Hindi-English Hate Speech Detection: Author Profiling, Debiasing, and Practical Perspectives

Shivang Chopra[1], **Ramit Sawhney**[2], **Puneet Mathur**[3], **Rajiv Ratn Shah**[4]
[1]Delhi Technological University, Delhi, [2]Netaji Subhas Institute of Technology, Delhi,
[3]University of Maryland College Park, [4]IIIT Delhi, Delhi
shivangchopra11@gmail.com, ramits.co@nsit.net.in, puneetm@cs.umd.edu, rajivratn@iiitd.ac.in

Abstract

Code-switching in linguistically diverse, low resource languages is often semantically complex and lacks sophisticated methodologies that can be applied to real-world data for precisely detecting hate speech. In an attempt to bridge this gap, we introduce a three-tier pipeline that employs profanity modeling, deep graph embeddings, and author profiling to retrieve instances of hate speech in Hindi-English code-switched language (Hinglish) on social media platforms like Twitter. Through extensive comparison against several baselines on two real-world datasets, we demonstrate how targeted hate embeddings combined with social network-based features outperform state of the art, both quantitatively and qualitatively. Additionally, we present an expert-in-the-loop algorithm for bias elimination in the proposed model pipeline and study the prevalence and performance impact of the debiasing. Finally, we discuss the computational, practical, ethical, and reproducibility aspects of the deployment of our pipeline across the Web.

Introduction

Context and Original Scope

"Social media has given people a platform to spew hate speech and propagate radical beliefs in an attempt to amplify fringe opinions" (Singh 2019). One of Twitter's most pressing challenge remains to deal with abusive behavior and hate speech (Tiku and Newton 2015). Hate speech is an act of offending, insulting, or threatening a person or a group of people on the basis of caste, religion, sexual orientation, or gender (Schmidt and Wiegand 2017). This hate speech thereby forms a big portion of content that is harmful and degrading to the mental health of users on social media in the long run. The widespread access of social media websites to individuals from linguistically distinct regions and cultures has led to a blend of natively spoken languages with English, popularly known as code-switched languages (Silva et al. 2016). Hinglish, a portmanteau of Hindi and English, is the macaronic hybrid use of English and South Asian languages from across the Indian subcontinent, involving code-switching between these languages whereby native languages are written in Roman script. Over the years, the interest in detecting and removing hate speech from content preset on online forums like Twitter and Facebook in an automated way has risen significantly. The lack of generalizability of off-the-shelf hate speech moderation systems for code-switched languages necessitates efficient techniques that can detect offensive content automatically on the Internet.

Motivation

The inability of mono-lingual hate-speech classifiers to detect the semantic cues in code-switched languages necessitates an efficient classifier that can detect offensive content automatically from code-switched languages. There exists an active field of research in the automatic detection of code-switched hate-speech on social media. It has been observed that a significant part of the hate-inducing content on social media comes from youth who participate in communities and are susceptible to the influence and fringe opinions of individuals of those communities. The active involvement in communities leads to the presence of a strong label as well as linguistic homophily among users in the same community (Mishra et al. 2019a). This approach has been successfully applied to hate-speech detection in English (Mishra et al. 2019a), and the general nature can be extended to code-switched languages where the societies should in-theory be even tightly coupled. Fig. 1 gives a semantic of the motivation of our approach. A very clear co-relation can be observed in the tweets posted by users in tightly coupled communities. The two concepts of label and linguistic homophily, when used leveraged in concurrency with our bias elimination pipeline, lead to a balance between the generalizability, specificity, and fairness of our model.

Challenges

Analysis of the baseline implementations shown in Table 3 reveal the prevalence of more significant bias against specific communities in Hinglish content as compared to mono-lingual content on social media. Additionally, being a low resource language with loosely defined semantic and grammatical rules, context identification becomes a problematic task for Hinglish (Mathur et al. 2018b). Furthermore, the pre-existing bias-elimination techniques for mono-lingual

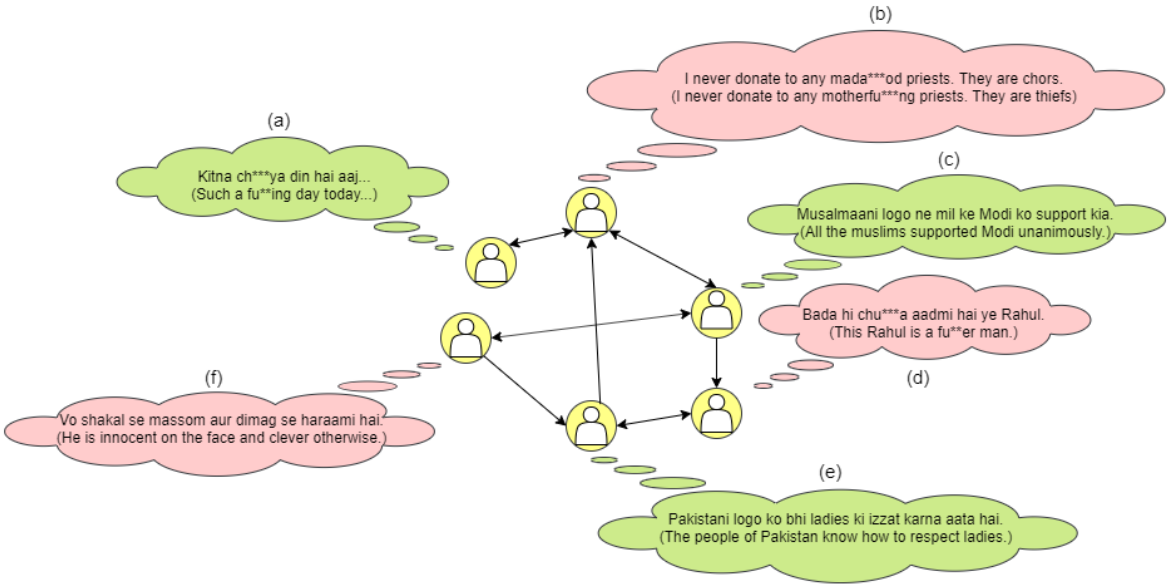


Figure 1: Hinglish tweets with varied bias and linguistic homophily (mappings taken from Profane Word list given by (Mathur et al. 2018b))

content are not directly applicable to code-switched data. The examples shown in Fig. 1 demonstrate the linguistic and label homophily present in the social media data along with a few cases where the elimination of bias becomes imperative. The green and red color of the tweets in Fig. 1 represent the ground truth labels of the tweets as non-hate speech and hate speech, respectively. The tweet shown in Fig. 1 (a) is not a hate-inducing tweet as it expresses a person's general opinion and not some hate speech against any particular section of the society. However, due to the presence of a profane word, it was labeled as hate speech by the baseline models. Furthermore, the tweets in Fig. 1 (c) and (e) are not instances of hate speech but were labeled as one due to the reference to a particular religion or community. This lack of proper semantic definitions for Hinglish constructs and the stereotypical bias present in the annotations makes the process of hate speech detection and bias elimination a herculean task.

Contributions

We develop a novel Hinglish hate speech detection pipeline that builds upon contextual cues and linguistic fundamentals as:

- **Profanity Vectors & Linguistic Modeling:** We leverage deep learning architectures based on CNN, LSTM, Attention layers for hate speech identification. Further, inspired by (Mathur et al. 2018b), we augment the linguistic information with a profanity vector to improve the performance of our model.
- **Graph Embeddings & Linguistic Homophily:** We construct a social network based graphs for the present dataset, which includes the social interaction cues of all the users in the dataset. Further, we use this graph to exploit the label and linguistic homophily present in the Hinglish-speaking communities to enhance the classification accuracy of our models.
- **Bias Elimination in Hinglish Code Mixed Speech:** Inspired by (Bolukbasi et al. 2016), we propose an algorithm for bias mitigation and demonstrate its effectiveness through extensive qualitative and quantitative analysis.

Through extensive qualitative and quantitative analyses of several traditional and state-of-the-art baselines along with an in-depth bias analysis done in Section 5, we demonstrate the effectiveness of both, the individual components and overall proposed pipeline, and pave the way for future work by highlighting the limitations, open challenges and social aspects of hate speech in Hinglish.

The remainder of the paper is organized as follows. In Section 2, we discuss related work in the areas of code-switched linguistics, community-based author profiling, and bias mitigation. In Section 3, we discuss and formalize the problem definition and present our methodology. We present experimental results and analysis in Sections 4, 5, respectively. We then briefly discuss the ethical considerations and limitations of our work in its present form. Finally, we conclude with a brief summary in Section 6.

Related Work

Sentiment Analysis in Code-Switched Languages

Our work builds extensively on the previous work done on handling the linguistic aspect of code-switching so as to preserve the syntactic and semantic peculiarities of the language constructs. (Lee and Wang 2015) used a multiple-classifier based automatic detection approach to perform sentiment analysis of Chinese-English code-switched data. (Ray 2015) took into account grammatical transitions to

perform sentiment analysis of Hindi-English code switched data. The approach of fine-tuning word embeddings onto the code-switched data, although tested on a Hindi-English code mixed dataset, is generic enough to be extended onto other code-mixed language sentiment analysis tasks.

Linguistic Hate Speech Detection

Having plagued online communities for years, technical, design, and moderation approaches have been designed to cope with abusive posts. Works like (Bohra et al. 2018) and (Ravi and Vadlamani 2016) used hand-crafted features and statistical machine learning methods to perform hate speech classification. However, contemporary advances in the field of Deep Learning-based hate speech analysis have bettered these approaches. A CNN-based transfer learning approach was used by (Mathur et al. 2018b) to detect offensive tweets. Furthermore, they introduced the HEOT dataset as well as the Profanity Lexicon Set, which are subsequently used in our experiments. (Kapoor et al. 2018) used LSTM based transfer learning on the HEOT dataset. (Santosh and Aravind 2019) further employed a hierarchical LSTM model with attention based on phonemic sub-words. Also, (Mathur et al. 2018a) used a Multi-Input Multi-Channel Transfer Learning Architecture to classify hate speech in Hinglish. All these approaches focused only on linguistics and did not include any community profiling information.

Community based Author Profiling

Representations of users in a social network have been utilized for author profiling because people connected on social media are likely to post similarly. Works (Chen and Ku 2016), and (Yang, Chang, and Eisenstein 2016) used representations derived from a social graph to perform specific tasks like entity linking and stance classification. (Mishra et al. 2019b) used Graph-based embeddings to perform Suicidal Ideation Detection in tweets, which motivated us to apply the same to our use case. Our work derives inspiration from (Mishra et al. 2019a) who incorporated social media-based graph embeddings on performing Hate Speech Detection in English and achieved better results than the state-of-the-art at that time. Being densely packed due to the inherent homophily in Hinglish speaking communities, our graph embeddings encapsulate more important structural community information and improve author profiling over less tightly coupled communities.

Bias Mitigation in Linguistic Deep Learning Models

(Brunet et al. 2018) demonstrated the fact that word embeddings have been shown to contain bias along multiple spectrums like gender, religion, race, ethnicity that is inherited from their training corpora. (Swinger et al. 2018) proposed an Unsupervised Bias Enumeration (UBE) algorithm, which is used to enumerate sets of bias-inducing words in an unsupervised manner. However, their existing work concentrated entirely on quantifying and alleviating such bias in English, and the analysis cannot be directly applied to code-switched languages, such as Hinglish.

Methodology

Problem Formulation

Let $\tau = \{t_1, t_2, ..., t_n\}$ be a set of n tweets and $Y = \{y_1, y_2, ..., y_n\}$ be the corresponding set of n labels where $y \in \{0, 1\}$, indicates the absence and presence of hate speech, respectively. The objective of our model is to predict the conditional label distribution $P(y|t)$. To enhance the performance of the model, we incorporate the user based information from the social graph and also analyze the impact of network embeddings and bias-elimination on the performance of our model. The overall architecture of our model can be seen in Fig. 2.

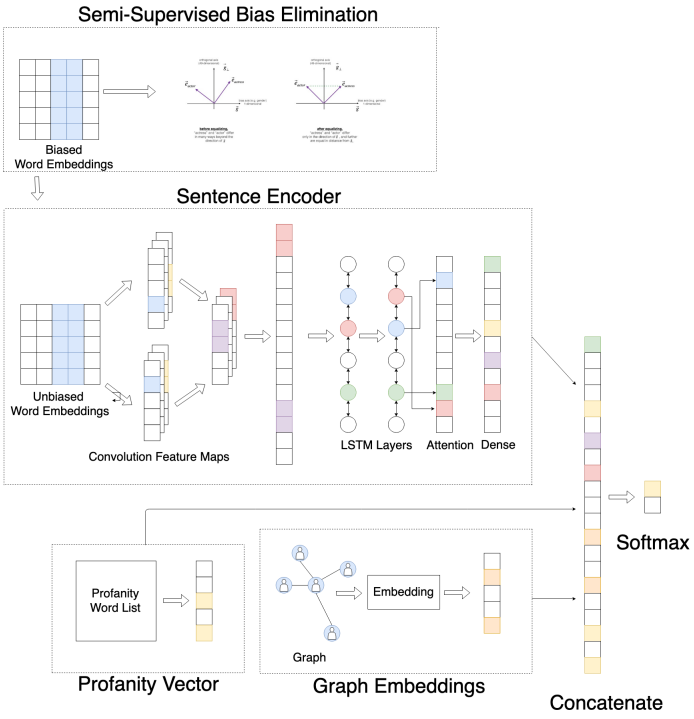


Figure 2: Overall pipeline

Data and Preprocessing

To validate the proposed hypothesis, we use two datasets, HS (Bohra et al. 2018) and HEOT (Mathur et al. 2018b). The HS dataset had 2195 non-offensive and 1280 offensive tweets, whereas, the HEOT dataset had 1121 non-offensive, 303 abusive, and 1765 offensive tweets. Furthermore, the HS dataset had tweets from 3005 unique users whose social media interactions were captured in the form of graphs. The corresponding data was unavailable for the HEOT dataset. The class-wise distribution of users in the HS and HEOT dataset can be seen in Fig. 3.

The preprocessing for the tweets involved the following steps:

- **Tokenization and Lemmatization**
  A tweet-tokenizer was used to parse the tweet and replace every username mentions, hashtags, and URLs with **mention**, **hashtag** and **url** respectively. The tokenized text

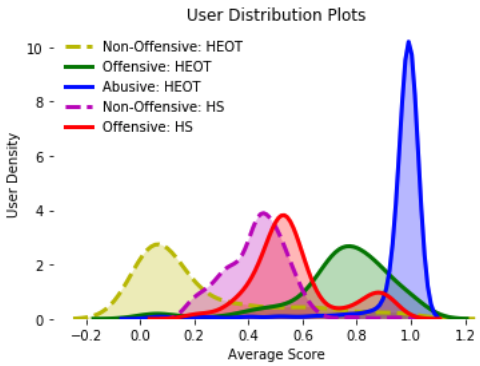


Figure 3: Class-wise distributions of the user density

then underwent stopword removal and was used as an input to WordNet Lemmatizer provided by NLTK. The lemmatization was done to transform the words to their root form so as to map them to their corresponding Graph embeddings.

- **Keras Tokenization and Padding**
  The lemmatized tokens were recombined, and these sentences were passed to the Keras Tokenizer, which mapped the words to unique integers and padded the sentences to a maximum length of 120 (Value obtained by finding the maximum length of a tweet in the dataset).

Baseline Classifiers

Transliteration-based Preprocessing: The initial baselines were using a seq2seq model to handle the Hinglish data. The preprocessing was a five-step process before the preprocessing steps defined in the previous subsection:

- Two separate sets of Hindi and English words were formed using the NLTK English dictionary and the IIT Bombay Hindi dictionary (Kunchukuttan, Mehta, and Bhattacharyya 2017). Also, an English-Hindi word mapping dictionary was formed for the words in the English set using Google translate.
- Each lemmatised tweet t_i was further broken into tokens $v_i \in V$. Each token v_i was passed through a sequence matcher to check if it is present in the English dictionary. If a match is found, the word is translated into its Hindi counterpart using the dictionary.
- If a match is not found, the token e_i is transliterated into Hindi, and the sequence matcher is used to check for the presence of the word in the Hindi word set.
- Any words left unmatched are transliterated directly to Hindi. This is valid because, as observed, the base language of code-mixing for the majority of the tweets was found to be Hindi.
- The Hindi tweet was then passed through a seq2seq Neural Machine Translation model trained on the IIT Bombay Parallel Dataset (Kunchukuttan, Mehta, and Bhattacharyya 2017). The output of the seq2seq model was then passed thought the two preprocessing steps, and the final tweet was passed as input to the models.

Fine-tuning based Preprocessing: Further, a second set of experiments were performed wherein a different kind of data preprocessing post the steps described in the previous subsection was used. Instead of transliterating and translating the code-mixed tweets into English, we fine-tuned the word2vec embeddings onto our Hinglish dataset. This yielded a better performing model for our experiments at the cost stereotypical biases being perpetuated by our models, which were later handled.

In order to determine a baseline architecture to be used as the backbone for our further experiments, we performed several experiments on CNN and LSTM based models. The post-preprocessed Hindi tweets were passed through each of the models, and the results are as summarised in the Text Comparison section of Table 3.

As observed, the best performance was given by the CNN + Bidirectional LSTM + Attention, and therefore, it was used as the backbone model for the rest of the experiments.

Profanity Vector Augmentation

Further, inspired by (Mathur et al. 2018a), we propose a Profanity Vector (**PV**) to be concatenated in the model for performance improvement. The profanity word list given by (Mathur et al. 2018a) is used to construct $\mathbf{PV}(210D)$ for each tweet $t_i \in \tau$ such that a corresponding 1 demarcates the presence of a particularly bad word while its absence is demarcated by 0 to emphasize the absence of a contextually subjective swear word. This profanity vector is then concatenated with the encoded sentence.

$$\mathbf{PV}^{(j)} = \begin{cases} 0 & \text{if } p_j \in t_i \\ 1 & \text{if } p_j \notin t_i \end{cases} \quad (1)$$

Graph based Author Profiling Model

The engagement between hate-inducing users was captured in the form of social network graphs. We define two social relationship graphs for each user - follower graph and retweet graph. Let $U = \{u_1, u_2, ..., u_m\}$ be the set of m users who authored the tweets in τ. Let $G = (V, E)$ be the social graph of the users where, $V = V_1, V_2, ...V_n$ is the set of nodes and bears one-to-one mapping with set U. $E = \{e_1, e_2, ..., e_z\}$ is the set of z edges where $e_i = (v_x, v_y)$ represents an undirected edge between nodes $v_x, v_y \in V$ such that either V_x follows V_y or V_x retweets a tweet posted by V_y. Table shows the statistical analysis of $G(V, E)$.

Figure 4: Follower and Retweet Graph

Statistic	Value
Number of nodes	3005
Number of edges	4448
Average degree	2.9604
Maximum path length	11
Largest Connected component	1481

Table 1: Graph statistics for HS dataset

Model	F1	Accuracy
node2vec + Dense	0.50	0.52
DeepWalk + Dense	0.52	0.63
CNN + BiLSTM + Attn + n2v	0.63	0.64
CNN + BiLSTM + Attn + DeepWalk	0.67	0.71

Table 2: Graph results with Accuracy and F1 score for the transliteration-based experiments.

To obtain the author profiles, nodes in the graphs were converted into feature representations using DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) and node2vec (Grover and Leskovec 2016).

Node2vec works on the lines of word2vec and determines the context of the nodes by looking into their neighborhoods in the graph. It constructs a fixed number of random walks of constant length for each of the nodes to define the neighborhood of the nodes. The random walks are governed by the parameters p (return parameter) and q (inout parameter), which have the ability to fluctuate the sampling between a depth-first strategy and a breadth-first strategy.

DeepWalk uses local information obtained from truncated random walks to learn latent representations by treating walks as an equivalent of sentences. For each vertex v_i in the graph, we perform γ random walks of length l and the probability of embeddings Φ of the vertex given the vertices v_j on the random path is maximized.

$$J(\Phi) = -log(P(v_i|\Phi(v_j)) \tag{2}$$

To enhance the performance of the model, we incorporate the author-profile information from the social graph into our model, as shown in Fig. 2. A summary of the experiments with node2vec and DeepWalk is summarised along with the resuults in Table 2.

Bias Elimination

The prevalence of bias along social, religious, and gender spectrums in models for Hinglish hate speech classification was observed in the labeled dataset. Inspired by (Swinger et al. 2018), we propose a Bias Elimination (BE) algorithm to mitigate the effects of such bias and describe it below.

- **Clustering Words**: Make k disjoint clusters of words $C_1, C_2, C_3, ... \in C$, and their corresponding centroids $c_1, c_2, c_3, ... \in c$ using k-Means Algorithm.
- **Two Centroid Sub-Clustering**: For each cluster, a hyperparameter λ is used to find the set of words closest to the centroid to be de-biased using a two centroid sub-clustering algorithm. The algorithm begins with two sets of words one has the first word, and the other has the remaining words in the sub-cluster. Then an iterative process begins wherein we keep on adding the word in second set nearest to the centroid of the first set and recomputing the centroid of both the sets. This process is repeated until the distance between the word and centroid becomes greater than λ. Post that, according to our hypothesis, the words get too far to be clustered together.
- **Making Pairs**: All possible combinations of words were used to make ${}^nC_2$ pairs for each of the n clusters in the above set are made.
- **Expert Segregation**: After the pairs are extracted as above, three university students, proficient in Hinglish and adept at Twitter, were provided with the guidelines to segregate the bias-inducing from the non-bias-inducing pairs. The guidelines were based on the following classification system:
  - **Bias-inducing**: The pair links a particular cast, gender, ethinicity, or religion to a positive or negative trait like "peace" and "terrorism," respectively. In addition to that, any word pairs linking particular words with a word from the profane word list.
  - **Non-Bias Inducing**: Any pair portraying logical connection between two words like "woman" and "pregnancy."

  An acceptable Cohen's Kappa score of 0.76 was found between the two annotations. In cases of ambiguity in labeling or conflicts in merging, the default class 0 (non-bias inducing) was assigned. We commit to releasing our annotated pairs list to the community.
- **Finding Bias Axis**: For each pair, a neutral $axis_{bias}$ is calculated using the word embeddings e_i and e_j of the two words as shown in equation (3).

$$axis_{bias} = e_i - e_j \tag{3}$$

- **Neutralising**: Once the bias-inducing pairs are found, the words w_i and w_j in each pair are neutralised to ensure that they are neutral to $axis_{bias}$ in their respective sub-space as shown in the following equations (4, 5, 6, 7).

$$\mu = (e_i + e_j)/2 \tag{4}$$

$$\mu_B = \frac{\mu.axis_{bias}}{||axis_{bias}||^2} * axis_{bias} \tag{5}$$

$$\mu_{orth} = \mu - \mu_B \tag{6}$$

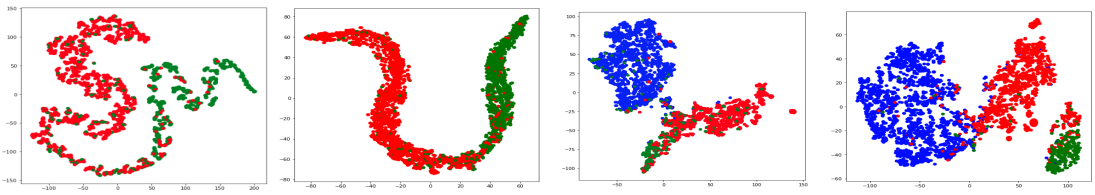
$$e_{w_B} = \frac{e.axis_{bias}}{||axis_{bias}||^2} * axis_{bias} \tag{7}$$

- **Equalizing**: Then we equalize the pair of words outside the subspace and thereby enforce the property that the neutral words are equidistant from the $axis_{bias}$.

$$e_{corrected} = \sqrt{|1 - ||\mu_{orth}||^2|} * \frac{(e_{wB} - \mu_B)}{|(e_w - \mu_{orth}) - \mu_B|} \tag{8}$$

	Model	F1(HS)	Acc(HS)	F1(HOT)	Acc(HOT)
Text Comparison	seq2seq + CNN	0.49	0.51	0.70	0.76
	seq2seq + CNN + LSTM	0.60	0.60	0.69	0.70
	seq2seq + CNN + BiLSTM	0.54	0.56	0.72	0.76
	seq2seq + BiLSTM	0.58	0.59	0.61	0.63
	seq2seq + BiLSTM + Attn	0.62	0.62	0.71	0.77
	seq2seq + CNN + BiLSTM + Attn	0.62	0.62	0.72	0.76
Graph Ablation	seq2seq + CNN + node2vec	0.50	0.52	-	-
	seq2seq + CNN + LSTM + n2v	0.61	0.61	-	-
	seq2seq + CNN + BiLSTM + n2v	0.57	0.57	-	-
	seq2seq + BiLSTM + n2v	0.59	0.59	-	-
	seq2seq + BiLSTM + Attn + n2v	0.62	0.63	-	-
	seq2seq + CNN + BiLSTM + Attn + n2v	0.63	0.64	-	-
	seq2seq + CNN + BiLSTM + Attn + DW	0.67	0.71	-	-
	seq2seq + PV + n2v	0.52	0.63	-	-
Debiasing Ablation	seq2seq + CNN + BiLSTM + Attn + PL	0.64	0.71	0.77	0.85
	seq2seq + CNN + BiLSTM + Attn + PV + DW	0.73	0.78	-	-
	FT + CNN + BiLSTM + Attn + PV	0.57	0.61	0.70	0.82
	FT + CNN + BiLSTM + Attn + PV + DW	0.64	0.69	-	-
	FT + CNN + BiLSTM + Attn + PV + Debias	0.64	0.71	**0.77**$^{+}$	**0.85**$^{+}$
	FT + CNN + BiLSTM + Attn + PV + DW + Debias	**0.73**$^{+}$	**0.78**$^{+}$	-	-
Comaparative	(Kapoor et al. 2018)	0.71*	0.74*	0.73*	0.87
	(Mathur et al. 2018b)	0.69*	0.72	0.71*	0.83*
	(Bohra et al. 2018)	0.62*	0.71	0.70*	0.76*
	(Santosh and Aravind 2019)	0.48	0.71	0.52*	0.63*
	Our Model	0.73	0.78	0.77	0.85

Table 3: Final Results with Accuracy and F1 score. - : No results due to unavailability of data * : Replication of baselines $^{+}$: Statistically significant results



(a) HS + GE (b) HS + GE + BE (c) HEOT (d) HEOT + BE

Figure 5: t-SNE Plots with graph embeddings (GE) and bias elimination (BE) for the HS (Bohra et al. 2018) and HEOT (Mathur et al. 2018b) datasets.

Experiments and Results

Experimental Setup

All the experiments were conducted with a train-test split of 80:20. The experiments were conducted in phases with features being added and the results noted sequentially. Initial models used pre-trained Word2Vec embeddings trained on Twitter data along with a seq2seq model. Later experiments were performed using Word2Vec Embeddings fine-tuned on our datasets. The models were trained for 20 epochs with Early Stopping used with a patience of 0.05 on the validation accuracy. We used the HS (Bohra et al. 2018) and HEOT (Mathur et al. 2018b) datasets throughout our experiments.

Results

An iterative process was followed while experimenting. The models used and the results are as follows:

Baseline Claasifiers: A number of experiments were performed using a combination of Convolution Layers, LSTM Layers, and Attention layers to select the most appropriate framework to be used as a backbone network. The results for all the models are summarised in the Text Comparison section of Table 3. As observed, a combination of CNN, Bidirectional LSTM, and Attention yielded the best results. The model ensured that spatial, temporal, and attention-based information was being captured by the Convolution, LSTM, and Attention-based layers respectively and therefore was able to outperform the other models.

Incorporating Graph Embeddings: The node2vec and DeepWalk based graph embeddings were used to capture any communities that were being formed on social media. Due to the inherent homophily present in the Hinglish speaking community and their social proximity resulted in an in-

Figure 6: Qualitative analysis of model pre debiasing and post debiasing

crease in the model's performance. The graph embeddings concatenated with the encoded sentence resulted in an increase of 7% in accuracy. The results after incorporating graph-based features are summarised in the Graph Ablation section of Table 3. The t-SNE plots for the various models with the graph data incorporated can be seen in Fig. 5. It is clear from the plots that the offenders form close-knit communities, which the model is able to capture and thereby improve classification performance.

Profanity Vector Incorporation: As pointed out by (Mathur et al. 2018b), the offensive tweets in Hinglish are often accompanied by the used of profane words specific to Hindi. Incorporating this information into our model yielded another significant improvement over the baselines.

Debiasing Study: A qualitative analysis of the performance of our model revealed striking biases being captured by it. Strong biases against particular religions like "Islam" and particular communities like the LGBT community were being propagated by our models. The Bias Elmination Algorithm used was able to eliminate such biases and was thereby able to correctly classify sentences which were earlier being misclassified by our model. This led to a more bias-neutral model with better classification accuracy. Some of the tweets being misclassified and the results pre and post debiasing are summarized in Fig. 6. The t-SNE plots shown in Fig. 5 demonstrate the improvement in the classification ability of the models post debiasing. A more comprehensible and separable t-SNE plot is observed in both cases prior to the application of debiasing algorithm. Also, Fig. 6 shows an in-depth qualitative analysis of the model pre and post debiasing. It can be clearly seen that the effect of religious, gender, and social bias on the classification of tweets has reduced drastically post debiasing.

Ethical Considerations and Limitations

Amid the controversy surrounding the freedom of expression, defining (online) hateful speech remains a complex subject of ethical, legal, and administrative interest. The preponderance of the work presented in our discussion can present heightened ethical challenges. We address the following limitations:

- **Confidentiality**: Individual consent from users was not sought as the data was publicly available. Therefore, we must address the trade-off between privacy and effectiveness. Access to the data is imperative for making our models effective. However, we must work with the purview of acceptable privacy practices to avoid social stereotyping that might lead to adverse conflicts. We, therefore, actively make efforts to hide identity, revealing information to ensure user anonymity at all points.
- **Prejudice**: Our work is not intended to be used to intentionally or inadvertently marginalize or influence prejudice against those groups who are already marginalized (by gender, race, religion, sexual orientation, etc.), or vulnerable, and are often the victims of hateful speech.
- **Potential Misrepresentation**: Although our work attempts to analyze aspects of users' nuanced and intricate experiences, we acknowledge the limitations and potential misrepresentations that can occur when researchers analyze social media data, especially data from an offensive population or group to which the researchers do not explicitly belong.
- **Obstacles to Deployment**: The language-specific nature compounded with the semi-manual approach impedes the generic employment of our bias elimination algorithm onto other code-switched languages. This restricts the scope of our algorithm in its current state.

Conclusion and Future Work

In this work, we proposed to enhance the hate speech detection of code mixed languages by incorporating social media-based features into our models, along with capturing the use of profane words. The qualitative analysis of our models on two real-world datasets revealed astonishing gender, social, and religious biases being induced into our models. A novel bias elimination algorithm was proposed to mitigate any present bias from the model, thereby rendering a fair classification architecture. We noted a statistically significant improvement of 0.04 and 0.07 in F1 score over the state-of-the-art in the HS and HEOT datasets. Our future agenda includes exploring the applicability of our methods and algorithms onto other code-switched languages.

References

Bohra, A.; Vijay, D.; Singh, V.; Akhtar, S. S.; and Shrivastava, M. 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 People's Opinions, Personality, and Emotions in Social Media*, 36–41. New Orleans, Louisiana, USA: Association for Computational Linguistics.

Bolukbasi, T.; Chang, K.; Zou, J. Y.; Saligrama, V.; and Kalai, A. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *CoRR* abs/1607.06520.

Brunet, M.; Alkalay-Houlihan, C.; Anderson, A.; and Zemel, R. S. 2018. Understanding the origins of bias in word embeddings. *CoRR* abs/1810.03611.

Chen, W.-F., and Ku, L.-W. 2016. UTCNN: a deep learning model of stance classification on social media text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 1635–1645. Osaka, Japan: The COLING 2016 Organizing Committee.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. *CoRR* abs/1607.00653.

Kapoor, R.; Kumar, Y.; Rajput, K.; Shah, R. R.; Kumaraguru, P.; and Zimmermann, R. 2018. Mind your language: Abuse and offense detection for code-switched languages. In *AAAI*, volume abs/1809.08652.

Kunchukuttan, A.; Mehta, P.; and Bhattacharyya, P. 2017. The IIT bombay english-hindi parallel corpus. *CoRR* abs/1710.02855.

Lee, S., and Wang, Z. 2015. Emotion in code-switching texts: Corpus construction and analysis. 91–99.

Mathur, P.; Sawhney, R.; Ayyar, M.; and Shah, R. 2018a. Did you offend me? classification of offensive tweets in Hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 138–148. Brussels, Belgium: Association for Computational Linguistics.

Mathur, P.; Shah, R.; Sawhney, R.; and Mahata, D. 2018b. Detecting offensive tweets in Hindi-English code-switched language. In *Proceedings of the International Workshop on Natural Language Processing for Social Media*, 18–26. Melbourne, Australia: Association for Computational Linguistics.

Mishra, P.; Tredici, M. D.; Yannakoudakis, H.; and Shutova, E. 2019a. Author profiling for hate speech detection. *CoRR* abs/1902.06734.

Mishra, R.; Prakhar Sinha, P.; Sawhney, R.; Mahata, D.; Mathur, P.; and Ratn Shah, R. 2019b. SNAP-BATNET: Cascading author profiling and social network graphs for suicide ideation detection on social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, 147–156. Minneapolis, Minnesota: Association for Computational Linguistics.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. *CoRR* abs/1403.6652.

Ravi, K., and Vadlamani, R. 2016. Sentiment classification of hinglish text. 641–645.

Ray, D. 2015. Sentiment analysis of mixed language employing hindi-english code switching.

Santosh, T. Y., and Aravind, K. V. 2019. Hate speech detection in hindi-english code-mixed social media text. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, CoDS-COMAD '19, 310–313. New York, NY, USA: ACM.

Schmidt, A., and Wiegand, M. 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*, 1–10. Valencia, Spain: Association for Computational Linguistics.

Silva, L. A.; Mondal, M.; Correa, D.; Benevenuto, F.; and Weber, I. 2016. Analyzing the targets of hate in online social media. *CoRR* abs/1603.07709.

Singh, Y. 2019. Need to crackdown on hate speech on social media: India calls for action at un.

Swinger, N.; De-Arteaga, M.; IV, N. T. H.; Leiserson, M. D. M.; and Kalai, A. T. 2018. What are the biases in my word embedding? *CoRR* abs/1812.08769.

Tiku, N., and Newton, C. 2015. Twitter ceo: 'we suck at dealing with abuse'.

Yang, Y.; Chang, M.; and Eisenstein, J. 2016. Toward socially-infused information extraction: Embedding authors, mentions, and entities. *CoRR* abs/1609.08084.