# Hope Speech Detection: Identifying Positive Actors in Toxic Discourses

Mohammad Aflah Khan
IIIT Delhi
aflah20082@iiitd.ac.in

Diksha Sethi
IIIT Delhi
diksha20056@iiitd.ac.in

Neemesh Yadav
IIIT Delhi
neemesh20529@iiitd.ac.in

Raghav Sahni
IIIT Delhi
raghav20533@iiitd.ac.in

## Abstract

*Health specialists believe that hope is significant for a person's well-being, recovery, and restoration of human life. Hope speech expresses the belief that one may discover routes to their desired goals and be inspired to take such courses. With the rise of the Internet, more and more people have started seeking support on social media platforms. Hope speech detection refers to analysing texts on social media that can invoke positive emotions in people. Through our project, we aim to detect Hope Speech which reinforces positivity in an online setting. This problem is relevant as, alongside penalising bad actors, we can reward good actors if we can identify them. In this paper, we experiment with various ML and DL techniques and analyse the results obtained. All our codes and experiments can be found here: https://github.com/aflah02/Hope_Speech_Detection*

## 1. Introduction

With the rise of the Internet, there has been a significant increase in the number of marginalized people seeking support online. Online social media comments and posts have been analyzed using tools like hate speech recognition, offensive language identification, and abusive language detection to find and limit the spread of negativity. However, these studies primarily focus on analyzing negativity in the English Language, but the problem is not just restricted to harmful content. Research must also focus on encouraging and supportive online content as a form of positive reinforcement. Through our project, we aim to detect Hope Speech which reinforces positivity in online discourse. This problem is relevant as, alongside penalizing bad actors, we can reward good actors if we can identify them. Some downstream applications of our project include using the classifier to curate more data which can be used to train generative models. These models can then be deployed in toxic-online settings to spread positivity.

Overall, we propose building explainable classifiers for Hope Speech Detection, which can classify text into three classes “Hope Speech”, “Non-Hope Speech”, and “Non-English” for the English language. Then we plan to train the same models on Malayalam and Tamil to analyze the results and the shortcomings of directly using models in multi-lingual settings. We also plan to utilize Machine Learning models for this task as that is a relatively unexplored area. The few works on the subject discussed in the next section primarily focus on Deep Learning based methods and show minimal use of ML Techniques

## 2. Literature Review

Hope Speech and Models for Hope Speech Detection The first known work on Hope Speech [1] defines Hope Speech in a very limited fashion. Their work focused on analyzing trends in Youtube Comments during peak hostile times between India and Pakistan. For them Hope Speech is “Web content which plays a positive role in diffusing hostility on social media triggered by heightened political tensions”. Our work is more closely related to [2] where the

definition is much broader and inclusive. The work defines Hope Speech as “Youtube comments/posts that offer support, reassurance, suggestions, inspiration and insight” and also provides a broad list of guidelines to annotate such speech. The work also creates the first publicly available dataset HopeEDI which provides data for 3 languages English, Tamil and Malayalam. We use this dataset for our work. Subsequent works emerged in the First Workshop On Language Technology For Equality, Diversity, Inclusion (LT-EDI-2021) where Hope Speech Detection was one of the tasks and several works were submitted in the competition. The finding paper [3] released after the Workshop discusses the various models and the best performing ones. The best performing models use Deep Learning based architectures such as BERT [4], RoBERTa [5], XLM-RoBERTa [6] etc. while traditional methods appear to lag. Hope Speech Detection again emerged as a Task in the Second Workshop on Language Technology For Equality, Diversity, Inclusion (LT-EDI-2022) where datasets for 2 new languages Spanish and Kannada were also added. The conclusions from the shared task are summarized in [7]. Another notable difference between the 2 Workshops is the change in evaluation metrics. The First Workshop used Weighted F1 while the Second Workshop used Macro F1 as the ranking metric which led to numerically lower but more meaningful scores. In our work we use both of these scores.

## 3. Dataset

### 3.1. EDA

Class Distribution: There are three main classes in the dataset (Non-Hope Speech, Hope- Speech and Non-English).

| Class | Number of Samples |
|---|---|
| Non-Hope Speech | 25940 |
| Hope-Speech | 2484 |
| Non-English | 27 |

Clearly the dataset is heavily biased. This was a major concern while selecting models to train on it. Moreover, balancing will be needed.

|  | English | Tamil | Malayalam |
|---|---|---|---|
| Train | 22762 | 16160 | 8564 |
| Development | 2843 | 2018 | 1070 |
| Test | 2846 | 2020 | 1071 |
| Total | 28451 | 20198 | 10705 |

WordCloud: The word clouds for non-hope speech and hope-speech did not have any significant differences showing that the task at hand is not trivial and cannot be done simply by seeing the words in a sentence. However, the Non-English word cloud was quite different from the rest.

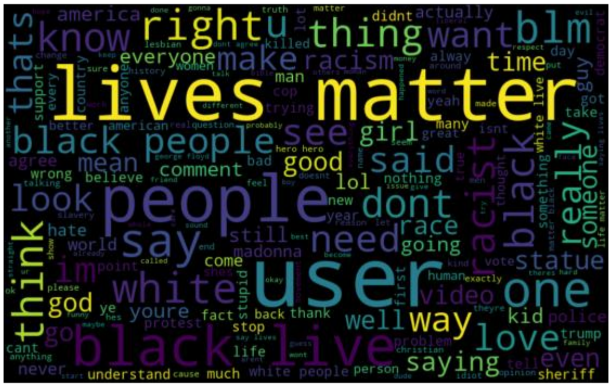


Figure 0: Word Cloud Non-Hope speech

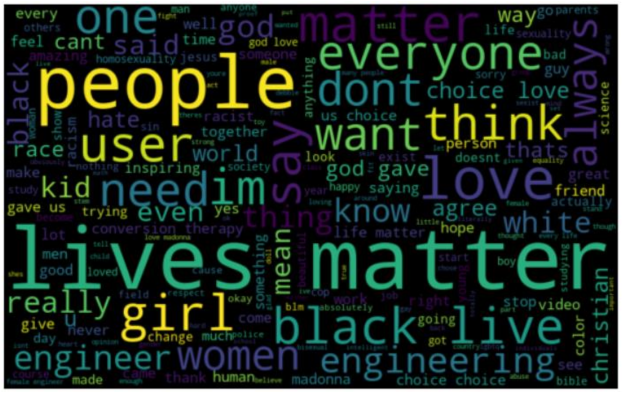


Figure 1: Word Cloud Hope speech

Sentiment Analysis: We also performed sentiment analysis on the data, even here for non-hope speeches there was no significant difference in the sentiment scores. For hope speech, however, a positive score was much more common than a negative or zero scores.

Topic Modeling: We perform topic modeling using BERTopic and evaluate the top-5 topics. We see clear identification markers amongst the topics. The topics captured in order are: Racism, Homosexuality, Women in Tech, References to Madonna, some General Words.

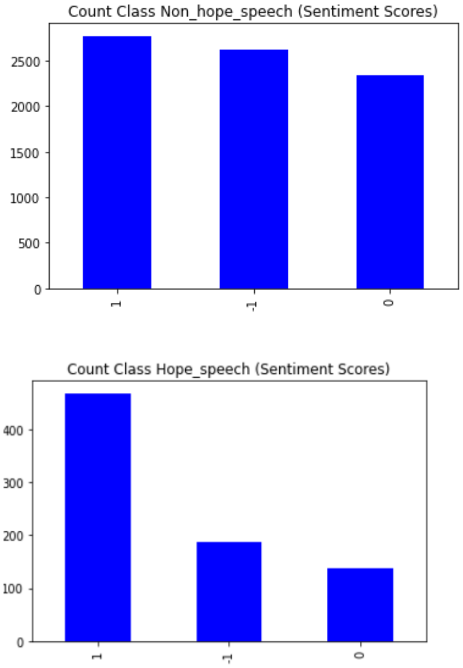


Figure 2: Sentiment Analysis on Hope Speech and Non-Hope Speech

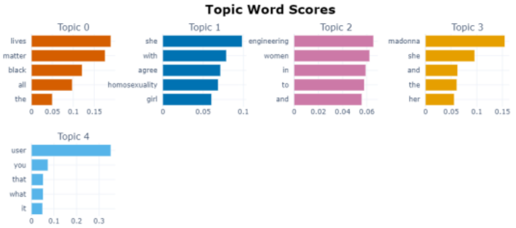


Figure 3: Topics captured using BERTopic

### 3.2. Preprocessing

To ensure that all the text inputs were uniform, we performed a few preprocessing steps. This was done to make sure that any of these textual inconsistencies would not affect the output in any way. The following steps were performed in the order in which they are listed:

1. Lower Case : To ensure that words with different case styles are not perceived in different contexts, we have lowercase each term in our dataset
2. Removal of URLs
3. Removal of usernames
4. Punctuation Removal
5. Removal of whitespaces
6. Removal of emojis : Used the emoji library for the operation
7. Tokenization: We have broken down the text sentences into smaller chunks or units using the Tweet Tokenizer. This step would essentially help to understand the meaning of the text by analysing smaller units at a time.

Several words such as usernames and URLs and special characters such as punctuation marks and emojis have been removed because we do not require them in our context.

3.3. Word Embeddings

At their core Word Embeddings are just ways to represent sentences as Vectors. There are numerous ways to convert sentences into their embeddings for our work we decide to use -

1. TF-IDF a statistical technique which tries to estimate how important each word is to a document in a collection of documents. We also use PCA to reduce the maximum number of dimensions to 1000.
2. Pretrained GloVe (2B tweets, 27B tokens, 1.2M vocab, uncased) [8], FastText (1 million word vectors, 16B Tokens) [9] and Word2Vec Embeddings (Uses subset of Google News dataset, contains 300-dimensional vectors for 3 million words and phrases) [10] using Gensim Library [11].
3. We also use 2 Transformer based pretrained embeddings, all-mpnet-base-v2 and all-MiniLM-L6-v2 from SBERT [12]. The former is the best pretrained model while the latter is the fastest. Subsequently we also perform PCA and create 2 new embeddings which retain 95% of the variance and also use them for our study.

We decide to stick to these Word Embeddings as they cover a wide range of techniques and different eras of Word Embeddings. TF-IDF derives its roots from Information Retrieval. GloVe, FastText and Word2Vec are some of the most used embeddings prior to the wide scale adoption of transformer based embeddings and SBERT is now one of the most used libraries by NLP researchers to get pre trained embeddings as empirically and theoretically Transformer Embeddings heavily outperform other embeddings in many tasks due to their Context Capturing Power.

3.4. Data Augmentation

Since the dataset was highly unbalanced, we performed data augmentation to generate samples for the "Hope Speech" class as it had very few samples in comparison to the other classes. We used various augmentation techniques such as synonym replacement, random word substitution, insertion, deletion, cropping and swapping. Each technique doubles the corpus size, and after performing each, we had a dataset of about 42,000 samples, on which we performed our experiments.

## 4. Methodology

We used the already existing dataset made available by [1] to perform our experiments. We performed the experiments on the English posts from the original dataset. We kept experiments on other languages (Tamil, and Malayalam) to be our downstream task. The task for this dataset is stated to be a three-way classification task where we try to predict each of the {Hope_Speech, Non_Hope_Speech, Other_Languages} labels.

For our experiments, we tried out different word embedding techniques (GloVe [8], FastText [9], word2vec [10], TF-IDF, and Sentence-BERT [12]) and also tried various combinations with them by performing PCA or leaving them as is, to see if we can retain some amount of data while also compressing the dimensions, which we have reported in our final results. After getting the final embeddings we dumped them for future use.

We define two tasks:

1. Task 1: Multiclass Hope Speech Detection; In this task, we categorize the tweets into three classes, Hope, Non-Hope and Non-English

2. Task 2: Two class classification; In this task we categorize the tweets as Hope and Non-Hope speech and drop the "Non-English" class. [7]

Initially, we started off by exploring various traditional ML techniques. Each of us then took up different types of Classifier models from sklearn, and performed the stated task using all the embeddings thus generated. Further, we used grid-search for all the classifiers. We also explored a number of deep learning models such as RNNs and LSTMs. Lastly, we experimented with advanced transformer models such as BERT, HateBERT and

BERTweet. We performed Task 1 on all the classifiers, however, for Task 2 we used our top five performing models from Task 1 as well as transformer-based models. We have reported the Accuracy, {Weighted, Macro, Micro} {F1, Recall, and Precision} scores for each of our experiments.

## 5. Result and Analysis

### 5.1. Results

Task 1: Multiclass Classification

| Algorithm | Embedding | Weighted F1 Score |
|---|---|---|
| Linear Discriminant Analysis | better-no-pca | 0.927859 |
| MLP | better-pca | 0.926217 |
| Logistic Regression | better-no-pca | 0.92167 |
| Perceptron | better-no-pca | 0.920376 |
| XGBoost | better-no-pca | 0.918274 |
| KNN | better-pca | 0.917781 |
| Random Forests | better-no-pca | 0.910008 |
| Naive Bayes | better-pca | 0.906833 |
| Decision Trees | better-pca | 0.906336 |
| ExtraTrees Classifier | better-no-pca | 0.903285 |
| Quadratic Discriminant Analysis | better-pca | 0.89442 |
| AdaBoost | better-pca | 0.886988 |
| Nearest Centroid | tf-idf | 0.811931 |
| K Means | word2vec | 0.688256 |
| K Medoids | better-no-pca | 0.544692 |
| SVM | better-pca | 0.927057 |

Task 2: Binary Classification- Hope and Non-Hope Speech

| Algorithm | Embedding | Macro F1 Score |
|---|---|---|
| HateBERT | N.A. | 0.75969 |
| BERT | N.A. | 0.75518 |
| BERTweet (for Augmented Data) | N.A. | 0.71254 |
| BERT (for Augmented Data) | N.A. | 0.704991 |
| HateBERT (for Augmented Data) | N.A. | 0.68723 |
| LSTM (for Augmented Data) | N.A. | 0.64246 |
| RNN (for Augmented Data) | N.A. | 0.6382 |
| LSTM | N.A. | 0.61447 |
| RNN | N.A. | 0.5683 |
| Linear Discriminant Analysis | better-no-pca | 0.56220 |
| SVM (for Augmented Data) | better-no-pca | 0.50919 |
| BERTweet | N.A. | 0.47700 |
| MLP | better-pca | 0.466631 |
| Logistic Regression | better-no-pca | 0.44476 |

| | | |
|---|---|---|
| Perceptron | better-no-pca | 0.41278 |

5.2. Analysis

Our LDA Model is the best performing model closely followed by MLP, Logistic Regression, Perceptron and XGBoost. When we compare with the SOTA which is an XLM-RoBERTa model we see we are in the same neighborhood with the best model only having a difference of 0.23 F1 Points. MLP, Logistic Regression and perceptron all cross the 0.92 Mark which was the score obtained by the 2nd Ranked Teams in the 2021 Shared Task. This tells us that proper care was not taken by the teams in using Word Embeddings and non-DL Models. The "Better Embedding" from Sentence BERT consistently performs well for our use case.

In comparison with the 2022 edition of the shared task we outperformed the SOTA Macro F1 score by large margins. The SOTA reported is 0.55 while our models shoot way past that into the high 0.7+ range. We attribute this success to the use of BERT and other specialized transformer networks which are able to capture great semantic meaning.

We also investigated the loss and accuracy curves for various configurations 2 of which are shown here –

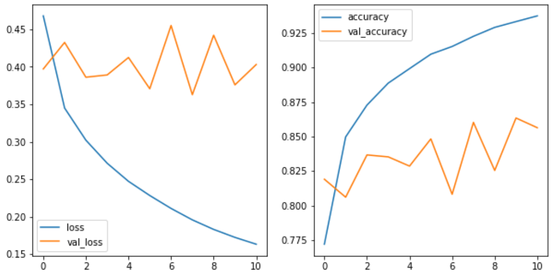


Figure 4: Loss Curve for LSTM (for Augmented Data)

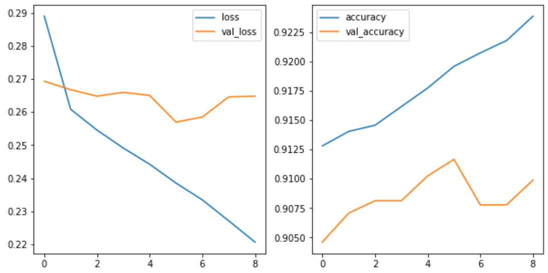
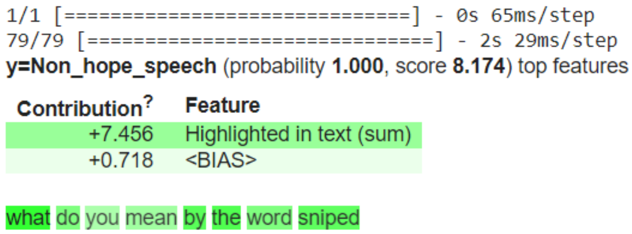


Figure 4: Loss Curve for LSTM (for Original Dataset)

As we use early stopping, training stops only after a few epochs. In both cases we see the curves for augmented data are much more unstable which might be due to differences in augmented and actual text.

5.3. Explainability

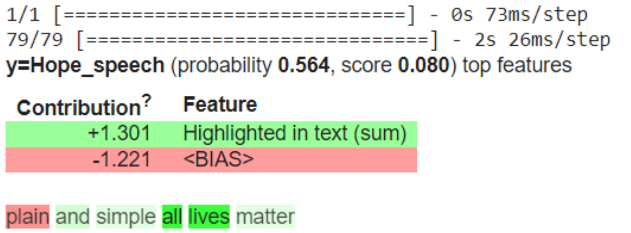


```
1/1 [==============================] - 0s 65ms/step
79/79 [==============================] - 2s 29ms/step
```

**y=Non_hope_speech** (probability **1.000**, score **8.174**) top features

| Contribution? | Feature |
|---|---|
| +7.456 | Highlighted in text (sum) |
| +0.718 | <BIAS> |

what do you mean by the word sniped



```
1/1 [==============================] - 0s 73ms/step
79/79 [==============================] - 2s 26ms/step
```

**y=Hope_speech** (probability **0.564**, score **0.080**) top features

| Contribution? | Feature |
|---|---|
| +1.301 | Highlighted in text (sum) |
| -1.221 | <BIAS> |

plain and simple all lives matter

DL models like BERT are black boxes. Thus, explaining why a model came up with a particular prediction is hard. This is why we tried searching for ways to explain why our model came up with a certain prediction. We use KTrain [14] to generate LIME visualizations. The input is randomly perturbed to examine how the prediction changes. This is used to infer the relative importance of different words to the final prediction using a linear interpretable model.

- The GREEN words contribute to the model prediction
- The RED (and PINK) words detract from the model prediction (Shade of color denotes the strength or size of the coefficients in the inferred linear model)

As we can see from the Visualizations the model is able to capture some indicators of hope and non hope speech while also catching some spurious relations.

## 6. Conclusion

The experiments show that our models are much better built than the models which entered the competition. This also shows that the choice of model and also tuning them matters a lot. The choice of embedding for ML Models can also lead to significant gains. The choice of data augmentation also led to significant improvement in some models such as TweetBERT, LSTM and RNN but also degraded some other models.

The degradation might be because the model already has sufficient data and also the augmentations might destroy meaning in some places. We also conduct extensive experiments for both the Task setups to arrive on the best models. Finally we also use LIME visualization to inspect our models and see whether they have actually learnt something or are just pushing spurious relations. Overall our systems beat SOTA and set new benchmarks.

## References

[1] Shriphani Palakodety, Ashiqur R. KhudaBukhsh and Jaime G. Carbonell. Hope Speech Detection: A Computational Analysis Of The Voice Of Peace. In 24th European Conference on Artificial Intelligence - ECAI 2020

[2] Bharathi Raja Chakravarthi. 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, Bharathi Raja Chakravarthi. 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,

[3] Bharathi Raja Chakravarthi and Vigneshwaran Muralidaran. Findings of the Shared Task on Hope Speech Detection for Equality, Diversity, and Inclusion. Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion, pages 61–72

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019, pages 4171–4186

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

[6] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzman, Edouard Grave, Myle Ott, Luke Zettlemoyer and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8440–8451

[7] Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, Subalalitha Cn, John McCrae, Miguel Ángel García, Salud María Jiménez-Zafra, Rafael Valencia-García, Prasanna Kumaresan, Rahul Ponnusamy, Daniel García-Baena and José García-Díaz. Overview of the Shared Task on Hope Speech Detection for Equality, Diversity, and Inclusion. Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion, pages 378 - 388

[8] Jeffrey Pennington, Richard Socher and Christopher D. Manning. GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)

[9] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch and Armand Joulin. Advances in Pre-Training Distributed Word Representations. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)

[10] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space.

[11] Radim Rehurek and Petr Sojka. Software Framework for Topic Modeling with Large Corpora. The LREC 2010 Workshop On New Challenges For NLP Frameworks

[12] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing

[13] Pedregosa et al. Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830, 201