GloVe and Word2Vec. All three parallel embedding vectors were then fed to three parallel Bi-LSTM networks. The features extracted from Bi-LSTMs were concatenated and fed to the fully connected dense layer for classification. The model was trained for 100 epochs.

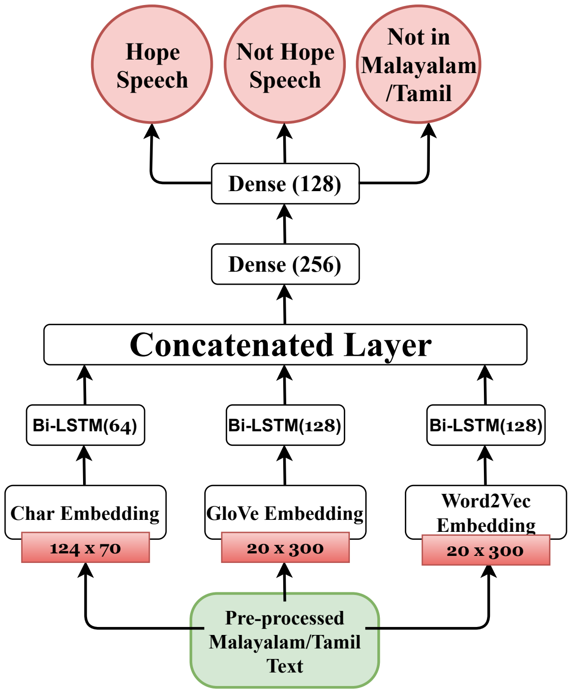

Figure 2: 3 Parallel Bi-LSTM model for Tamil and Malayalam hope speech classification

Similar steps were followed for Tamil hope speech identification except for a few differences that in Tamil corpus total unique characters were 218, total unique words were 32041, and training epochs were 150. In both Tamil and Malayalam cases, the optimizer was *Adam* and loss function was *binary-crossentropy*.

5 Results

As discussed above several classification models were built and experimented. We are reporting the results of the submitted models in the competition and a few other comparable models. All models were built in *Python* using libraries *Sklearn*, *Keras*,

Language	**Type**	**Model**	**Embedding/Feature**	**Dev weighted F1**
English	Conventional learning models	SVM	Tf-idf	0.85
		LR	Tf-idf	0.75
	Deep learning models	LSTM	GloVe	0.90
		Bi-LSTM	GloVe	0.90
		2 Layered CNN	Random	0.79
	Hybrid learning models	3 parallel LSTM	Glove, Word2Vec, Word2Vec	0.90
		2 parallel CNN-LSTM	**GloVe, Word2Vec**	**0.91**
		3 parallel CNN-LSTM	Word2Vec, Word2Vec, Word2Vec	0.88
		2 parallel LSTM-Bi-LSTM	Word2Vec, GloVe	0.90
Tamil	Conventional learning models	SVM	Tf-idf	0.48
		LR	Tf-idf	0.50
	Deep learning models	Bi-LSTM	Random	0.52
		CNN	Word2Vec	0.55
		2 layered CNN	Random	0.55
	Hybrid learning models	2 parallel LSTM	Word2Vec	0.50
		2 parallel CNN	Word2Vec	0.52
		3 parallel Bi-LSTM	**Word2Vec, Random**	**0.56**
Malayalam	Conventional learning models	SVM	Tf-idf	0.65
		LR	Tf-idf	0.72
	Deep learning models	CNN	Word2Vec	0.70
		LSTM	Word2Vec	0.75
		Bi-LSTM	Random	0.74
	Hybrid learning models	2 parallel LSTM	Word2Vec, Random	0.73
		2 parallel Bi-LSTM	Random, Random	0.77
		3 parallel CNN	Word2Vec, Random, Random	0.70
		3 parallel Bi-LSTM	**Word2Vec, Random, Random**	**0.78**

Table 2: Development dataset results of various models

Language	**Model**	**Embedding/Features**	**Dev Weighted F1**	**Test Weighted F1**	**Ranking**
English	2-Parallel CNN-LSTM	GloVe and Word2Vec	0.91	0.90	4
Tamil	3-Parallel Bi-LSTM	Word2Vec and Random	0.56	0.54	8
Malayalam	3-Parallel Bi-LSTM	Word2vec and Random	0.78	0.79	5

Table 3: Development and test dataset results of best performing submitted models

Pandas, *Numpy* an so on. The results reported here are for development data and test data. The metric used to evaluate the performance of a model was the weighted F1-score (or weighted F1). Table 2 shows the experimental results of various machine learning, deep learning, and hybrid learning models on the development dataset. For conventional classifiers SVM, and LR shown in Table 2, the feature used was Tf-idf vector whereas, for deep and hybrid models, features were embeddings.

In conventional models, for the English dataset, SVM reported weighted F1 0.85 which was higher than LR where weighted F1 was 0.75. The performance of the classification was even better for most of the deep and hybrid classifiers. As it is shown in Figure 2, LSTM and Bi-LSTM models with GloVe embedding reported 0.90 weighted F1, but at the same time, the performance of 2 layered CNN model with random embedding was very low as weighted F1 score was 0.79. The best-reported results were from the hybrid model *2-parallel CNN-LSTM* with GloVe and Word2Vec embeddings where weighted F1 was 0.91. The performance of *2-parallel LSTM-BiLSTM* and *3-parallel LSTM* was also comparable as it reported weighted F1 0.90. For the Tamil dataset, in conventional learning, the performance of LR (weighted F1 0.50) was better than SVM (weighted F1 0.48). The best performing models were 2-layered CNN and *3-parallel Bi-LSTM* with weighted F1 0.55 and 0.56 respectively. A similar pattern was observed for Malayalam data also. The best performing model for Malayalam set was *3 parallel Bi-LSTM* with weighted F1 0.78. Even, *2 parallel Bi-LSTM* with random embedding showed comparable results with weighted F1 0.77. The performance of SVM was reported least with a weighted F1 of 0.65.

The best performing models of English, Tamil, and Malayalam dataset was then validated with test dataset provided by organizers. The test dataset contained only a comment field, therefore, the predictions on test data for best models were submitted to the competition. Later, the organizer evaluated the submitted prediction label with the actual label of test data. The test dataset results are shown in Table 3. As it is reported in Table 3, on the English test dataset the weighted F1 was 0.90. For Tamil and Malayalam, the weighted F1 was 0.54 and 0.79 respectively. In the last column of Table 3, the final ranking of our models obtained in the competition for respective languages is shown.

As can be seen from Table 2 and 3, weighted F1 reported on development data and test data are almost equal. That verifies the authenticity of model training. The other observation is for the similar model *3-parallel Bi-LSTM* the classification accuracy of Malayalam hope speech was higher but for the Tamil dataset, it was only 0.54. That means to learn Tamil hope speech comment some other features and model can be investigated.

6 Conclusion

The current paper identified hope speech contents in YouTube comments using several machine learning, deep learning, and hybrid models. The task was proposed by *EACL 2021* for three different languages English, Tamil, and Malayalam. Every comment was categorized in one of three classes hope speech, not hope speech, and not in intended language. On the English dataset, the best performing model was *2-parallel CNN-LSTM* with GloVe and Word2Vec embeddings and it reported weighted F1 0.91 and 0.90 for development and test set respectively. Similarly, the best performing model of Tamil and Malayalam was *3-parallel Bi-LSTM*. For Tamil it reported weighted F1 0.56 and 0.54 on development and test dataset respectively. Similarly, for Malayalam, the reported weighted F1 was 0.78 and 0.79 on the development and test dataset.

References

Flor Miriam Plaza-del Arco, M Dolores Molina-González, L Alfonso Ureña-López, and M Teresa Martín-Valdivia. 2021. Comparing pre-trained language models for spanish hate speech detection. *Expert Systems with Applications*, 166:114120.

Mohit Bhardwaj, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. 2020. Hostility detection dataset in hindi. *arXiv preprint arXiv:2011.03588*.

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 Lan-*

guage Technology for Equality, Diversity and Inclusion. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*.

Jong Hyup Lee, Sun Ho Jung, and JaeHong Park. 2017. The role of entropy of review text sentiments on online wom and movie box office sales. *Electronic Commerce Research and Applications*, 22:42–52.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326.

Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence*, 48(12):4730–4742.

Sunil Saumya and Jyoti Prakash Singh. 2018. Detection of spam reviews: a sentiment analysis approach. *Csi Transactions on ICT*, 6(2):137–148.

Inna Vogel and Peter Jiang. 2019. Fake news detection with the new german dataset “germanfakenc”. In *International Conference on Theory and Practice of Digital Libraries*, pages 288–295. Springer.