Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, pages 42–48
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

A Comparative Study of Different State-of-the-Art Hate Speech Detection Methods for Hindi-English Code-Mixed Data

Priya Rani, Shardul Suryawanshi, Koustava Goswami,
Bharathi Raja Chakravarthi, Theodorus Fransen, and John P. McCrae
Insight SFI Research Centre for Data Analytics, Data Science Institute, National University of Ireland Galway
{priya.rani, shardul.suryawanshi, koustava.goswami, bharathi.raja, theodorus.fransen, john.mccrae}@insight-centre.org

Abstract

Hate speech detection in social media communication has become one of the primary concerns to avoid conflicts and curb undesired activities. In an environment where multilingual speakers switch among multiple languages, hate speech detection becomes a challenging task using methods that are designed for monolingual corpora. In our work, we attempt to analyze, detect and provide a comparative study of hate speech in a code-mixed social media text. We also provide a Hindi-English code-mixed data set consisting of Facebook and Twitter posts and comments. Our experiments show that deep learning models trained on this code-mixed corpus perform better.

Keywords: Hate Speech, Code mixing, Convolutional Neural Networks

1. Introduction

Hate speech is a direct or indirect statement targeted towards a person or group of people intended to demean and brutalize another or use derogatory language on the basis of ethnicity, religion, disability, gender or sexual orientation (Schmidt and Wiegand, 2017). Due to the massive rise in user-generated content from social media, hate speech has also steadily increased. Hate speech, targeting a particular individual or group of people, can cause personal trauma, cyberbullying, panic in the society, and discrimination. In response to the growth in the hate content from social media, there has been a large number of works on automatic hate speech detection to alleviate online harassment (Warner and Hirschberg, 2012; Zimmerman et al., 2018; MacAvaney et al., 2019; Ibrohim and Budi, 2019; Nobata et al., 2016).

Code mixing is a phenomenon which occurs when the speaker uses two languages together in the course of a single utterance (Wardhaugh, 1986; Chakravarthi et al., 2018; Chakravarthi et al., 2019). The speaker makes use of the grammar or lexicon from more than one language. It is considered as a natural and common phenomenon in multilingual societies and is reflected in user-generated content on social media (Ranjan et al., 2016; Jose et al., 2020; Priyadharshini et al., 2020; Chakravarthi et al., 2020b; Chakravarthi et al., 2020a). The task of identifying hate speech becomes even more challenging when the content is code-mixed since lexical items, phrases and sentences from different languages may co-exist within a sequence, and computational models are required to recognize and process these simultaneously. Hate Speech is common on social media, and content generated by Indian-language speakers is no exception (Suryawanshi et al., 2020a; Suryawanshi et al., 2020b). It assumes an additional significance due to high internet infiltration and rich linguistic diversity. In addition to this, the use of the Roman script for Indian languages mixed with native scripts is widespread among social networking sites due to difficulty in typing tools and familiarity with English, which adds to the overall complexity of the problem.

While there is some relevant and independent work on code-mixed social media content, few efforts have been made to detect hate speech in Hindi-English code-mixed data. In the light of the gap in this research area, our contributions described in this paper are the following:

- An annotated Hindi-English code-mixed data set containing hate speech. To the best of our knowledge, this is the first Hindi-English code-mixed data set which contains posts/tweets written in both the Roman and the native Devanagari script.
- A comparative study of performance of five different classifiers including machine learning and deep learning on the three different Hindi-English code-mixed data sets.
- An extensive discussion of the micro F1 score of all the trained models for each data set, not provided in the experiments reported on by Bohra et al. (2018).

We have also evaluated the performance of the classifiers and deep learning model on the same data set used by Bohra et al. (2018). The rest of the paper is organized as follows. We explain related works in Section 2. Section 3. presents the details of the data set. Section 4. reports on approaches we used to classify the hate speech content. In Section 5., we present our results accompanied by a detailed error analysis. Section 6. concludes the paper.

2. Related Work

In the digital era of the global world, various areas of research have studied computer-mediated communication from different perspectives. Language usage on social media websites, in emails and in chat rooms has been studied concerning phenomena such as speech acts, code-switching, gender, communalism, politeness and impoliteness. Lots of research has been done on gender and sexuality in hate speech detection, and there has been significant progress over time.

Important early work on hate speech detection was carried out by Spertus (1997), who built a prototype system *Smokey* using a C4.5 decision tree generator to determine feature-based rules that could categorize abusive messages. Since then, hate speech detection has achieved milestones, and several models have been trained to detect hate speech. Yin et al. (2009) were the first to use a supervised learning approach to detect harassment on web 2.0. They classified social media posts using a support-vector machine (SVM) based on local contextual and sentiment features. Malmasi and Zampieri (2017) examined character n-grams, word n-grams and skip-grams to detect hate speech in social media. They trained their classifier on an English data set with three labels and achieved an accuracy of 78%. A Hindi-English code-mixed data set was created to study the problem of hate speech detection in such data. They classified the tweets using character n-grams, word n-grams, punctuation, lexicon and negations features with an SVM and random forest. The best result was obtained by SVM with an accuracy of 71.7% when all the features were used together to detect hate speech (Bohra et al., 2018). A convolution neural network model was proposed by Mathur et al. (2018) to detect offensive tweets in Hindi-English code switched language. Bohra et al. (2018) created a Hindi-English code-mixed data set to study the problem of hate speech detection in such data. The data set contains Twitter data in the Roman script only. They classified the tweets using character n-grams, word n-grams, punctuation, lexicon and negations features with an SVM and random forest. They reported results on the linear classifying approach that uses hand-engineered features. The best result was obtained by SVM with an accuracy of 71.7% when all the features were used together to detect hate speech.

3. Corpus Creation and Annotation

Taking into account the aim of the present study, we chose to use social media data, as this data is best known for code-mixing. The corpus used for the study comes from two of the biggest social networking sites: Facebook and Twitter.

3.1. Corpus collection

Three different data sets have been used for the current study.

- The first data set was collected from Github [1]. Data set-1 consist of 4575 Hindi-English code-mixed annotated tweets in the Roman script only. Tweets were extracted from twitter using the Twitter API. In order to remove the noise from the data set, rigorous preprocessing was carried out, which resulted in the removal of URLs and punctuation, replacing user names and emoticons (Bohra et al., 2018).
- Data set-2 was taken from a Shared Task called HASOC, which was organised at FIRE 2019. It consists of 4665 annotated posts partially collected from Twitter and Facebook. The collection was done with the help of the Twitter API using specific hashtags and keywords which helped in crawling an unbiased data set (Mandl et al., 2019).
- In addition to Data set-1 and Data set-2 set we created a third data set (Data set-3) which has also been used for an aggression detection task (Kumar et al., 2018). This unannotated data set contains 3367 posts and tweets which were annotated by us. The data for the current corpus was crawled from Facebook and Twitter. The data was collected using some of the popular hashtags around such contentious themes as a beef ban, India vs Pakistan cricket matches, election results, opinions on movies, etc., i.e., topics that are typically discussed among Indians and may give rise to hate speech.

Detailed statistics of the three data sets are provided in Table 1.

Data Set	Hate	Not-Hate	Total
DATA SET-1 (Bohra et al., 2018)	2290	2289	4579
DATA SET-2 (HASOC data set)	2419	2246	4665
DATA SET-3 (ours)	478	2889	3367

Table 1: Statistics of the three data sets. Data set-1 contains Posts/Tweets in the Roman script only, Data set-2 has the Posts/Tweets in Devanagari script only and Data set-3 (our data) has Posts/Tweets in both the Roman and the Devanagari script.

3.2. Annotation Guidelines

Annotation is an integral part in the development of any automatic recognition system. Annotated data provides useful quantitative information about the occurrence of certain contextual features. As the first two data sets were already annotated, we carried out annotation only for our data set. The annotation was carried out using a flat tag set described in the annotation guideline[2]. It is used for training and testing the system for automatic hate speech recognition. A simple binary classification method in which we distinguish between hate speech and non-hate speech posts was applied. The two labels use for this categorization are *Hate* and *Not Hate*.

- A post has been marked as hate if the post contains any linguistic behaviour which is intended to target an individual or community and shows dissent using offensive and abusive content. This includes both direct and indirect offensive language as well as threats. Indirect offensive posts are expressed through sarcasm, satire or apparently polite language. Hate speech content also includes offensive reference to one's sexuality and sexual orientation as well as race and religion, i.e., posts targeting a specific community to demean them. Any post in a thread endorsing previously expressed hate speech was also marked as hate (HATE).

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

[2] https://www.dropbox.com/s/lydv9tt7kh4k01b/Hate%20speech%20annotation%20-Data%20line.pdf?dl=0

Label	Examples
Hate	Tweets: "rssabvp vhp bajarangdal have no balls whenever bjp came at center laffada bjpvhp rssabvp start voilence in streets colleges hotels pubs as if they have no balls" Translation : RSS ABVP VHP Bajarangda has no balls. Whenever Bjp came to power, there is chaos everywhere. BJP VHP RSS ABVP starts violence in streets, colleges, hotels, and pubs. Tweets : " मर्द की कीमती चीज उसकी ज़ुबान होती है सीना तो बहुत से हीजड़ो का भी 56 इंच होता है .. #BHU_लाठीचार्ज #bhu_molestation #BHUProtests" Translation: The most precious thing about a man is his words, not the length of the chest as even a spado has 56" chest.
Not Hate	Tweets: "Lathi Charge in BHU. छात्राओं पर बरसी लाठी.#bhu_molestation #UnsafeBHU #BHULathiCharge #BHUUproar #BHURow " Translation: Lathi Charge in BHU. Female students were beaten by sticks Tweets: "@gurmeetramrahim एक तेरा सहारा मिल जाए रबा दुनिया दी परवाह नहीं करना।। blessing chahiye bht sari msg dikhani h #blockbustermsg " Translation: If I get your support @gurmeetramrahim than I won't care for anything else, I need your blessing to show the messages.

Figure 1: Examples of the posts/tweets with their labels

- Posts which do not contain any offence or profanity, either covert or overt, and do not target any individual, community or group were marked as non-hate (NOT HATE).

A list of relevant examples illustrating this binary classification is shown in Figure 1.

3.3. Inter-annotator agreement

In order to test the validity of the annotation, an inter-annotator agreement was calculated using Kripendorff's α using Krippendorf 0.32 based on the Thomas Grill implementation[3]. The annotation was completed by six annotators: three male and three female in three different phases. In order to make the annotation process more accessible and user-friendly, 33 Google forms were made which contained the necessary annotator information, annotation scheme and 100 posts in each Google form. In the very first phase, 500 posts were annotated by all the six annotators. An inter-annotator agreement was calculated before the completion of the first annotation phase, after which changes in the annotation guidelines [4] were made since the inter-annotator agreement score was below par for hate speech detection. The second phase of the annotation was conducted with another set of 500 posts/tweets.
While calculating the inter-annotator agreement after the second round of annotation, we found that one of the annotators had difficulty understanding social media language while another annotator was unable to finish the annotation task; consequently, the inter-annotator agreement was very poor. Therefore we eliminated both annotators, which resulted in a much higher agreement score compared to the previous score. After completion of the second round of annotation, a preliminary experiment was done to train the system, followed by a third phase of annotation, conducted on the rest of the tweets. The final inter-annotator agreement was calculated on 4 sets x 3367 posts each. Krippendorff α score turned out to be 0.47, which is quite reliable.

[3] https://pypi.org/project/krippendorff/

[4] https://github.com/sharduls007/Hate_speech_detection_Hindi_English_codemixed

In those cases where annotators did not agree, there was generally not enough context to infer the true meaning and intent of a post. Examples of such posts are given in the next subsection.

3.4. Complicated cases

The results of the inter-annotator experiment after the completion of the first phase of annotation gave very poor agreement among the annotators. One of the main reasons for the poor agreement among the annotators was the annotation guidelines. The initial annotation guidelines were not adequate enough to pinpoint important distinctions between hate speech and non-hate speech and the interpretation of the tags as well as hashtags. Therefore, specific changes were made in the annotation guidelines to continue the second phase of annotation. Secondly, several posts were not very explicit from a pragmatical point of view; hence, each annotator made their own subjective inference about the post. A few instances are being discussed here.
Example 1 and 2 show a strong criticism of the BJP government by the users on specific events that happened recently. Rather than marking these examples as non-hate, one of the annotators felt that these posts are more than mere criticisms; these were perceived as an insult to the current government, i.e., as hate speech, where users are targeting and demeaning a particular political organisation.

(1) *The protest against #bhu_molestation and the way govt is dealing again shows how scared the BJP is of independent movements #BHU*

(2) *Sirf banaras ghumiye mat yaha ke bare me sooche bhi #bhu_molestation*
Translation - Do not think about Banaras just come and take a tour.

Another set of tweets which were difficult to annotate were the ones which consist of one single phrase and hashtags as given in examples 3 and 4. Whether the words in these tweets reflect mere criticism or contain demeaning content

Models trained	Data set-1	Data set-2	Data set-3	Combined
SVM	0.62	0.52	0.87	0.64
MNB	0.63	0.66	0.87	0.65
KNN	0.63	0.60	0.87	0.50
DT	0.57	0.65	0.85	0.66
Character-lavel CNN	0.71	0.74	0.82	0.86

Table 2: Accuracy of linear classifiers and character level CNN model trained individually and combined on the three data sets

Models	Data set-1	Data set-2	Data set-3	Combined
SVM	0.38	0.34	0.47	0.39
MNB	0.42	0.64	0.47	0.46
KNN	0.53	0.54	0.53	0.47
DT	0.55	0.61	0.61	0.65
Character-label-CNN	0.67	0.74	0.71	0.74

Table 3: Micro F1 score of the trained linear classifiers and character level CNN model trained individually and combined on the three data sets

and explicitly target some individual or group is not very clear and hence quite subjective.

(3) *landacquisitionbill #landacquisitionordinance !!!*

(4) *abki_bar_ beti_par_war #bhu_molestation*
Translation - Violence against daughters

In order to tackle the difficulty in annotating these cases we redefined the definition of hate speech for our data set. We marked the tweets/posts as hate speech only if they directly or indirectly target an individual, a group or an organisation based on race, religion, caste or gender. Posts which merely criticize such entities are not considered hate speech. We also marked posts/tweets which led to any kind of violence towards any individual, group or organisation as hate speech.

4. Classification Performance

All three data sets were used for the hate/non-hate detection task with traditional machine learning and deep learning algorithms. We conducted the experiments with four different machine learning classifiers, namely a support-vector machine (SVM), K-Nearest Neighbours (KNN), multinomial naïve Bayes (MNB) and a decision tree (DT). Term frequency (TF) weighting was employed as feature.

For the Deep Learning model, we experimented with a character-based Convolution Neural Network (CNN) (Zhang et al., 2015). The idea behind adopting the state-of-the-art model is that Twitter data contains sentences with lots of different characters (e.g., hashtags, emoticons) which are an inherent part of the message being communicated. A character-based CNN model takes all these character sequences into account, pre-empting the need for pre-processing and reducing the need for feature engineering. It was hypothesised that it should give a better understanding of the sentences compared to the linear classifiers in terms of defining classes. Therefore, no feature engineering or pre-processing was carried out. The CNN model is capable of taking all the characters into account to build a character embedding space. As these posts are short sentences, we have adjusted the number of filters to 128 compare to main paper where 256 filters are used and have kept the filter size as it is which 7*7 with convolution layers, two dense layers which used 1024 neuron and 50% dropout to adjust the overfitting issue keeping in mind that the texts are short text.

Out of the total data in each data set, 20% was set aside as test set and 10% as validation set. The remaining 70% of the data was used to train the models. More extensive experimentation and research were performed using our data set to show problems of the code-mixed text. One of the main challenges while building the model was the class distribution imbalance in data set-3, wherein it contains less hate-speech than non-hate-speech, which was forcing the model for imbalance training. To overcome the issue, we have taken the help of weighted classes where we have calculated the distribution of two classes 'hate-speech' and 'non-hate-speech' over the data. Based on the calculation, a weight of ratio 1:6 was given to the classes, which means the class 'not hate-speech' has six times higher weights of class 'hate speech' while computing the loss function. In this case the loss function will not be only based on the main class distribution data but the loss becomes a weighted average when the weight of each sample is specified by class weights and its corresponding class. Thus weighting the data helped the model to be trained more accurately.

4.1. Results

Overall we see varying performance across the classifier, with some performing much better out-of-sample than others. Every experiment was carried out with each data set once and also on the combined data set. Table 2 describes the accuracy for each data set using SVM, KNN, MNB and DT. The accuracy for Data set 1, 2 and 3 and the Combined data set using the CNN model is 0.71, 0.74, 0.82 and 0.86, respectively. Table 3 shows the micro F1 score of the models trained with the data sets. The F1 scores using the CNN model for Data set 1-3 and the Combined data set are 0.67, 0.74 0.71 and 0.74, respectively. It was found that

the character-level CNN model gives a better performance than the other classifiers in all cases. Looking at the micro F1 score of the models, we can observe that the character-level CNN model is quite good with "real" social media data as contained in our data set. When we say "real" data, we mean natural, raw data, not subjected to pre-processing, containing a high level of code-mixing. It was fed into the model with all the stop words, punctuation, emoticons, URLs and hashtags. SVM and MNB perform worst with an identical F1 score of 0.47. KNN performs slightly better with 0.53, while DT is better again with 0.61. The reason behind the poor performance of the classifiers is that these need cleaned data.

Model	Accuracy
(Bohra et al., 2018) (SVM)	0.71
(Bohra et al., 2018) (Random Forest)	0.66
Character-level CNN	0.71

Table 4: A comparison of the accuracy of the Linear approaches in the baseline paper with our Deep learning model for Data set-1.

As mentioned in the previous section, one of the data sets was developed by Bohra et al. (2018). We compared the results of their experiment (which we treat as the baseline) with our CNN model. Table 4 compares the results based on the accuracy obtained by the baseline paper and our CNN model. It is interesting to note that the baseline experiment with the SVM using Character N-Grams, Word N-Grams, Punctuation, Lexicon and Negations as the features obtains the same accuracy as the CNN model which is 0.71, while random forest obtains an accuracy of 0.66. It would have been much easier to compare the performance of the two systems if Bohra et al. (2018) had reported the F1 score of their experiments.

5. Manual Evaluation

To understand the shortcomings of the models and to get a deeper understanding of the problems associated with code-mixed data classification, a manual inspection has been performed on a set of wrongly classified sentences.

(5) *Bhai tu khud **rape** karega to bhi kuch nahi bolenge. Khush?*
Translation - Brother even if you do the rape, we will not say anything. Happy?

(6) *bjp Wale rajyo me **murder** ya rape nahi hote kya...*
Translation - No rape has been done in BJP ruled states.

A possible reason for the fact that example (5) and example (6) were wrongly marked as hate speech is the presence of lexical "rape" and "murder", shown in bold letters; the model might have taken these as a key for a hate-speech utterance.

(7) *Once a chutiya always a chutiya...*
Translation - Once a fucker always a fucker

(8) *Yup and this is a most disturbing part of this. Yaani yaar nobody is going to ask the girl even **rape** ho jaey k aagy uski life kysi guzray gi.*
Translation - Yup and this is a most disturbing part of this. It means even if a girl has been raped; no one is going to ask her how her life will be in future.

The linear classifiers could not classify most of the tweets correctly if the sentence structure is complicated, as shown in example number 7 and 8 where Hindi words are incorporated into the English word order. The case becomes even more complicated when one part of the tweet is represented with English word order and the other with Hindi. The fact that example 7 and 8 were correctly classified by the character-level CNN model shows that the deep learning method performed much better than the linear classifiers. It is likely that since the CNN model classifies the tweets on character basis, the context as well as the linguistic structure is more appropriately captured than in the case of the other classifiers.

(9) *lagta hai ki kiran bedi ki jamanat bi japt ho gayi! #delhidecides*
Translation - It seems that Kiran Bedi's bail was also confiscated

The tweets which were sarcastic, such as the one in example (9), also played an important role as these were misclassified. The tweets target one of the individuals from a leading political organisation, and as the presence of a targeted entity in an utterance is obligatory in our definition of hate speech, this tweet should have been marked as hate. However, the system marked it as not hate. Other kinds of "indirect hate tweets" were not correctly classified by either the linear classifiers or the CNN model.

(10) *#FallofBJPStarts #bhu_molestation #BHUunsafe @KPadmaRani1 @ neo_pac @ pankhuripathak @ polysmind*

Moreover, the tweets (see example number 10) which contain only hashtags were classified randomly by linear classifiers. On the other hand, the CNN model marked all theses tweets as not-hate. This is the most interesting and debatable case; even the annotators faced difficulty in annotating tweets of this type due to the lack of the written context, which is necessary to infer the real intention of the users, and agreeing on one tag.

6. Conclusion and Future Work

In this paper, we presented an annotated corpus of Hindi-English code-mixed text, consisting of tweets and the corresponding annotations. We have discussed the development of a hate speech annotated data set of 3.5k tweets and Facebook comments in English-Hindi code-mixed language. We have discussed the annotation scheme that was used to annotate the data set. We believe that the annotation of hate speech or any other cyberbullying task depends on how we define it and is necessary to state our definition clearly to the annotators. This data set could prove to

be an invaluable resource for understanding as well as automatically identifying hate speech and other related phenomena like trolling over the web, mainly on social media platforms.

We have also given a description of the supervised systems built using linear classifiers and a character-level CNN model for Hate Speech detection on three different data sets. In contrast to linear methods, the deep learning model was able to capture the syntax and semantics of the hate speech more accurately even in the case of unbalanced and unprocessed data set. Thus, we could observe the fundamental difference in the way linear classifiers like SVM and CNN models learn.

In the future, we plan to apply and experiment with techniques that could successfully cover/identify larger linguistic patterns that our shallow parses currently cannot detect. We also plan to model a system which could be useful for detecting hate speech in closely-related and minority language code-mixed data.

7. Acknowledgment

This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 (Insight), SFI/12/RC/2289_P2 (Insight_2), & SFI/18/CRT/6223 (CRT-Centre for Research Training in Artficial Intelligence) co-funded by the European Regional Development Fund as well as by the EU H2020 programme under grant agreements 731015 (ELEXIS-European Lexical Infrastructure), 825182 (Prêt-à-LLOD), and Irish Research Council grant IRCLA/2017/129 (CARDAMOM-Comparative Deep Models of Language for Minority and Historical Languages). The authors are grateful to Ajay Bohra and his team for sharing their data set and for their support. We would also like to thank our annotators for their contribution and lending us their precious time.

8. 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*, pages 36–41.

Chakravarthi, B. R., Arcan, M., and McCrae, J. P. (2018). Improving wordnets for under-resourced languages using machine translation. In *Proceedings of the 9th Global WordNet Conference (GWC 2018)*, page 78.

Chakravarthi, B. R., Priyadharshini, R., Stearns, B., Jayapal, A., S, S., Arcan, M., Zarrouk, M., and McCrae, J. P. (2019). Multilingual multimodal machine translation for Dravidian languages utilizing phonetic transcription. In *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages*, pages 56–63, Dublin, Ireland, August. European Association for Machine Translation.

Chakravarthi, B. R., Jose, N., Suryawanshi, S., Sherly, E., and McCrae, J. P. (2020a). A sentiment analysis dataset for code-mixed Malayalam-English. In *Proceedings of the 1st Joint Workshop of SLTU (Spoken Language Technologies for Under-resourced languages) and CCURL (Collaboration and Computing for Under-Resourced Languages) (SLTU-CCURL 2020)*, Marseille, France, May.

Chakravarthi, B. R., Muralidaran, V., Priyadharshini, R., and McCrae, J. P. (2020b). Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop of SLTU (Spoken Language Technologies for Under-resourced languages) and CCURL (Collaboration and Computing for Under-Resourced Languages) (SLTU-CCURL 2020)*, Marseille, France, May. European Language Resources Association (ELRA).

Ibrohim, M. O. and Budi, I. (2019). Multi-label hate speech and abusive language detection in Indonesian twitter. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 46–57, Florence, Italy, August. Association for Computational Linguistics.

Jose, N., Chakravarthi, B. R., Suryawanshi, S., Sherly, E., and McCrae, J. P. (2020). A survey of current datasets for code-switching research. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*.

Kumar, R., Reganti, A. N., Bhatia, A., and Maheshwari, T. (2018). Aggression-annotated corpus of hindi-english code-mixed data. *In the Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.

MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N., and Frieder, O. (2019). Hate speech detection: Challenges and solutions. *PLOS ONE*, 14(8):1–16, 08.

Malmasi, S. and Zampieri, M. (2017). Detecting hate speech in social media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 467–472, Varna, Bulgaria, September. INCOMA Ltd.

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., and Patel, A. (2019). Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 14–17.

Mathur, P., Shah, R., Sawhney, R., and Mahata, D. (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.

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., and Chang, Y. (2016). Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 145–153, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Priyadharshini, R., Chakravarthi, B. R., Vegupatti, M., and McCrae, J. P. (2020). Named entity recognition for code-mixed Indian corpus using meta embedding. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*.

Ranjan, P., Raja, B., Priyadharshini, R., and Balabantaray, R. C. (2016). A comparative study on code-mixed data of indian social media vs formal text. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pages 608–611, Dec.

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*, pages 1–10, Valencia, Spain, April. Association for Computational Linguistics.

Spertus, E. (1997). Smokey: Automatic recognition of hostile messages. In *Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence*, pages 1058–1065.

Suryawanshi, S., Chakravarthi, B. R., Arcan, M., and Buitelaar, P. (2020a). Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, Marseille, France, May. European Language Resources Association (ELRA).

Suryawanshi, S., Chakravarthi, B. R., Verma, P., Arcan, M., McCrae, J. P., and Buitelaar, P. (2020b). A dataset for troll classification of Tamil memes. In *Proceedings of the 5th Workshop on Indian Language Data Resource and Evaluation (WILDRE-5)*, Marseille, France, May. European Language Resources Association (ELRA).

Wardhaugh, R. (1986). An introduction to socinguistic.

Warner, W. and Hirschberg, J. (2012). Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26, Montréal, Canada, June. Association for Computational Linguistics.

Yin, D., Xue, Z., Hong, L., Davison, B. D., Kontostathis, A., and Edwards, L. (2009). Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB*, 2:1–7.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

Zimmerman, S., Kruschwitz, U., and Fox, C. (2018). Improving hate speech detection with deep learning ensembles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).