Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models

Satyajit Kamble
K J Somaiya College of Engineering
Mumbai, India
satyajit.k@somaiya.edu

Aditya Joshi
CSIRO Data61
Sydney, Australia
aditya.joshi@csiro.au

Abstract

This paper reports an increment to the state-of-the-art in hate speech detection for English-Hindi code-mixed tweets. We compare three typical deep learning models using domain-specific embeddings. On experimenting with a benchmark dataset of English-Hindi code-mixed tweets, we observe that using domain-specific embeddings results in an improved representation of target groups. We also show that our models result in an improvement of about 12% in F-score over a past work that used statistical classifiers.

1 Introduction

Hindi is one of the official languages of India[1], spoken by more than 551 million speakers[2]. As is typical of social media in any language, Hindi speakers on social media occasionally manifest hate towards one another. Hate speech refers to the use of hateful language, tone or prosody directed towards a person or a group of individuals, with the negative intention to provoke, intimidate, express contempt or cause harm to them. The membership to a group could be based on attributes such as race, religion, sexual orientation, ethnic origin, disability and so on.

Hate speech detection is the automated task of detecting if a piece of text contains hate speech. Hateful messages can be used to misinform people or result in violent incidents arising due to hate, therefore, hate speech detection assumes importance. In a recent news report, the Indian Government also expressed its intention to introduce a law to deal with online hate speech[3]. A tool for hate speech detection on social media in India is the need of the day.

As a country with high internet penetration and rich linguistic diversity, hate speech detection assumes an additional change in the case of Indian languages (Bali et al., 2014). Due to the difficulties in typing tools and familiarity with the English QWERTY keyboard, using a mixture of English words and transliterated Indian language words is common amongst the Indian internet users. Referred to as code-mixing or code-switching, the phenomenon corresponds to the use of transliterated words from one or more languages along with words in the language of the script. Challenges of creating and using code-mixed datasets are well-understood (Jamatia et al., 2016).

Towards this, we present an approach that uses deep learning for hate speech detection. We compare our approach with the past work by Bohra et al. (2018) and report a substantial improvement. The contribution of our work is:

1. We compare our deep learning-based approach with a statistical approach, and evaluate it on the same dataset as the statistical approach. We observe an improvement in the performance.

2. Instead of using pre-trained word embeddings, we train word embeddings on a large corpus of relevant code-mixed data. We demonstrate that this results in improved similarity values.

The rest of the paper is organised as follows. We describe related work in Section 2. The architecture is in Section 3 while the experiment setup is

[1] https://en.wikipedia.org/wiki/Hindi

[2] https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India

[3] https://www.thehindu.com/news/national/centre-moves-for-law-on-online-abuse/article23295440.ece

Proc. of ICON-2018, Patiala, India. December 2018 ©*2018 NLPAI*, pages 150–155

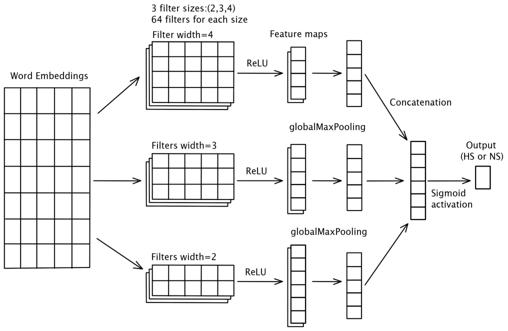

Figure 1: CNN model for hate speech detection.

in Section 4. We present our results in Section 5, and analyse the errors in Section 6.

2 Related Work

Approaches for hate speech detection have been reported (Schmidt and Wiegand, 2017; Warner and Hirschberg, 2012). Code-mixed datasets for Indian languages have been explored for several NLP tasks such as part-of-speech tagging (Jamatia et al., 2015), language identification (Das and Gambäck, 2014) and so on. Also, work concerning with hate speech in English language exists (Waseem and Hovy, 2016; Djuric et al., 2015; Davidson et al., 2017; Nobata et al., 2016). In a way, code-mixed datasets represent a majority of datasets from India, on the social media. Bohra et al. (2018) introduces a dataset of Hindi-English code-mixed tweets, and reports results on a statistical approach that use hand-engineered features. We download tweets from their dataset and compare with their results. Another work by Mathur et al. (2018) uses deep learning for hate speech detection. Our work differs from theirs in two ways: (a) We experiment with a different dataset, and compare performance on that dataset with the past work that reports results on the dataset, (b) We use domain-specific word embeddings that we show to be better indicative of semantics in the hate speech context. Our approach of using domain-specific embeddings is motivated by Tkachenko et al. (2018). They train two sets of word embeddings: one from a Wikipedia corpus and another from an Amazon review corpus. For sentiment-related tasks (such as sentiment classification), embeddings on the Amazon review corpus result in a higher performance as compared to those from the Wikipedia corpus. On the other hand, for topic-related tasks (such as topic classification), embeddings trained using the Wikipedia corpus outdo those from the Amazon review corpus.

3 Architecture

We propose three deep learning models for hate speech detection. These models are shown in Figures 1, 2 and 3 respectively. In the forthcoming sections, we describe each of the models.

3.1 CNN-1D

Figure 1 shows the CNN-1D model. It is fed in with domain-specific embeddings corresponding to sentences in the training data. The filters(3 filter sizes) with the specifications listed, convolve over the embeddings and produce the feature maps. Following this, we use a layer of globalMaxPooling having a dropout probability of 0.5. Then, the results are concatenated to form a single feature vector. Here, we apply the sigmoid activation to produce our final results.

3.2 LSTM

Figure 2 shows the LSTM model. Owing to the sequential nature of the code-mixed data, we make use of the LSTM model to compare our results. The results of the input embeddings, on passing through the LSTM layer, are made to accumulate at each proceeding timestep. The model is tuned to return the sequences of each of these timesteps. Next, the compiled sequences are given as an input to the globalMaxPooling layer. Lastly, the resulting output from the pooling layer is passed through the sigmoid activation function to give a final prediction.

3.3 BiLSTM

Figure 3 shows the BiLSTM model. Taking into consideration that the temporal dynamics can be better captured when a piece of text is analysed from both the directions, we make use of the BiLSTM to further compare our results. Here, instead of retrieving the sequences from a single direction, we do it for both the directions and concatenate the results. The vector now produced, goes through the globalMaxpooling layer. Finally, the result produced, is passed through the sigmoid activation to generate the final output.

3.4 Creation of Domain-Specific Word Embeddings

Using the Twitter API, we search for tweets containing Hindi cuss words and names of minority

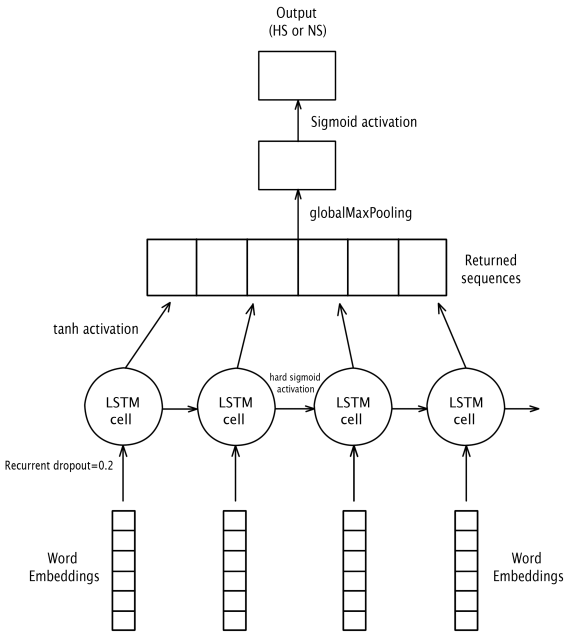

Figure 2: LSTM model for hate speech detection.

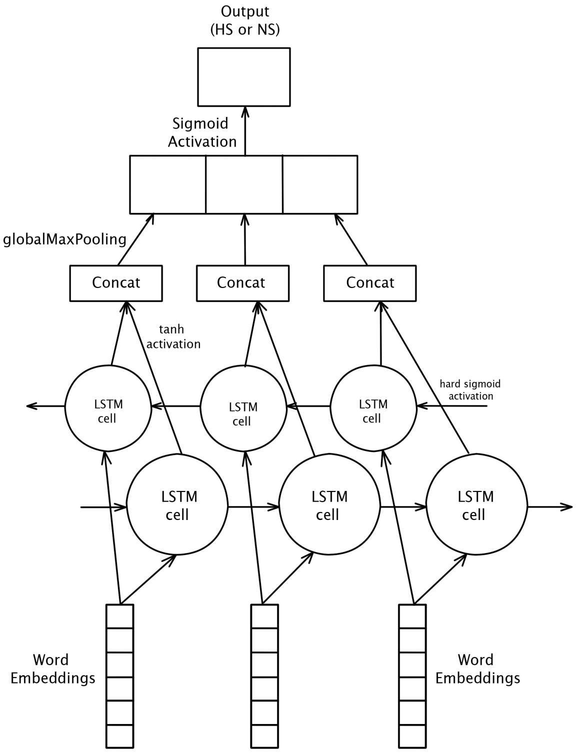

Figure 3: BiLSTM model for hate speech detection

Dataset Charecteristics	Size
Number of Tweets	255,309
Number of Timelines Extracted	7232
Number of Retweets	76,645
Total Number of Words	4,975,642
Size of Vocabulary	168,638
% Hindi Words per Tweet	18.63%

Table 1: Dataset Statistics of the Domain-Specific Word Embeddings

groups in their transliterated form. This is motivated by the definition of hate speech: hateful language that is used towards minority groups. We download a dataset of 255,309 tweets. Statistics of the dataset are in Table 1. Tweets collected were used only to train word embeddings. The dataset by Bohra et al. (2018) is used for evaluation of the 3 deep learning models.

We use the gensim (`https://radimrehurek.com/gensim/models/word2vec.html`) library to train word embeddings from this dataset, and use these domain-specific embeddings to initialize our deep learning models. We also utilize the Google Translate API to measure the average Hindi proportion of all the collected tweets. Using the API, we calculate the number of Hindi words in a tweet and calculate it's percentage with respect to the total number of words in the tweet. This is done for all tweets and an average is computed. *We commit to make our domain-specific word embeddings available for download at: `https://github.com/satyaSK/Hate-Speech-Detection`*.

4 Experiment Setup

We download the dataset by Bohra et al. (2018) using the Twitter API. Due to typical issues such as timeline restrictions, we obtain *3849* tweets, of which *1436* are labelled as hateful. We report 10-fold cross-validation performance on this dataset. We compare our models with a baseline re-implementation as given in Bohra et al. (2018). We implement feature extraction and use classification algorithms as described in their paper.

For the deep learning models, we use Keras, a neural network API (`https://keras.io/`). We experimentally determine the values of the parameters. For the CNN-1D model, we use the following hyperparameters:

1. Embedding dimension = 300
2. Number of filters of each filter size = 64, Batch size = 64, Epochs = 5, Dropout = 0.5
3. Pooling layer : Global max pooling
4. Filter sizes being 2,3 and 4 for the 3 CNNs in parallel.
5. Loss function : Binary cross-entropy loss
6. ReLU activation to obtain feature maps
7. Optimization algorithm : Adam

For LSTM and BiLSTM, we use the following configuration:

1. Number of LSTM units = 100, Recurrent dropout = 0.2
2. Loss function : Binary cross-entropy loss
3. Recurrent Activation : Hard sigmoid
4. Activation : tanh

We report Precision, Recall, F-score and accuracy values using methods in scikitlearn(Pedregosa et al., 2011).

5 Results

5.1 Qualitative Evaluation of domain-specific word embeddings

Table 2 shows cosine similarity between 'women' and words of three minority groups: religious, caste and sexual. We have not mentioned the specific names of the corresponding groups due to their controversial nature. *We wish to highlight that the word 'women' is used as a reference word solely because women might be a target of hate speech on social media.* Each row in the table is computed using the cosine similarity between the word 'women' and representative words of the specific minority group. The similarity between a pair of related social groups is consistently higher in the case of domain-specific embeddings as compared to general embeddings. For example, in case of sexual minority (which we consider as '*transgender*'), the similarity in the case of domain-specific embeddings is 0.726 while that in case of general embeddings is 0.348. This implies that domain-specific embeddings are able to capture the societal relationships and correlations between minority groups more accurately. An additional point to note is that, swear words in Hindi may not be present in pre-trained Google news embeddings. Specifically, we observe that 18 swear words in Hindi that were used to download the dataset, and were used to train domain-specific embeddings are not present in the Google news embeddings at all.

Minority Group	Domain-specific	General
Religious Minority	0.637	0.224
Caste Minority	0.615	0.204
Sexual Minority	0.726	0.348

Table 2: Cosine Similarity of 'women' with words representing three minority groups.

Therefore, higher similarity between groups that are targets of hate speech and higher coverage in terms of words that indicate expressions of hate, highlight the importance of using domain-specific embeddings.

5.2 Quantitative evaluation of hate speech detection

Bohra et al. (2018) train their classifiers using SVM and Random Forest algorithm, but only report accuracy. For a better comparison, we re-implement their features and obtain Precision, Recall and F-score values as well. The reported values and our values are compared with the deep learning models in Table 3. It must be noted that the accuracy values as reported and as obtained from re-implementation are close - indicating that the precision and recall are also likely to be comparable. We observe that using CNN-1D results in the highest performance with a F-score of 80.85% and an accuracy of 82.62%. This improvement in F-score is about 12% higher than the statistical baseline that we compare against. The improvement is in both precision and recall. An example of a correctly classified instance of hate speech by the CNN-1D model is '@.. *inke 6month ke works dekh lijiy nafrat ho jayegi aapko inse anandpal ke liye julus aur julus me public ko khule aam patthro ki barish karna dhamkana public ke sir fodna hate all of u*' which is translated to '@.. *look at the 6 month works of these people, you will start to hate them. A group of people rallying for Anandpal, has been stone-throwing and threatening the public. hate all of you.*'. Among the deep learning models, we observe that CNN-1D results in the highest precision while BiLSTM gives the best recall by a difference of approximately 0.40% as

	P (%)	R (%)	F (%)	A (%)
(Bohra et al., 2018) (SVM)	74.94	63.15	68.54	71.03 (71.7*)
(Bohra et al., 2018) (Random Forest)	62.43	58.88	60.60	65.78 (66.7*)
CNN-1D	**83.34**	78.51	**80.85**	**82.62**
LSTM	81.11	75.80	78.36	80.21
BiLSTM	82.04	**78.90**	80.43	81.48

Table 3: Comparison of Statistical Approach with Our Deep Learning-based Approach for Hate Speech Detection; * indicates reported values in the baseline paper; P: Precision, R: Recall, F: F-score, A: Accuracy.

compared to the CNN-1D. For example, this tweet '*@.. he is right x y may gundo ka palka kutha hai jo koi karwai nai kartha gundo par u.p no1 state in muders rape*' *(@.. he is right, x is a dog pet by the mafias of y, and so, he does not call for the investigation of the crimes they committed. u.p is number 1 state in rape and murders)* has been correctly classified as hate speech by the CNN-1D model while the LSTM and BiLSTM models incorrectly classify the tweet as non-hate speech.(x and y are anonymised names of a politician and a state respectively). In general, these results show that our deep learning models outperform the statistical approach.

6 Error Analysis

To understand the shortcomings of our models, we analyse and elucidate the errors made by our best-performing approach, which motivate future directions of experimentation. Some of these errors include:

- **Code-switched tweets in Hindi**: These are tweets written, following the grammatical structure of Hindi with a few English words. Many mis-classified examples include such tweets. An example is '@.. @.. @.. @.. *aur tum jahan hoti ho wahan balatkar badh jata hai baba bhi rape karne lagte hin (sic)*'. This tweet is translated as '@.. @.. @.. @.. *and rape cases start to increase wherever you go, baba also starts to rape'*. This has been identified to be a recurring error which occurs due to the code-mixed nature of the data at hand, where the text piece contains an imbalance between tokens from the Hindi and English scripts.
- **Series of swear words**: Some mis-classified instances are a string of swear words with a few function words between them. We skip an example here, on purpose, due to the obscene nature of these tweets. These errors may be because the model does not solely rely on the presence of swear words. Other context may be necessary to detect hate speech. This shows that the presence of explicit hate keywords or swear words is not the only determining factor for deciding whether a piece of text is hate speech or not, which points towards the necessity of capturing the underlying semantics and sense of the text in discussion.
- **Possibly incorrect labels**: Some tweets contain swear words but are not hateful towards any group as such. So, even though our models predict them as non-hate-speech, the instance is marked as mis-classified. For example, a hateful tweet calls someone the child of a rape victim but the gold label is negative. On the other hand, '*x ke samay me isase double rape hote the lekin us samay y bolti thi hai na (In times of x, the number of rapes were double as this, but y would always call it out, isn't it?!)* has the gold label as positive. (x and y are anonymised names of politicians).

7 Conclusion & Future Work

This paper explored hate speech detection in Hindi-English code-mixed tweets. We used three typical deep-learning models for detecting hate speech and empirically demonstrated their effectiveness. In contrast to statistical methods, our models were able to better capture the semantics of hate speech along with their context. We additionally demonstrated the efficacy of domain-specific word embeddings in adding intrinsic value to the code-mixed landscape.

Our work uses a benchmark dataset, and shows how deep learning models improve best-known work using statistical classifiers. In that, we make a small contribution to hate speech detection for Hindi-English code-mixed tweets. Novel deep learning techniques capable of assimilating textual cues more accurately, can be used to improve upon our work. Other nuances of hate speech in terms of sarcasm or misinformation can also be incorporated in future work.

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.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset of hindi-english code-mixed social media text for hate speech detection. In *Proceedings of the Second Workshop on Computational Modeling of Peoples Opinions, Personality, and Emotions in Social Media*, pages 36–41.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text.

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*.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30. ACM.

Anupam Jamatia, Björn Gambäck, and Amitava Das. 2015. Part-of-speech tagging for code-mixed english-hindi twitter and facebook chat messages. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 239–248.

Anupam Jamatia, Björn Gambäck, and Amitava Das. 2016. Collecting and annotating indian social media code-mixed corpora. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 406–417. Springer.

Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. 2018. Detecting offensive tweets in hindi-english code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 18–26.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

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.

Maksim Tkachenko, Chong Cher Chia, and Hady Lauw. 2018. Searching for the x-factor: Exploring corpus subjectivity for word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1212–1221.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.