Hinglish-Hindi Hateful and Offensive Comments Detection

Hitesh Bhandari IIIT Delhi
Ananya Dabas IIIT Delhi
Aaloke Mozumdar IIIT Delhi
Akshita Gupta IIIT Delhi

Abstract

Social media platforms are vital for discourse on social and political issues. Automatic hate speech detection ensures that political discourse isn't marred by hateful comments against minority communities and offensive comments. This paper uses HASOC 2021 conference's subtask of binary classification on Hinglish dataset. A ML classifier based model along multilingual transformer architecture will be described for hate speech and offensive language detection purposes.

1 Problem Statement

The proliferation of the internet in India, especially post-2015, has enabled the general populace to voice their social and political opinions on social media platforms. From seeing political arguments and national incidences on television and newspapers, to now, there is now an affordance of dynamic-user interaction by citizens on social media pertaining socio-political and economic issues. This dynamic interaction, however, makes it hard to track hateful and offensive speech. The hateful and offensive content can arise due to party rivalries, caste, religion, personal hate, etc. These comments cause tangible perceptual changes and lead to minority discrimination, besmirching of famous personalities etc. On a personal level, this can cause lead to increased mental stress, emotional outburst and negative psychological impact. Hence these hateful comments need to be stopped. Although there are strict guidelines by social media platforms for posting only appropriate content, inappropriate content spreads too fast to contain them. A real-time hateful and offensive comments detection system will ensure that hate speech doesn't proliferate to an unavoidable situation.

Hate speech has been a serious topic for research, and there have been tremendous advancements in hate detection techniques. The multi-lingual and code-mixed discourse in India poses a unique problem due to scarce data resources. The advent of large pre-trained models has given tremendous boost to NLP-related tasks. But they've shown that real world code-mixing data is scarce to pre-train such models. The existing pre-trained models either cater to English speaking world or have been unable to capture the subtleties of multilingual semantics and sentiments.

Because Hindi is the world's third most spoken language, behind English and Mandarin4, Hinglish, a combination of Hindi and English (3; 13), has recently gained popularity in the Indian subcontinent. Because it is challenging to create a large scale code-mixed dataset, the research has leaned toward creating synthetic code-mixed datasets (12). At the same time, genuine code-mixed data has been demonstrated to outperform synthetically created datasets (11). Hasoc 2021 dataset sufficiently fits our needs.

In the further sections of this paper we will be describing our dataset and show use of various DL models and will introduce HingRoBeRTa for twitter hate detection. Finally, we show some comparisons with state-of-the-art pre-trained models after ablations.

2 Related Work

For automated hate and offensive speech identification, many machine learning and deep learning algorithms have been tried. The bulk of classic machine learning algorithms extract characteristics from voice text such as words, n-grams, lexical and linguistic aspects. Word embedding algorithms have recently been proposed for similar purposes. However, these techniques fall short of capturing the whole context of the speech.

Deep learning algorithms have longer context retention and are scalable by the amount of data given. They have been becoming increasingly popular in text categorization, sentiment analysis, language modelling, machine translation, and other

Figure 1: F1 scores of models for different downstream tasks (8)

Model	LID	POS-UD	POS-FG	NER	Sentiment	HingLID
BERT	78.69	83.70	70.75	79.27	59.16	96.04
m-BERT	82.56	83.68	69.58	76.64	58.42	95.59
XLMRoBERTa	85.93	87.24	70.95	77.01	61.57	95.42
HingBERT	84.44	88.42	71.04	**81.80**	63.72	96.21
HingMBERT	84.90	89.47	71.55	80.09	63.51	96.27
HingRoBERTa	**86.69**	**90.17**	**71.69**	81.13	66.43	96.15
HingMBERT-mixed	83.26	90.06	70.34	81.12	63.51	96.29
HingRoBERTa-mixed	86.13	89.87	70.73	80.68	**66.73**	95.96
HingBERT-LID	-	-	-	-	-	**98.77**

fields. Convolutional Neural Networks (CNNs) (14), Recurrent Neural Networks (RNNs)(9), and the most recent transformer-based design, Bidirectional Encoder Representations (BERT)(4), are some of these techniques.

As BERT-based designs gain prominence, studies on pre-training and fine-tuning them on diverse tasks have been conducted. Variations on the BERT architecture, such as RoBERTa(6) and ALBERT (Lan et al.), have aided in various use cases, such as accuracy and latency improvements. Models such as multilingual-BERT and XLM-RoBERTa(2) have mostly focused on multilingual and cross-lingual data representations.

When analysing code-mixed tasks, it was discovered that training on code-mixed sentences produced better results than training on various monolingual corpora (1). Bertlogicomix (11) shown that actual code-mixed data performs significantly better than synthetically created data after fine-tuning on several BERT-based architectures. All models mentioned above were pre-trained on at least 100k genuine code-mixed phrases.

A recent paper (2022) by R. Nayak et al (7) introduced L3Cube-HingCorpus, a first-of-its-kind large-scale code-mixed dataset for Hinglish, Hindi, and English tweets. It consists of nearly 52M tweets and 1.04B tokens. This dataset was used to train multilingual models. The paper introduces BERT-based pre-trained models like HingBERT, HingMBERT, HingRoBERTa, and also HingGPT, which have been trained on the L3Cube-HingCorpus dataset. They been tested on several downstream tasks like: Language Identification (LID), Part of Speech (POS) tagging, NER (Named Entity Recognition) and Sentiment analysis. Pretraining models on code-mixed L3Cube-HingCorpus achieve SOTA on majority of the tasks. HingRoBERTa gives the best overall performance and can be potentially used for sentiment analysis.

3 Dataset and Pre-processing

The dataset for the given subtask is balanced and consists of binary labels for hate detection namely HOF and NOT. The train set consists of 7593 labelled tweets text which is split into 90%-10% train-val split for our experiments. The test set comprises of 844 unlabelled tweets. The labels are

Splits	Samples Count
Train	6833
Val	760
Test	844

Table 1: Train/Val/Test split of the dataset

as follows:

1. **Hate and offensive(HOF):** Tweets that include hateful and offensive content
2. **Non-Hate-Offensive(NOT):** Tweets that doesn't include hateful and offensive content

We have used below text processing (10) on both train and test dataset for some experiments:

- Lower-casing
- Removal of HTML and URL tags
- Removal of usernames and extra white-spaces
- Removal of contraction, punctuation
- Lemmatize and removal of stopwords
- Tokenize

4 Methodology

We have used both ML and DL based classifiers in our approaches for binary classification task. The generic pipeline includes text pre-processing, tokenized embedding and a classifier. Below are the architectures for our approaches:

1. **ML based models with TF-IDF representations**
   The n gram TF-IDF representations are extracted after pre-processing the dataset. Now, these embeddings act as features for XGBoost classifier, which is trained to generate a prediction.

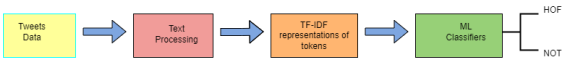

Figure 2: Architecture for ML based Model

2. **Bert Based models** Firstly, we generate word embeddings from a pre-trained BERT tokenizer. Then, we use a pre-trained model for sequence classification and fine-tune it on our dataset for hate detection.

Figure 3: Architecture for Bert based Model

We have experimented on below pre-trained models:

- **RoBERTa**
  Robustly Optimized BERT is a transformer based model with architecture similar to BERT. The modifications include removing Next Sentence Prediction (NSP) objective, training with bigger batch sizes, longer sequences and dynamically changing the masking pattern. It was trained on multiple dataset namely BOOK CORPUS and English Wikipedia dataset, CC-NEWS, OPENWEBTEXT and STORIES.

- **DistillBERT**
  The architecture is similar to BERT but with 6 encoder layers. It is only pre-trained for masked language prediction task and not next sentence prediction. It is trained using triplet loss function namely language model loss (similar to BERT), distillation and cosine-distance loss. The loss function displays a student-teacher learning relationship between Distilbert and BERT. Further, it is pre-trained on same dataset as BERT i.e. BookCorpus.

- **XLMRoberta**
  This is also a transformer based multilingual model trained on more than 100 languages. Thus, it makes it suitable for our task since our dataset is code-mixed (Hindi and English). It has shown great results in cross-lingual tasks.

- **HingRoBERTa**
  This model uses a pre-trained RoBERTa

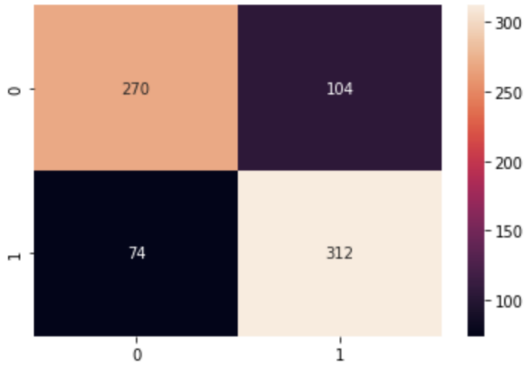

Figure 4: Confusion Matrix of Baseline XGBoost Classifier trained on tf-idf features on validation data.

and fine-tune it on L3Cube-HingCorpus using MLM objective with a masking probability of 15%. This is a Hindi-English code-mixed corpus, containing 52.93M sentences and 1.04B tokens.

- **HingRoBERTa-mixed**
  This model uses a pre-trained RoBERTa trained on both roman and Devanagari text which is mixed script and hence the name.

5 Experimental Results and Analysis

Model	LR	F1-score
Tf-idf XGBoost	0.5	74.2
RoBERTa	5e-7	77.5
DistilBERT	5e-7	76.1
HingRoBERTa	2e-5	80.0
HingRoBERTa-mixed	2e-6	77.6

Table 2: F1 score comparison table with Batch-size = 16 , Embedding Size = 128 ,Tf-idf Embedding Size = 6,833 and Optimizer = AdamW

1. We trained an XGBoost classifier on the tf-idf vectors. We got the best performance for n_estimators = 100, and a learning rate of 0.5. We got an F1-score of 74.2. We used this as a baseline for the subsequent models trained.

2. Thereafter, we tested a couple of monolingual BERT based models, namely - RoBERTa and DistilBERT. We used the obtained embeddings to train a subsequent sequence classifier. In each case we experimented with various embedding size, batch-size and learning rates.

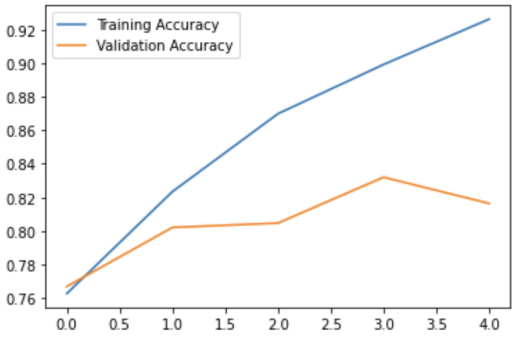

Figure 5: Accuracy vs Epochs plot for HingRoBERTa model

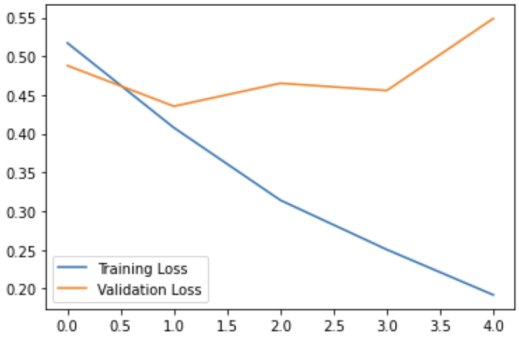

Figure 6: Loss vs Epochs plot for HingRoBERTa model with validation loss and F1 score as stopping criteria

We have used validation loss and F1 score as our criteria to select the best model.

We observed that increasing embedding size beyond 128 would not provide any major improvements to F1 score, but would significantly increase training time. Further, we observed that we got the best performance at a batch size of 16. Throughout our experiments we used AdamW optimizer, varying the learning rate to achieve the best performance for each model. We accordingly change the weight decay to 1/10th the learning rate.

The monolingual models, after hyperparameter tuning, showed an improvement of 2-3% over the baseline model.

3. Further, we tried out multilingual models. (8) showed that their models HingRoBERTa and HingRoBERTa-mixed, which were trained on Hinglish tweet corpus data, significantly outperformed other BERT based multilingual models such as XLM RoBERTa and multilingual-BERT, on downstream tasks on Hinglish data. We verified this on our own data; XLM-RoBERTa gave us similar performances as the monolingual models on our dataset. However, HingRoBERT gave us the highest F1 score of 80.0% on test data. Hing RoBERTa-mixed on the other hand, did not perform as well, with an F1 score of 77.6% on test data, even though it gave us the lowest loss, and the highest F1 score on the validation data.

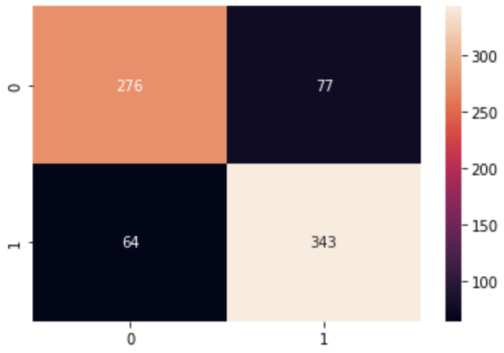

Figure 7: Confusion Matrix of HingRoBERTa on validation data

4. Finally, from our experiments it is clear that multi-lingual models(HingRoBERTa) perform much better than monolingual on code-mixed dataset. This is due to closeness of our dataset with that of corpus on which it was trained (Hinglish tweets). Its variant, HingRoBERTa-mixed, has a higher perplexity than HingRoBERTa and hence has comparitively poor performance.

6 Contribution

Ananya Dabas: DL based models training, finetuning, experimental results and analysis
Aaloke Mozumdar: DL based models training, finetuning, experimental results and analysis
Hitesh Bhandari: Introduction Related works, baseline model training
Akshita Gupta: Introduction Related works, baseline model training

References

[1] Ansari, M., Beg, M., Ahmad, T., Khan, M., and Wasim, G. (2021). Language identification of hindi-english tweets using code-mixed bert.

[2] Conneau, A., K. K. G. N. C. V. W. G. G. F. G. E. O. M. Z. L. S. V. (2020). Unsupervised cross-lingual representation learning at scale.

[3] Gupta, D., Ekbal, A., and Bhattacharyya (2020). *A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning.* Findings of the Association for Computational Linguistics, Washington, DC.

[4] J. Devlin, M.-W. Chang, K. L. K. T. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

[Lan et al.] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R.

[6] Liu, Y., O.-M. G. N. D. J. J. M. C. D. L. O. L. M. Z. L. S. V. R. (2019). A robustly optimized bert pretraining approach.

[7] Nayak, R. and Josh, R. (2022). L3cube-hingcorpus and hingbert: A code mixed hindi-english dataset and bert language models.

[8] Nayak, R. and Joshi, R. (2022). L3cube-hingcorpus and hingbert: A code mixed hindi-english dataset and bert language models. *arXiv preprint arXiv:2204.08398*.

[9] Pitsilis, G. K., Ramampiaro, H., and Langseth, H. (2018). Effective hate-speech detection in twitter data using recurrent neural networks, applied intelligence.

[10] Ponnurangam Kumaraguru, A. Kadama, A. G. J. J. J. S. K. M. S. M. R. P. K. T. A. M. S. (2021). Battling hateful content in indic languages hasoc '21. *HASOC*.

[11] S., S., A., S., and M., C. (2021). Bertologicomix: How does codemixing interact with multilingual bert? *AdaptNLP EACL 2021*.

[12] Srivastava, V., S. M. (2021). Hinge: A dataset for generation and evaluation of codemixed hinglish text. *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200—-208.

[13] Srivastava, V. and Singh, M. (2021). Challenges and considerations with code-mixed nlp for multilingual societies. 1.

[14] Z. Zhang, L. L. (2019). *Hate speech detection: A solved problem? the challenging case of long tail on twitter*. Semantic Web 10.