Hate Speech Detection in Hindi-English Code-Mixed Social Media Text

T.Y.S.S. Santosh
IIT Kharagpur
Kharagpur, West Bengal, India
santoshtyss@gmail.com

K.V.S. Aravind
IIT Kharagpur
Kharagpur, West Bengal, India
kollipara.aravind@gmail.com

ABSTRACT

With the increase in user generated content, particularly on social media networks, the amount of hate speech is also steadily increasing. So, there is a need to automatically detect such hateful content and curb the wrongful activities. While relevant research has been done independently on code-mixed social media texts and hate speech detection, this paper deals with the task of identification of hate speech from code-mixed social media text. We perform experiments with available code-mixed dataset for hate speech detection using two architectures namely sub-word level LSTM model and Hierarchical LSTM model with attention based on phonemic sub-words.

CCS CONCEPTS

• **Human-centered computing** → *Social tagging*; *Social media*;

KEYWORDS

code-mixing,hate speech,deep learning

ACM Reference Format:
T.Y.S.S. Santosh and K.V.S. Aravind. 2019. Hate Speech Detection in Hindi-English Code-Mixed Social Media Text. In *6th ACM IKDD CoDS and 24th COMAD (CoDS-COMAD '19), January 3–5, 2019, Kolkata, India.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3297001.3297048

1 INTRODUCTION

In the recent years there has been a rapid increase in usage of social media platforms like Facebook, Twitter. With the increase in user generated content, particularly on social media networks, the amount of hate speech is also steadily increasing. The term 'hate speech' was formally defined as 'any communication that disparages a person or a group on the basis of some characteristics such as race, colour, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics'. So, there is a need to automatically detect such hateful content and curb the wrongful activities. Hate speech detection has several applications like sentiment analysis, investigating cyber bullying and examining socio-political controversies. In multilingual countries, people don't always tend to express their opinions in a single language. In India, people use Hindi along with English to share their opinions. This phenomenon is called Code Mixing where linguistic units such as phrases, words or morphemes of one language are embedded into an utterance of another language. Code mixing allows ease-of-communication among speakers by providing a much wider variety of phrases and expressions. But this has made the task for developing NLP tools more difficult, highlighted by [6], [19], [3]. The romanized code mixed data on social media presents additional challenges such as contractions, spelling variations and non-grammatical sentence constructions. This use of Roman script for a Hindi word may produce various spelling variations which makes NLP tasks much more challenging on code mixed text. While relevant research has been done independently on code-mixed social media texts and hate speech detection, hate speech detection on code-mixed social media texts hasn't been explored much.

The structure of paper is as follows. In Section 2, we describe about related work in the area of code mixing and hate speech detection. In Section 3, we describe our proposed architectures. In Section 4, we describe our experiments and results which include dataset description, experiment set up and the results of experiments using the architectures proposed in section 3. In the last section, we conclude our paper, followed by future work.

2 RELATED WORK

Hate speech classification: While lookup of hateful terms in a dictionary is one possible approach [18] to detect hate speech,but such methods didn't perform well [16]. Early approaches employed relatively simple classifiers and relied on manually extracted features such as bag of words, word and character n-grams to represent data [7], [14], [20]. [8] used the Continuous Bag Of Words model with paragraph2vec algorithm [13] to accurately detect hate speech than that of the plain Bag Of Words models. [1] improved accuracy by using Gradient Boosted Decision Tree classifiers using word embeddings learned using a long short-term memory (LSTM) models initialized with random embeddings and also experimented with multiple deep learning architectures to learn semantic word embeddings to handle the complexity associated with natural language constructs used in social media.

Code-Mixing: [2] performed analysis on posts from Facebook generated by Hindi-English users and showed that significant amount of code-mixing was observed in the posts. [19] created a POS tag annotated Hindi-English code-mixed corpus and reported the challenges in the Hindi-English code-mixed text. [17] addressed the problem of shallow parsing of Hindi-English code-mixed social media text and developed a system that can identify the language of the words, normalize them to their standard forms, assign their POS tag and segment them into chunks. [10] addressed the problem

CoDS-COMAD '19, January 3–5, 2019, Kolkata, India

ACM ISBN 978-1-4503-6207-8/19/01…$15.00
https://doi.org/10.1145/3297001.3297048

of Mixed-Script IR (MSIR). They also proposed a solution to handle the mixed-script term matching and spelling variation where the terms across the scripts are modelled jointly in a deep-learning architecture and can be compared in a low-dimensional abstract space. [11], [9] performed sentiment identification in code-mixed social media text.

3 METHODOLOGIES

In this section, we present the baselines and two other approaches namely sub-word level LSTM model and Hierarchical LSTM model with attention based on phonemic sub-words.

3.1 Baselines

We follow the methodology suggested in [5] as our baseline. We extract character n-grams, word n-grams, negation words, punctuation marks and used them as features to train our supervised machine learning model. We use Support Vector Machines with radial basis function kernel and Random Forest classifier. Since the size of feature vectors formed are very large, we apply chi-square feature selection algorithm which reduces the size of our feature vector.

3.2 Sub-word level LSTM model

As proposed by [15], we describe the LSTM model based on sub-word level representations. Considering the possibility of out-of-vocabulary words, word based models do not yield good results. Given the paucity of the dataset, character level models can not perform well. Incorporating linguistic information of words as sub-words into model can help to perform better. The sub-word level representations are generated as follows through 1-D convolutions on character inputs for a given sentence. We perform convolution with a filter after which we add a bias and apply a non-linearity to obtain a feature map. Thus we can get sub-word level (morpheme-like) feature map. Finally, we pool the maximal responses from feature representations corresponding to selecting sub-word representations. We then pass the sub-word representations through LSTM in order to capture the inter-relations among the words and classify the sentence as hate or non-hate. Figure 1 presents an overview of the above architecture.

3.3 Hierarchical LSTM model with attention based on phonemic sub-words

As proposed by [12], we describe the phonemic sub-word units for Hindi-English code-mixed text along with a hierarchical deep learning model which uses these sub-word units for prediction. The variations in spellings which are predominantly observed in social media text can give rise to out-of-vocabulary words and hence we segment words into phonemic sub-words. The segmentation is based on dividing word into consonant-vowel sequences(C^+V^+ acts as an approximate syllable). We expect that these sub-word units will be helpful to handle out-of-vocabulary problem which usually arises with word-level deep learning models. Our hierarchical attention based model consists of different parts namely embedding layer, syllable encoder, word encoder, word attention and output layer.

Embedding layer: Each sentence s_i is tokenized into list of words w_{ij}. Each word w_{ij} is tokenized into a list of syllables u_{ijk}. These syllables are sent as input to the embedding layer which transforms the syllable u_{ijk} into a low dimensional representation x_{ijk}.

Syllable Encoder: Given a word w_{ij} with syllables u_{ij1}, u_{ij2}, ..., u_{ijP}, where P is the number of syllables in the word. We encode each word using a bi-directional LSTM by summarizing the information of the syllables in both directions. The forward LSTM reads the syllables from u_{ij1} to u_{ijP} and backward LSTM reads the syllables from u_{ijP} to u_{ij1}.

$$\overrightarrow{h_{ijk}} = \overrightarrow{LSTM}(x_{ijk}), k \in [1, P]$$

$$\overleftarrow{h_{ijk}} = \overleftarrow{LSTM}(x_{ijk}), k \in [1, P]$$

where $\overrightarrow{h_{ijk}}$, $\overleftarrow{h_{ijk}}$, P are the forward hidden state for u_{ijk} and backward hidden state for u_{ijk}, number of syllables in the word respectively. The representation for the word w_{ij} is obtained by concatenating forward hidden state and backward hidden state $\overrightarrow{h_{ijP}}$ $\overleftarrow{h_{ij1}}$ vectors as $y_{ij} = [\overrightarrow{h_{ijP}}, \overleftarrow{h_{ij1}}]$ where y_{ij} is vector representation of word w_{ij}.

Word Encoder: Given a word representation y_{ij}, we annotate the word using Bi-Directional LSTM by summarizing information on both directions of the word, and therefore incorporate the contextual information in the annotation.

$$\overrightarrow{H_{ij}} = \overrightarrow{LSTM}(y_{ij}), j \in [1, L]$$

$$\overleftarrow{H_{ij}} = \overleftarrow{LSTM}(y_{ij}), j \in [1, L]$$

where $\overrightarrow{H_{ij}}$, $\overleftarrow{H_{ij}}$, L are the forward hidden state, backward hidden state, length of the sentence respectively. Thus we annotate each word representation H_{ij} by concatenating $\overrightarrow{H_{ij}}$ and $\overleftarrow{H_{ij}}$ as $H_{ij} = [\overrightarrow{H_{ij}}, \overleftarrow{H_{ij}}]$

Word Attention: In each sentence S_i, not all words are equally important. So, we use the attention mechanism to attend the important words and annotate the sentence representation based on them.

$$U_{ij} = tanh(WH_{ij} + B), j \in [1, L]$$

$$\alpha_{ij} = \frac{exp({U_{ij}}^T U)}{\sum_j exp({U_{ij}}^T U)}, j \in [1, L]$$

$$V_i = \sum_j \alpha_{ij} H_{ij}, j \in [1, L]$$

U_{ij} is the hidden representation of H_{ij} which is obtained by feeding H_{ij} to single-layer MLP. W and B denote trainable weights and bias matrices respectively. The importance α_{ij} of each word is measured from the similarity between U_{ij} and context vector U normalized using a softmax function. Here, U is initialized randomly and learned in the course of training process. The output vector V_i is obtained by the weighted sum of the annotations of each word with its corresponding importance as weight.

Output Layer: The output of the word attention layer is fed to a final single neuron layer, that performs binary classification

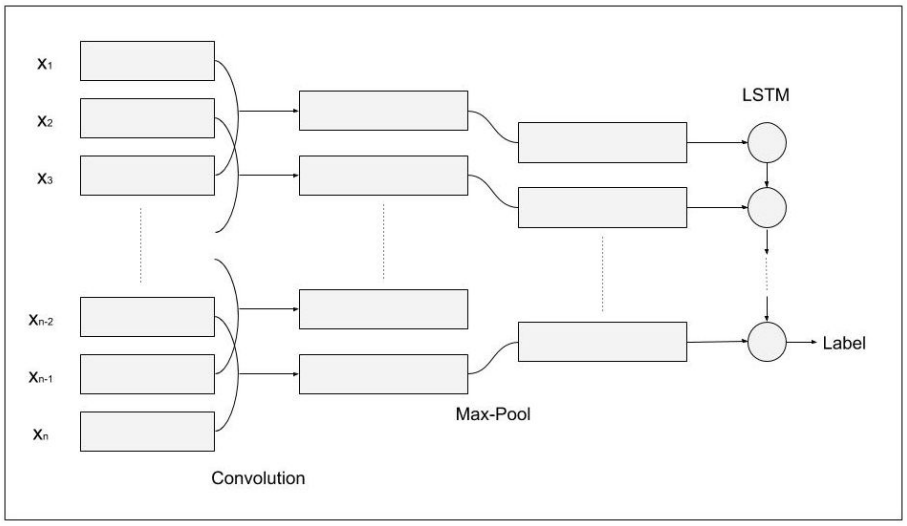

Figure 1: Sub-word level LSTM model

(logistic regression). Figure 2 shows the schematic overview of above architecture.

4 EXPERIMENTS

4.1 Dataset and Experiment settings

We experimented with dataset of 3800 Tweets [5] which consists of 2300 tweets as hate and 1500 tweets as non-hate tweets. Table 1 provides an example of a hate and a non-hate speech. It is a common practice on social media to use camel case while writing hashtags. Thus we extracted the hashtags from each tweet and extracted separate tokens from it by removing the '#' and using a hashtag decomposition approach [4] assuming it is written in camel-case. URLs, user-mentions, stop words, emoticons and punctuations are removed from tweets for further processing. We performed 10-Fold Cross Validation and calculated accuracy, recall and F1-scores. In baselines we used the chi-square feature selection algorithm to reduce feature size to 1200 [15]. We used the Scikit-learn library. In other two approaches, we used Adam optimizer to train this setup in an end-to-end fashion using batch size of 32. We used Keras deep learning library with Theano backend. We used a very simplistic deep learning architectures because of the constraint on the size of the dataset. As the datasets in this domain expand, we would like to scale up our approach to bigger architectures.

4.2 Results

Table 2 shows the results of methods described in section 3 on the hate speech detection task. Support vector machine with character n-grams, word n-grams, negation words, punctuation marks as features gives good accuracy where as Hierarchial LSTM model with attention, based on phonemic sub-words gave the good recall and F1-score by a large margin. Our objective in using this network is to use phonemic sub-words to circumvent the problem of rare words in case of word based models and unavailability of large

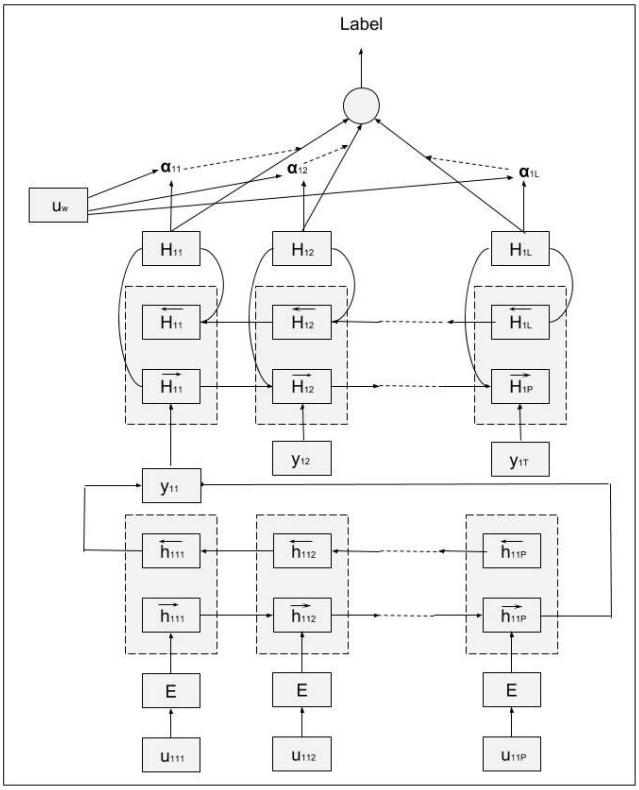

Figure 2: Hierarchical LSTM model with attention based on phonemic sub-words

hate speech	Bawli booch mein Hindustani hu Teri tarah ISI ka agent nhi tu aur Tera baap krta hai nafrat ki rajneeti hum to bhaichare wale h
non-hate speech	Sabhi jagah to media wale nafrat or jhgde to faila hi rhe hai ,ab logo k ghar me v suru ho gye

Table 1: An example of a hate and a non-hate speech

Model	Accuracy	Recall	F1-score
SVM Classifier	**70.7**	31.3	42.9
Random Forest Classifier	65.1	19.3	29.2
Sub-word level LSTM model	69.8	36.5	45.8
Hierarchical LSTM model with attention based on phonemic sub-words	66.6	**45.1**	**48.7**

Table 2: Results on models described in section 3

data-sets for training character based models and also architecture with linguistic priors may lead to the better performance on the task.

5 CONCLUSION

In this paper, we investigated different methods namely sub-word level LSTM model and Hierarchical LSTM model with attention, based on phonemic sub-words for hate speech detection on social media code-mixed text. We wish to explore other deep learning architectures which can jointly learn semantic and morphological information for hate speech detection of code mixed text in future.

REFERENCES

[1] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 759–760.

[2] 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*. 116–126.

[3] Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the first workshop on computational approaches to code switching*. 13–23.

[4] Billal Belainine, Alexsandro Fonseca, and Fatiha Sadat. 2016. Named Entity Recognition and Hashtag Decomposition to Improve the Classification of Tweets. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*. 102–111.

[5] 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 PeopleâĂŹs Opinions, Personality, and Emotions in Social Media*. 36–41.

[6] Gokul Chittaranjan, Yogarshi Vyas, Kalika Bali, and Monojit Choudhury. 2014. Word-level language identification using CRF: Code-switching shared task report of MSR India system. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*. 73–79.

[7] Karthik Dinakar, Roi Reichart, and Henry Lieberman. 2011. Modeling the detection of Textual Cyberbullying. *The Social Mobile Web* 11, 02 (2011), 11–17.

[8] 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*. ACM, 29–30.

[9] Souvick Ghosh, Satanu Ghosh, and Dipankar Das. 2017. Sentiment Identification in Code-Mixed Social Media Text. *arXiv preprint arXiv:1707.01184* (2017).

[10] Parth Gupta, Kalika Bali, Rafael E Banchs, Monojit Choudhury, and Paolo Rosso. 2014. Query expansion for mixed-script information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 677–686.

[11] Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2482–2491.

[12] Upendra Kumar, Vishal Singh Rana, Chris Andrew, Santhoshini Reddy, and Amitava Das. 2018. Consonant-Vowel Sequences as Subword Units for Code-Mixed Languages. (2018).

[13] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. 1188–1196.

[14] 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*. International World Wide Web Conferences Steering Committee, 145–153.

[15] Ameya Prabhu, Aditya Joshi, Manish Shrivastava, and Vasudeva Varma. 2016. Towards Sub-Word Level Compositions for Sentiment Analysis of Hindi-English Code Mixed Text. *arXiv preprint arXiv:1611.00472* (2016).

[16] Haji Mohammad Saleem, Kelly P Dillon, Susan Benesch, and Derek Ruths. 2017. A web of hate: Tackling hateful speech in online social spaces. *arXiv preprint arXiv:1709.10159* (2017).

[17] Arnav Sharma, Sakshi Gupta, Raveesh Motlani, Piyush Bansal, Manish Srivastava, Radhika Mamidi, and Dipti M Sharma. 2016. Shallow parsing pipeline for hindi-english code-mixed social media text. *arXiv preprint arXiv:1604.03136* (2016).

[18] Stéphan Tulkens, Lisa Hilte, Elise Lodewyckx, Ben Verhoeven, and Walter Daelemans. 2016. A dictionary-based approach to racism detection in dutch social media. *arXiv preprint arXiv:1608.08738* (2016).

[19] Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 974–979.

[20] 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*. 88–93.