A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection

by

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed S. Akhtar, Manish Shrivastava

in

16th Annual Conference of the North American Chapter of the Association for Computational Linguistics (*NAACL-2018*)

New Orleans, USA

Report No: IIIT/TR/2018/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2018

A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection

Aditya Bohra*, **Deepanshu Vijay***, **Vinay Singh, Syed S. Akhtar, Manish Shrivastava**
International Institute of Information Technology
Hyderabad, Telangana, India
{aditya.bohra, deepanshu.vijay, vinay.singh, syed.akhtar}@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

Hate speech detection in social media texts is an important Natural language Processing task, which has several crucial applications like sentiment analysis, investigating cyber bullying and examining socio-political controversies. While relevant research has been done independently on code-mixed social media texts and hate speech detection, our work is the first attempt in detecting hate speech in Hindi-English code-mixed social media text. In this paper, we analyze the problem of hate speech detection in code-mixed texts and present a Hindi-English code-mixed dataset consisting of tweets posted online on Twitter. The tweets are annotated with the language at word level and the class they belong to (Hate Speech or Normal Speech). We also propose a supervised classification system for detecting hate speech in the text using various character level, word level, and lexicon based features.

1 Introduction

With recent surge in the amount of user generated social media data, there has been a tremendous scope in automated text analysis in the domain of computational linguistics. Popularity of opinion-rich online resources like review forums and microblogging sites has encouraged users to express and convey their thoughts all across the world in real time. This often results in users posting offensive and abusive content online using hateful speech. These may be directed towards an individual or community to show their dissent. Detecting hate speech is thus important for lawmakers and social media platforms to discourage occurence of any wrongful activities. Previous research related to this task has mainly been focused on monolingual texts (Malmasi and Zampieri, 2017; Schmidt and Wiegand, 2017; Davidson et al., 2017) due to their large-scale availability. However, in multilingual societies like India, usage of code-mixed languages (among which Hindi-English is most prominent) is quite common for conveying opinions online.

Code-Mixing (CM) is a natural phenomenon of embedding linguistic units such as phrases, words or morphemes of one language into an utterance of another (Myers-Scotton, 1993; Gysels, 1992; Duran, 1994; Muysken, 2000). Following are some instances of Hindi-English code-mixed texts also transliterated in English.

T1 : *"Mujhe apne manager se nafrat hai, I want to kill that guy."*
Translation : "I hate my manager, I want to kill that guy."

T2 : *"Aaj ka day humesha yaad rahega humein because India won the World Cup! :D"*
Translation : "We'll forever remember this day because India won the World Cup! :D "

T3 : *"Jisne bhi Nirbhaya ka rape kiya should be bloody hanged till death."*
Translation : "Whoever raped Nirbhaya, should be bloody hanged till death."

It can be observed that **T1** and **T3** contain hate speech, while **T2** is an instance of normal speech.

To the best of our knowledge, currently there are no online code-mixed resources available for detecting hate speech. We believe that our initial efforts in constructing a Hindi-English code-mixed dataset for hate speech detection will prove to be extremely valuable for linguists working in this domain.

The structure of the paper is as follows. In Section 2, we review related research in the area of

* These authors contributed equally to this work.

code mixing and hate speech detection. In Section 3, we describe the corpus creation and annotation scheme. In Section 4, we present our system architecture which includes the pre-processing steps and classification features. In Section 5, we present the results of experiments conducted using various character-level, word-level and lexicon features. In the last section, we conclude our paper, followed by future work and references.

2 Background and Related Work

(Bali et al., 2014) performed analysis of data from Facebook posts generated by Hindi-English bilingual users. Analysis depicted that significant amount of code-mixing was present in the posts. (Vyas et al., 2014) created a POS tag annotated Hindi-English code-mixed corpus and reported the challenges and problems in the Hindi-English code-mixed text. They also performed experiments on language identification, transliteration, normalization and POS tagging of the dataset. (Sharma et al., 2016) 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. (Barman et al., 2014) addressed the problem of language identification on Bengali-Hindi-English Facebook comments. They annotated a corpus and achieved an accuracy of 95.76% using statistical models with monolingual dictionaries. (Raghavi et al., 2015) developed a Question Classification system for Hindi-English code-mixed language using word level resources. The shared tasks have been also organized on classifying code-mixed cross-script question and on information retrieval of Hindi-English code-mixed tweets where the task was to retrieve the top k tweets from a corpus for a given query consisting of Hind-English terms where the Hindi terms are written in Roman transliterated form(Banerjee et al., 2016). (Gupta et al., 2014) addressed the problem 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. They also did empirical analysis of the proposed method along with the evaluation results in an ad-hoc retrieval setting of mixedscript IR where the proposed method achieves significantly better results (12% increase in MRR and 29% increase in MAP) compared to other state-of-the-art baselines. (Joshi et al., 2016; Ghosh et al., 2017) performed Sentiment Identification in code-mixed social media text.

(Malmasi and Zampieri, 2017) examined methods to detect hate speech in social media. They presented a supervised classification system which uses character n-grams, word n-grams and word skip grams. They were able to achieve accuracy of 78% on dataset which contains English tweets annotated with three labels, namely, hate speech (HATE), offensive language but no hate speech (OFFENSIVE); and no offensive content (OK). (Del Vigna et al., 2017) addressed the problem of Hate speech detection for Italian language. They built their annotated corpus using comments retrieved from the Facebook public pages of Italian newspapers, politicians, artists, and groups. They conducted two different classification experiments: the first considering three different categories of hate (Strong Hate, Weak Hate and No Hate) and the second considering only two categories, No Hate and Hate, where the last category was obtained by merging the Strong Hate and Weak Hate classes. In the two experiments they were able to achieve the best accuracies of 64.61% and 72.95% respectively.

3 Corpus Creation and Annotation

We constructed the Hindi-English code-mixed corpus using the tweets posted online in last five years. Tweets were scrapped from Twitter using the Twitter Python API[1] which uses the advanced search option of twitter. We have mined the tweets by selecting certain hashtags and keywords from politics. public protests, riots, etc., which have a good propensity for the presence of hate speech. We retrieved 1,12,718 tweets from Twitter in json format, which consists of information such as timestamp, URL, text, user, re-tweets, replies, full name, id and likes. An extensive processing was carried out to remove all the noisy tweets. Furthermore, all those tweets which were written either in pure English or pure Hindi language were removed. As a result of manual filtering, a dataset of 4575 code-mixed tweets was created.

[1] https://pypi.python.org/pypi/twitterscraper/0.2.7

```
<tweet>
<id>954297321843433472<\id>
<word lang="eng">Congress</word>
<word lang="hin">ke</word>
<word lang="eng">agents</word>
<word lang="hin">ho</word>
<word lang="hin">ya</word>
<word lang="hin">maha</word>
<word lang="hin">murkh.</word>
<word lang="hin">rape</word>
<word lang="hin">rape</word>
<word lang="hin">hota</word>
<word lang="hin">hai</word>
<word lang="eng">use</word>
<word lang="hin">dalit</word>
<word lang="eng">or</word>
<word lang="hin">non</word>
<word lang="hin">dalit</word>
<word lang="eng">see</word>
<word lang="hin">Jo</word>
<word lang="hin">Kar</word>
<word lang="hin">mat</word>
<word lang="hin">dekho</word>
</tweet>
<class>
Hate Speech
</class>
```

Figure 1: Annotated Instance

3.1 Annotation

Annotation of the corpus was carried out as follows:

Language at Word Level : For each word, a tag was assigned to its source language. Three kinds of tags namely, 'eng', 'hin' and 'other' were assigned to the words by bilingual speakers. 'eng' tag was assigned to words which are present in English vocabulary, such as "School", "Death", etc. 'hin' tag was assigned to words which are present in the Hindi vocabulary such as "nafrat" (Hatred), "marna" (dying). The tag 'other' was given to symbols, emoticons, punctuations, named entities, acronyms, and URLs.

Hate Speech or Normal Speech : An instance of annotation is illustrated in Figure 1. Each tweet is enclosed within <tweet></tweet>tags. First line in every annotation consists of tweet id. Language tags are added before every token of the tweet, enclosed within <word></word>tags. Each tweet is annotated with one of the two tags (Hate Speech or Normal Speech). Hate speech is detected in 1661 tweets. Remaining 2914 code-mixed tweets in the dataset comprise of normal speech. The annotated dataset with the classification system is made available online[2].

3.2 Inter Annotator Agreement

Annotation of the dataset to detect presence of hate speech was carried out by two human annotators having linguistic background and proficiency in both Hindi and English. A sample annotation set consisting of 50 tweets (25 hate speech and 25 non hate speech) selected randomly from all across the corpus was provided to both the annotators in order to have a reference baseline so as to differentiate between hate speech and non hate speech text. In order to validate the quality of annotation, we calculated the inter-annotator agreement (IAA) for hate speech annotation between the two annotation sets of 4575 code-mixed tweets using Cohen's Kappa coefficient. Kappa score is 0.982 which indicates that the quality of the annotation and presented schema is productive.

4 System Architecture

In this section, we present our machine learning model which is trained and tested on the code-mixed dataset described in the previous sections.

4.1 Pre-processing of the code-mixed tweets

Following are the steps which were performed in order to pre-process the data prior to feature extraction.

1. **Removal of URLs:** All the links and URLs in the tweets are stored and replaced with "URL", as these do not contribute towards any kind of sentiment in the text.

2. **Replacing User Names:** Tweets often contain mentions which are directed towards certain users. We replaced all such mentions with "USER".

3. **Replacing Emoticons :** All the emoticons used in the tweets are replaced with "Emoticon".

4. **Removal of Punctuations:** All the punctuation marks in a tweet are removed. However, before removing them we store the count of

[2]https://github.com/deepanshu1995/HateSpeech-Hindi-English-Code-Mixed-Social-Media-Text

each punctuation mark since we use them as one of the features in classification.

4.2 Feature Identification and Extraction :

In our work, we have used the following feature vectors to train our supervised machine learning model.

1. **Character N-Grams (C):** Character N-Grams are language independent and have proven to be very efficient for classifying text. These are also useful in the situation when text suffers from misspelling errors (Cavnar and Trenkle, 1994; Huffman, 1995; Lodhi et al., 2002). Group of characters can help in capturing semantic meaning, especially in the code-mixed language where there is an informal use of words, which vary significantly from the standard Hindi and English words. We use character n-grams as one of the features, where n vary from 1 to 3.

2. **Word N-Grams (W) :** Bag of word features have been widely used to capture emotion in a text (Purver and Battersby, 2012) and in detecting hate speech (Warner and Hirschberg, 2012). Thus we use word n-grams, where n vary from 1 to 3 as a feature to train our classification models.

3. **Punctuations (P):** Punctuation marks can also be useful for hate speech detection. Users often use exclamation marks when they want to express strong feelings. Multiple question marks in the text can denote anger and dissent. Usage of an exclamation mark in conjunction with the question mark indicates annoyed feeling. We count the occurrence of each punctuation mark in a sentence and use them as a feature.

4. **Negation Words (N) :** A list of negation words was taken from Christopher Pott's sentiment tutorial[3]. We count the number of negations in a tweet and use the count as a feature.

5. **Lexicon (L) :** Users often use a particular set of words to express hate. Previous research on various NLP tasks such as Emotion Detection has demonstrated that lexicon features provide a significant gain in classification accuracy when combined with corpus-based features, if training and testing sets are drawn from the same domain (Mohammad, 2012). We identified 177 Hindi and English hate words from the dataset and took them as a feature for classification.

[3]http://sentiment.christopherpotts.net/lingstruc.html

Features	Accuracy
Character N-Grams	71.6
Word N-Grams	70.1
Punctuations	63.6
Lexicon	64.2
Negations	63.6
All features	**71.7**

Table 1: Accuracy of each feature using Support Vector Machines

5 Results

We performed experiments with two different classifiers namely Support Vector Machines with radial basis function kernel and Random Forest Classifier. Since the size of feature vectors formed are very large, we applied chi-square feature selection algorithm which reduces the size of our feature vector to 1200[4]. For training our system classifier, we have used Scikit-learn (Pedregosa et al., 2011). In all the experiments, we carried out 10-fold cross validation. Table 1 and Table 2 describe the accuracy of each feature along with the accuracy when all features are used, in the case of Support vector machine and Random forest classifier respectively. Support vector machine performs better than Random forest classifier and gives a highest accuracy of 71.7% when all features are used. Character N-Grams proved to be most efficient in SVM, while Word N-Grams resulted in most accuracy in the case of Random Forest Classifier.

6 Conclusion and Future Work

In this paper, we present an annotated corpus of Hindi-English code-mixed text, consisting of tweet ids and the corresponding annotations. We also present the supervised system used for detection of Hate Speech in the code-mixed text. The

[4]The size of feature vector was decided after empirical fine tuning

Features	Accuracy
Character N-Grams	66.8
Word N-Grams	**69.9**
Punctuations	63.2
Lexicon	63.8
Negations	63.6
All features	66.7

Table 2: Accuracy of each feature using Random Forest Classifier

corpus consists of 4575 code-mixed tweets annotated with hate speech and normal speech. The words in the tweets are also annotated with source language of the words. The features used in our classification system are character n-grams, word n-grams, punctuations, negation words and hate lexicon. Best accuracy of 71.7% is achieved when all the features are incorporated in the feature vector using SVM as the classification system.

As a part of future work, the corpus can be annotated with part-of-speech tags at word level which may yield better results. Moreover, the annotations and experiments described in this paper can also be carried out for code-mixed texts containing more than two languages from multilingual societies, in future.

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.

Somnath Banerjee, Kunal Chakma, Sudip Kumar Naskar, Amitava Das, Paolo Rosso, Sivaji Bandyopadhyay, and Monojit Choudhury. 2016. Overview of the mixed script information retrieval (msir) at fire-2016. *Organization (ORG)*, 67:24.

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*, pages 13–23.

William B Cavnar and John M Trenkle. 1994. N-gram-based text categorization. *Ann arbor mi*, 48113(2):161–175.

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*.

Fabio Del Vigna, Andrea Cimino, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on facebook. *In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy*.

Luisa Duran. 1994. Toward a better understanding of code switching and interlanguage in bilinguality: Implications for bilingual instruction. *The Journal of Educational Issues of Language Minority Students*, 14(2):69–88.

Souvick Ghosh, Satanu Ghosh, and Dipankar Das. 2017. Sentiment identification in code-mixed social media text. *arXiv preprint arXiv:1707.01184*.

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*, pages 677–686. ACM.

Marjolein Gysels. 1992. French in urban lubumbashi swahili: Codeswitching, borrowing, or both? *Journal of Multilingual & Multicultural Development*, 13(1-2):41–55.

Stephen Huffman. 1995. Acquaintance: Language-independent document categorization by n-grams. Technical report, DEPARTMENT OF DEFENSE FORT GEORGE G MEADE MD.

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*, pages 2482–2491.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting hate speech in social media. *arXiv preprint arXiv:1712.06427*.

Saif Mohammad. 2012. Portable features for classifying emotional text. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 587–591. Association for Computational Linguistics.

Pieter Muysken. 2000. *Bilingual speech: A typology of code-mixing*, volume 11. Cambridge University Press.

Carol Myers-Scotton. 1993. Dueling languages: Grammatical structure in code-switching. *Oxford University Press*.

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.

Matthew Purver and Stuart Battersby. 2012. Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491. Association for Computational Linguistics.

Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, and Manish Shrivastava. 2015. Answer ka type kya he?: Learning to classify questions in code-mixed language. In *Proceedings of the 24th International Conference on World Wide Web*, pages 853–858. ACM.

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.

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*.

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)*, pages 974–979.

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.