Aggression Detection on Social Media Text Using Deep Neural Networks

by

Vinay Singh, Aman Varshney, Syed S. Akhtar, Deepanshu Vijay, Manish Shrivastava

in

2018 Conference on Empirical Methods in Natural Language Processing
(*EMNLP-2018*)

Brussels, Belgium

Report No: IIIT/TR/2018/-1

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

Aggression Detection on Social Media Text Using Deep Neural Networks

Vinay Singh, Aman Varshney, Syed S. Akhtar, Deepanshu Vijay, Manish Shrivastava
Language Technologies Research Centre (LTRC)
International Institute of Information Technology Hyderabad, Telangana, India
{vinay.singh, aman.varshney, syed.akhtar, deepanshu.vijay}@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

In the past few years, bully and aggressive posts on social media have grown significantly, causing serious consequences for victims/users of all demographics. Majority of the work in this field has been done for English only. In this paper, we introduce a deep learning based classification system for Facebook posts and comments of Hindi-English Code-Mixed text to detect the aggressive behaviour of/towards users. Our work focuses on text from users majorly in the Indian Subcontinent. The dataset that we used for our models is provided by **TRAC-1**[1] in their shared task. Our classification model assigns each Facebook post/comment to one of the three predefined categories: "Overtly Aggressive", "Covertly Aggressive" and "Non-Aggressive". We experimented with 6 classification models and our CNN model on a 10 K-fold cross-validation gave the best result with the prediction accuracy of 73.2%.

1 Introduction

It is observed that multilingual speakers often switch back and forth between languages when speaking or writing, mostly in informal settings. This language interchange involves complexing grammar, and the terms "code-switching" and "code-mixing" are used to describe it (Lipski, 1978). Code-mixing refers to the use of linguistic units from different languages in a single utterance or sentence, whereas code-switching refers to the co-occurrence of speech extracts belonging to two different grammatical systems (Gumperz, 1982). As both phenomena are frequently observed on social media platforms in similar contexts, we have considered the Code-Mixing scenario for our work.

Following is an instance from the dataset used:

T1 : *"Post tabah krne se kuch nhi hoga 2 k badale 200 ko maro"*

Translation: *"No point in destroying the Post, kill 200 in return for your 2 dead."*

Due to the massive rise of user-generated web content, in particular on social media networks, the amount of hate, aggressive, bully text is also steadily increasing. It has been estimated that there has been an increase of approximately 25% in the number of tweets per minutes and 22% increase in the number of Facebook posts per minute in the last 3 years. It is estimated that approximately 500 million tweets are sent per day, 4.3 billion Facebook messages are posted and more than 200 million emails are sent each day, and approximately 2 million new blog posts are created daily over the web [2]. Over the past years, interest in online hate/aggression/bullying detection and particularly the automatization of this task has continuously grown, along with the societal impact of the phenomenon (Ring, 2013). Natural language processing methods focusing specifically on this phenomenon are required since basic word filters do not provide a sufficient remedy. What is considered as an aggressive text might be influenced by aspects such as the domain of an utterance, its discourse context, as well as context consisting of co-occurring media objects (e.g. images, videos, audio), the exact time of posting and world events at this moment, identity of author and targeted recipient.

Hence, we can say that aggression and bullying by/against an individual can be performed in several ways beyond just using obvious abusive

[1] https://sites.google.com/view/trac1/shared-task?authuser=0

[2] https://www.gwava.com/blog/internet-data-created-daily

Proceedings of the Second Workshop on Abusive Language Online (ALW2), pages 43–50
Brussels, Belgium, October 31, 2018.

language (Vandebosch and Van Cleemput, 2008) (Sugandhi et al., 2015) – e.g., via constant sarcasm, trolling, etc. This can have deep effects on one's mental as well as social health and status (Phillips, 2015).

The structure of this paper is as follows. In Section 2, we review related research in the area of hate/aggression/bullying detection in social media texts. In Section 3, we describe the process of dataset creation which is a work of (Kumar et al., 2018). In Section 4, we discuss the pre-processing and data statistics. In Section 5, we summarize our classification systems and the construction of the feature vectors. In Section 6, we present the results of experiments conducted using various features and classification models along with CNN. In the last section, we conclude our paper, followed by future work and references.

2 Background and Related work

There have been several studies on computational methods to detect abusive/aggressive language published on social media in the last few years (Razavi et al., 2010) (Watanabe et al., 2018). The first thing to observe is that majority of the work in this domain has been done in English (Del Bosque and Garza, 2014) and a few more languages (Alfina et al.), (Mubarak et al., 2017), (Tarasova, 2016), but we know that social media abuse, bullying or aggression is independent of demography or language. With the advancement of new language keypads and social media websites supporting many new languages brings with itself the negative side of social media to those languages too. Hence, there is a need to address this problem and many others (Singh et al., 2018) for low resourced languages or say informal languages. (Bali et al., 2014) performed analysis of data from Facebook posts generated by English-Hindi bilingual users. Analysis depicted that significant amount of code-mixing was present in the posts. (Vyas et al., 2014) formalized the problem, 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 for Hindi-English code-mixed text that can identify the language of the words, normalize them to their standard forms, assign them their POS tag and segment into chunks.

Script	No. of posts/comments
Roman	10,000
Devnagari	2,000
Total	12,000

Table 1: Text statistics in corpus

Tag	Count
CAG	4869
NAG	2275
OAG	4856

Table 2: Tags and their Count in Corpus

3 Dataset

We used the Hindi-English code-mixed dataset (Kumar et al., 2018) published as a shared task for 1^{st} Workshop on Trolling, aggression and Cyberbullying (**TRAC-1**) [3] . The data was crawled from public Facebook Pages and Twitter. The data was mainly collected from the pages/issues that are expected to be discussed more among the Indians (and in Hindi) for the reason of the presence of Code-Mixed text.

While collecting data from Facebook more than 40 pages were identified and crawled. It included pages of the below-mentioned types:

- News websites/organizations like NDTV, ABP News, Zee News, etc.
- Web-based forums/portals like Firstost, The Logical Indian, etc.
- Political Parties/groups like INC, BJP, etc.
- Students' organisations/groups like SFI, JNUSU, AISA, etc.
- Support and opposition groups built around incidents in last 2 years in Indian Universities of higher education like Rohith Vemula's suicide in HCU, February 9, 2016, incident in JNU, etc.

For Twitter, the data was collected using some of the popular hashtags around such contentious themes as "beef ban", "India vs. Pakistan cricket

[3] https://sites.google.com/view/trac1/shared-task?authuser=0

match", "election results", "opinions on movies", etc. During collection, the data was not sampled on the basis of language and so it included data from English, Hindi as well as some other Indian languages. In the later stages, the data belonging to other languages was removed leaving only Hindi, English and Hindi-English Code-Mixed data.

The collected dataset was labelled into three classes naming:

Covertly-Aggressive (CAG): It refers to texts which are an indirect attack against the victim and is often packaged as (insincere) polite expressions (through the use of conventionalized polite structures), In general, a lot of cases of satire, rhetorical questions, etc. An example is given below -

T2 : *"Harish Om kya anti-national ko bail mil sakti hai? ? ?"*

Translation: *"Harish Om can an anti-national get bail?"*

Overtly-Aggressive (OAG): This refers to the texts in which aggression is overtly expressed either through the use of specific kind of lexical items or lexical features which is considered aggressive and/or certain syntactic structures. An example is given below -

T1 : *"Agar inke bas ki nahi hai toh Hume bhej do border"*

Translation: *"If they can't handle it, then send us to border"*

Non-Aggressive (NAG): It refers to texts which are not lying in the above two categories. An example is given below -

T1 : *"Waise bandhu jet lag se bachne ke liye Raat ko 10 baje ke baad so jao"*

Translation: *"By the way brother, sleep after 10 o'clock at night to avoid jet lag"*

3.1 Aggression and Abuse

Abuses and aggression are often correlated but neither entails the other. In cases of certain pragmatic practices like 'banter' and 'jocular mockery', abusive constructions are used for establishing inter-personal relationships and increasing solidarity. So these instances cannot be labelled as aggressive. Moreover, In this dataset, both use of aggression and abuse is present in the text.

However, both aggression and abuse do co-occur in a lot of cases and a lot of times we are probably more concerned with (actual) abuses (and not the banter/teasing) than aggression itself. As such, we may consider abuse/curse as one aspect of aggression (even though not strictly a subtype of aggression). However, a more in-depth analysis is needed to discover the relationship between the two.

Tag	Average post length
CAG	28.10
NAG	27.40
OAG	27.63

Table 3: Average post length of different class text

Tag	Average word length
CAG	4.24
NAG	4.77
OAG	4.24

Table 4: Average word length in different class text

4 Pre-processing and Data Statistics

4.1 Data Statistics

The format of data provided was the "post/comment ID", "post/comment", "Tag". Where **ID** refers to users who posted the content, **post/comment** refers to the actual text content of the post/comment which we need to process to develop our features on, and **Tags** are the three class labels. It (the data) contained posts/comments both in Roman scripts as well as Devanagari scripts. Table 1 shows the statistics of the data distribution in Roman and Devanagari scripts. Table 2 shows the count of tags in the corpus.

4.2 Pre-Processing

The pre-processing step is done after extracting our useful features from the text as many elements get removed in pre-process step as they are not important for textual feature creation as well helps to keep the dimension of our feature vector small and

Tag	Precision	Recall	F1-score
CAG	0.51	0.72	0.60
NAG	0.98	0.13	0.23
OAG	0.60	0.60	0.60
avg / total	0.64	0.56	0.53

Table 5: Multi-modal Naive Bayes Model

Tag	Precision	Recall	F1-score
CAG	0.49	0.50	0.50
NAG	0.44	0.42	0.43
OAG	0.53	0.53	0.53
avg / total	0.50	0.50	0.50

Table 6: Decision Tree Model

dense. Below mentioned are the steps we did on our text for pre-processing:

- Transliterated Devnagari text to Roman using the system by (Bhat et al., 2014).
- Removed stop words.
- Removed Punctuation.
- Replaced multiple spaces (" ") or "." to a single one.
- Removed URLs.
- Removed emoticon Uni-codes and other unknown Uni-codes from text.
- Removed phone numbers ("+91-...").

5 System architecture and Features

5.1 Convolutional Neural Network

In this section, we outline the Convolutional Neural Networks (Fukushima, 1988) for classification and also provide the process description for text classification in particular. Convolutional Neural Networks are multistage trainable Neural Networks architectures developed for classification tasks (LeCun et al., 1998). Each of these stages, consist the types of layers described below (Georgakopoulos and Plagianakos, 2017):

- **Convolutional Layers:** These are major components of the CNN. A convolutional layer consists of a number of kernel matrices that perform convolution on their input and produce an output matrix of features where a bias value is added. The learning procedures aim to train the kernel weights and biases as shared neuron connection weights.
- **Pooling Layers:** These are the integral components of the CNN. The purpose of a pooling layer is to perform dimensionality reduction of the input feature images. Pooling layers make a sub-sampling to the output of the convolutional layer matrices combing neighbouring elements. The most common pooling function is the max-pooling function, which takes the maximum value of the local neighbourhoods.
- **Embedding Layer:** It is a special component of the CNN for text classification problems. The purpose of an embedding layer is to transform the text inputs into a suitable form for the CNN. Here, each word of a text document is transformed into a dense vector of fixed size.
- **Fully-Connected Layer:** It is a classic Feed-Forward Neural Network (FNN) hidden layer. It can be interpreted as a special case of the convolutional layer with kernel size 1x1. This type of layer belongs to the class of trainable layer weights and it is used in the final stages of CNN.

The training of CNN relies on the Back-Propagation (BP) training algorithm (LeCun et al., 1998). The requirements of the BP algorithm is a vector with input patterns x and a vector with targets y, respectively. The input x_i is associated with the output o_i. Each output is compared to its corresponding desirable target and their difference provides the training error. Our goal is to find weights that minimize the cost function

$$E_w = \frac{1}{n} \sum_{p=1}^{P} \sum_{j=1}^{N_L} (o_{j,p}^{L} - y_{j,p})^2$$

where P is the number of patterns, $o_{j,p}^{L}$ is the output of j^{th} neuron that belongs to L^{th} layer, N_L is the number of neurons in output of L^{th} layer, $y_{j,p}$ is the desirable target of j^{th} neuron of pattern p. To minimize the cost function E_w, a pseudo-stochastic version of SGD algorithm, also called mini-batch Stochastic Gradient Descent (mSGD), is usually utilized (Bottou, 1998).

5.2 LSTMs

As mentioned in (Lample et al., 2016) Recurrent neural networks (RNN) are a family of neural networks that operate on sequential data. They take

Tag	Precision	Recall	F1-score
CAG	0.54	0.68	0.60
NAG	0.70	0.31	0.43
OAG	0.60	0.59	0.59
Avg / total	0.59	0.57	0.56

Table 7: SVM Model with L2 penalty

Tag	Precision	Recall	F1-score
CAG	0.54	0.51	0.51
NAG	0.74	0.79	0.75
OAG	0.52	0.53	0.52
avg / total	0.41	0.42	0.39

Table 8: MLP model

an input sequence of vectors $(x_1, x_2, \ldots, x_n)$ and return another sequence $(h_1, h_2, \ldots, h_n)$ that represents some information about the sequence at every step of the input. In theory, RNNs can learn long dependencies but in practice, they fail to do so and tend to be biased towards the most recent input in the sequence (Bengio et al., 1994). Long Short Term Memory networks or "LSTMs" are a special kind of RNN, capable of learning long-term dependencies. Here with our data where posts/comments are not very long in the size LSTMs can provide us with a better result as keeping previous contexts is one of the specialities of LSTM networks. LSTM networks were first introduced by (Hochreiter and Schmidhuber, 1997) and they were refined and popularized by many other authors. They work well with a large variety of problems especially the one consisting of sequence and are now widely used. They do so using several gates that control the proportion of the input to give to the memory cell, and the proportion from the previous state to forget. These network has been used in the past for tasks similar to our task like hate speech detection (Badjatiya et al., 2017), bullying detection (Agrawal and Awekar, 2018), Abusive language detection (Chu et al., 2016), etc on social media text. Hence, we experiment out data with LSTM model and compare the results as to how good our CNN model works as compares to LSTMs.

5.3 Features

- **Text Based:** In this stretch, we look into the presence of hashtags, uppercase text (indication of intense emotional state or 'shouting'), number of emoticons (emoticons and exclamation marks can be associated with more aggressive forms of online communication (Clarke and Grieve, 2017)), presence and repetition of punctuation, URLs, phone numbers, etc. The median value for URLs for "bully", "spam", "aggressive", and normal users is 1, 1, 0.9, and 0.6, respectively. The maximum number of URLs between users also varies: for the bully and aggressive users it is 1.17 and 2 respectively, while for spam and normal users it is 2.38 and 1.38. Thus, normal users tend to post fewer URLs than others. Also aggressive and bully users have a propensity to use more hashtags within their tweets, as they try to disseminate their attacking message to more individuals or groups (Chatzakou et al., 2017).
- **Abusive or Aggressive words:** We observe that the text with tags as aggressive either Covertly or Overly contains Abusive and Aggressive language usage which can be used as one of the important features to identify the aggressive posts/comments. It's not always though that the aggressive text contains these words but it's a feature which gives some certainty for the presence of Aggressive nature of the text (Chatzakou et al., 2017).
- **Numerical features:** It is observed that the average length of post/comment for aggressive texts is, in general, greater as compared to non-aggressive posts. It is also observed that the average size of words in the aggressive texts are smaller as compared to Non-aggressive posts which deny the findings of (Nobata et al., 2016). The stats for the average length of post/comment and that of words in these three class are shown in Table 3 and 4.

Tag	Precision	Recall	F1-score
CAG	0.63	0.62	0.63
NAG	0.83	0.83	0.83
OAG	0.69	0.69	0.69
avg / total	0.58	0.57	0.58

Table 9: LSTM model

While creating the sentence vectors with the use of vocabulary from out dataset (top 4000 words) we removed sentences which had sizes

Tag	Precision	Recall	F1-score
CAG	0.63	0.63	0.63
NAG	0.83	0.85	0.84
OAG	0.69	0.68	0.69
avg / total	0.57	0.59	0.58

Table 10: CNN model

greater than 400, which is a good threshold looking at the average size of a sentence which is 28. After removing the sentence having size more than 400 we are left with 11,617 sentences and our dimensionality reduced to 11617x400 from 11634x5000 as there were few sentences of 5000 length (noise in social media text). This reduction in dimensionality helps our training model to run faster without affecting the results/learning much.

Tag	Count
CAG	974
NAG	466
OAG	960
Total	2400

Table 11: Support Test instances for each Tags

List of all features that we used for our systems are as follows:

- Sentence vector after pre-processing.
- Count of abusive/aggressive/offensive words.
- Number of tokens.
- Size of post/comment.
- Presence of URLs.
- Presence of phone numbers.
- Presence of hash-tags.
- Number of single letters.
- Average length of words.
- Number of words with uppercase characters.
- Number of Punctuation.

We experimented with the different set of features for the CNN model which we have discussed in Section 6 and a report for which can be seen in Table 13.

Model	Accuracy
Multimodal NB	0.56
Decision Tree	0.49
SVM	0.57
MLP	0.42
LSTM	0.58
CNN	0.73

Table 12: Test Accuracy of different models

6 Experiments

This section presents the experiments we performed with different combinations of features and models. The models on which we ran experiments are:

- Multimodal Naive Bayes
- Decision Tree
- Support Vector Machine (SVM)
- Multi layer Perceptrons (MLPs)
- Long-short Term Memory (LSTM) Networks
- Convolutional Neural Networks (CNNs)

For experiments on the first three models, we used only the text as features and used library feature extraction method which turns our text content into numerical features with bag-of-words strategy, ignoring the relative positions of words. The classification report for these three models has been shown in Table 5, 6, 7 respectively with their accuracy as shown in Table 12. The support for each tag during the experiments on our models shown in Table 5, 6 and 7 have the same numbers of data per tag which is shown in Table 11.

We then experimented with the three above mentioned neural networks and their classification report is shown in Tables 8, 9 and 10.

In order to determine the effect of each feature and parameter of different models, we performed several experiments with some and all feature at a time simultaneously changing the values of the parameters as well. We arrived at the provided values of parameters and hyper-parameters after fine empirical tuning.

7 Results and Observations

The classification report of all the models is shown in Tables 5, 6, 7, 8, 9, 10. From the experiments above we can conclude that CNN works best for our case classifying posts 73.2% of the times to

Feature Eliminated	Accuracy
None	72.8
Size of post	72.4
Avg. length of words	72.6
Single letters/chars count	**73.0**
Number of Tokens	72.2
Presence of URL	**73.2**
Presence of Phone-number	72.9
Total Uppercase words	72.5
Presence of hash-tags	72.3
Number of punctuation's	**73.1**
Aggressive words	72.2
All except sent vector	**73.2**

Table 13: Impact Of Each Feature Calculated By Eliminating One at A Time for CNN Model.

the correct class. The best classification accuracy of all the models is shown in Table 12.

One observation to keep in mind is that the nature of data that we used in our work also makes this classification task difficult to generalize (Davidson et al., 2017), this is because of the presence of noisy text in social media data.

8 Conclusion and Future work

In this paper, we experimented with machine learning as well as deep learning classification models for classifying social media Hindi-English Code-Mixed sentences as aggressive or not. We cannot always rely on neural networks to perform better than simple machine learning algorithms (eg. SVM performs better than MLP). CNN worked best with an accuracy of 73.2% and the best f1-score of 0.58. To make our predictions and models results more significant, we would like to choose a greater variety of social media text that could be considered as offensive/aggressive/hate speech. In addition, many of the posts were from the same thread i.e not much diverse. This has advantages and disadvantages. One advantage may be that this makes the system more fine-tuned: if two people are discussing the same topic, what differentiates one as using "aggressive/hate speech" versus one who is not? But on the other hand, many of the posts were similar in meaning and did not add much to our model to learn. In future, we would like to create a larger, more representative dataset of social media post/comments, perhaps those flagged as offensive by users/annotators as well as covering more diverse and general topic discussions on social media. We also plan to explore some more features from a different variety of texts and experiment them with the deep learning methodologies available in natural language processing. The processed dataset as well as the system models are made available online [4].

References

Sweta Agrawal and Amit Awekar. 2018. Deep learning for detecting cyberbullying across multiple social media platforms. In *European Conference on Information Retrieval*, pages 141–153. Springer.

Ika Alfina, Rio Mulia, Mohamad Ivan Fanany, and Yudo Ekanata. Hate speech detection in the indonesian language: A dataset and preliminary study.

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*, pages 759–760. International World Wide Web Conferences Steering Committee.

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.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2014. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 48–53. ACM.

Léon Bottou. 1998. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. 2017. Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on web science conference*, pages 13–22. ACM.

Theodora Chu, Kylie Jue, and Max Wang. 2016. Comment abuse classification with deep learning. *Von https://web. stanford. edu/class/cs224n/reports/2762092. pdf abgerufen*.

[4] https://github.com/SilentFlame/AggressionDetection

Isobelle Clarke and Jack Grieve. 2017. Dimensions of abusive language on twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 1–10.

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*.

Laura P Del Bosque and Sara Elena Garza. 2014. Aggressive text detection for cyberbullying. In *Mexican International Conference on Artificial Intelligence*, pages 221–232. Springer.

Kunihiko Fukushima. 1988. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130.

Spiros V Georgakopoulos and Vassilis P Plagianakos. 2017. A novel adaptive learning rate algorithm for convolutional neural network training. In *International Conference on Engineering Applications of Neural Networks*, pages 327–336. Springer.

John J Gumperz. 1982. *Discourse strategies*, volume 1. Cambridge University Press.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ritesh Kumar, Aishwarya N Reganti, Akshit Bhatia, and Tushar Maheshwari. 2018. Aggression-annotated corpus of hindi-english code-mixed data. *arXiv preprint arXiv:1803.09402*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

John Lipski. 1978. Code-switching and the problem of bilingual competence. *Aspects of bilingualism*, 250:264.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

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

Whitney Phillips. 2015. *This is why we can't have nice things: Mapping the relationship between online trolling and mainstream culture*. Mit Press.

Amir H Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. Offensive language detection using multi-level classification. In *Canadian Conference on Artificial Intelligence*, pages 16–27. Springer.

Caitlin Elizabeth Ring. 2013. Hate speech in social media: An exploration of the problem and its proposed solutions.

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*.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for hindi-english code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35.

Rekha Sugandhi, Anurag Pande, Siddhant Chawla, Abhishek Agrawal, and Husen Bhagat. 2015. Methods for detection of cyberbullying: A survey. In *Intelligent Systems Design and Applications (ISDA), 2015 15th International Conference on*, pages 173–177. IEEE.

Natalya Tarasova. 2016. Classification of hate tweets and their reasons using svm.

Heidi Vandebosch and Katrien Van Cleemput. 2008. Defining cyberbullying: A qualitative research into the perceptions of youngsters. *CyberPsychology & Behavior*, 11(4):499–503.

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.

Hajime Watanabe, Mondher Bouazizi, and Tomoaki Ohtsuki. 2018. Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE Access*, 6:13825–13835.