Detecting Hateful Content on Social Media

Aaniya Gouse[1*], Afaq Alam Khan[1]

[1]Department of IT, Central University of Kashmir. India.

Abstract

With the growth of hateful content all over the web, detecting hatred has gained utter importance. To combat the nuisance self-regulatory methods are found to be in place but mostly these fail to serve the purpose. In this paper we have addressed the issue by training a supervised classifier that is trained based on semantic features. Just as semantic features work well with other experiments regarding sentiment analysis, this work is outperforms the state-of-the-art methods. Observing the performance of the classifier, we incorporate additional features in order that the performance of the classifier is maximized.

***Keywords**: Social media, Natural Language Processing, Hate Speech, Semantic Features, Supervised Learning.*

Introduction

The term hate speech does not have a clearly defined boundary. It stands practically undefined and its loosely defined boundaries are often crept over by “free speech”. For instance, calling a person a name could cleanly be categorized as being “free speech” but at the same time it could be full of hatred innately. Various aspects of hate are cyber bullying [Hosseinmardi et al (2015)], abuse [Nobata etal (2018)], flaming [Nitin et al (2012)], toxicity [Jigsaw (2018)] etc. Time and again every social media user has faced hatred online in forms varying from threat to abuse and so forth. Hateful language has an immense contribution to users abstaining from using social media altogether. Many platforms implement self-regulatory methods to keep a check on hateful content being propagated on the social media which include a user purposely reporting a particular profile as being offensive or violating certain guidelines. These methods being completely dependent over users’ discretion and their own definitions of hate are under qualified to be banked upon. Therefore, as communication online grows a need for an automated hate detector is ever increasing. Prospectively, our hate speech classifier shall prevent all hate crime while still arising. The goal of this research is to combat genocide, suicide, cyber bullying, trolling, terrorist propaganda etc. Our challenge is to detect hate out of the ulterior faces put on by it in the form of sarcasm, offense or misspellings. The state-of-the-art methods mainly employ statistical features but these do not lead to high accuracy and are error-prone as well. As semantic features in other classification tasks are observed to perform better we decided to make use of the same, making ours a one-of-a-kind hate speech classifier. In addition to semantic features, we shall incorporate lexical, morphological and contextual in order that the classifier turns out to be more intelligent.

Contribution of Author

The research is about designing a multi-class general-purpose classifier that is capable of detecting hateful content on social media. Specifically, our contribution can be stated as follows:

- An efficient method for pre-processing of text which includes correcting misspellings and eradicating slangs.
- A classifier that is capable of classifying hatred out of a given corpus.
- Linguistic analysis of text to reveal syntactic and semantic details of language.
- Evaluation of classifier on datasets of varying sizes and types in order to observe variations in performances.

- Validation and evaluation tests that attest efficient performance of our system.

Related Work

Since hate speech is not properly defined till date and is confused with free speech, different works in the field have addressed the same issue but from various different views. Hosseinmardi et al (2015) took up the task of classifying instagram comments as being of Cyber bullying nature or otherwise using a pre-labeled dataset consisting of images and associated comments. They threw light on the differences that lie between cyber bullying and cyber aggression. Sood et al (2012) identified the profanity from among internet comments as well as pioneered the task of crowdsourcing for the purpose of annotating abusive language. Most of the work in the field of detection of hate speech is based on supervised learning approach. Nevertheless Michael Wiegand et al (2018) presented a lexicon based approach to detection of abuse on social media and termed abused as being an aspect of hate which has explicit offense contained within it. W. Warner et al (2012) focused more on hateful content on social media rather than the abuse that the language contains. Initially they manually annotated a corpus of comments from the internet followed by training a supervised classifier that is capable of classifying comments that direct hatred towards a minority group. Y.Chen (2012) being the pioneer that combined lexical in addition to parser features for detecting offensive language from Youtube using SVM for training and n-grams as features. Thomas Davidson et al (2017) remarked that lexical methods may be efficient to identify hate speech but are inaccurate at the same time, thereby, bringing forth the vitality that the context possesses. They aim at bringing about the subtler of differences that exist between hate and offense. Irene Kwok et al trained a supervised classifier using labeled data from Twitter under labels “racist” and “nonracist.” They deducted that their Bag-of-Words model doesn’t efficiently classify anti-black tweets. Therefore, requires improvement such that bigrams are expected to help perform better.

Learning Methods

Generally speaking, classification means choosing a class to which an item might most likely belong. In general, there are two ways, in which a classification task is performed:

a. **Multi-step classification** involves learning more than one (usually two, called binary classification) classifiers. This type is used when solving problems about multiple categories. Here, we learn individual classifiers for each category termed as “one-versus-rest” [3]. For instance, the classifier learned for the whole document will decide based on the individual topics which of the polarity classifier should be used to classify unseen data [10].

b. **Standard one-step classification** involves learning a classifier for the whole of the task as opposed to a multi-step classification which divides the dataset into sub-datasets and learns a classifier for each. For instance, this will learn straight from the document, oblivious of individual topics. This way is observed to perform better than multi-step classification [10].

I. Machine learning based approach

Broadly, machine learning involves feeding some data to the computer along with some specific examples of the output that the data should produce. Based on this data and output instances, the computer learns a model. Thus given a problem the model based on some data/experience and knowledge performs in a specified way. So far we have discussed the classification mode of learning. Classification is one form of machine learning. As such, machine learning involves three different ways of learning models:

i. Supervised learning
ii. Unsupervised learning
iii. Semi-supervised learning.

i. Supervised learning: Essentially, supervised learning consists of two phases; the learning phase where the classifier is learned using data which is associated with their respective labels; the classification phase is that which involves determining the accuracy of the rules over test data. If found to be accurate it will be used to classify unseen data. Figure 1 illustrates the general idea behind supervised learning [11].

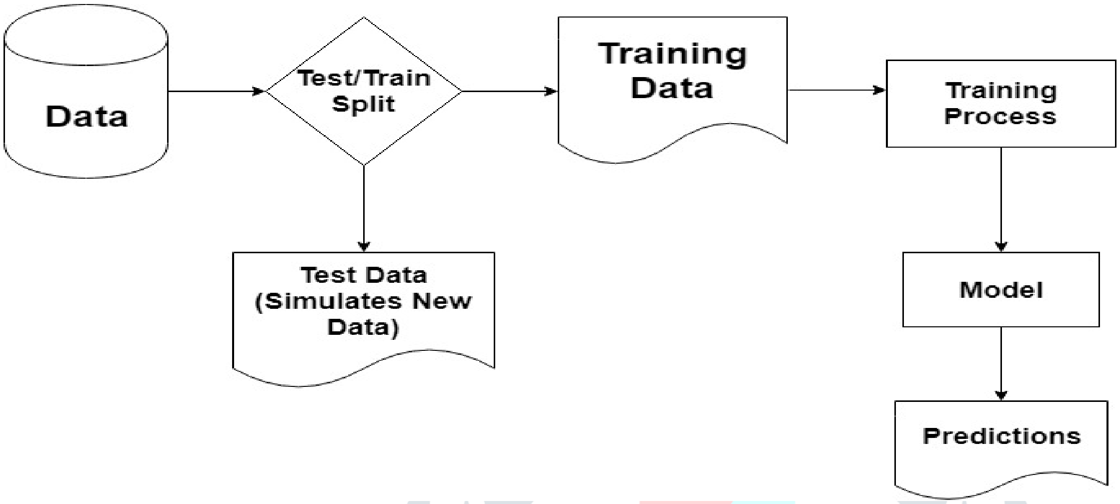

Fig.1. Supervised Learning Approach.

ii. Unsupervised learning: It is used to find patterns within the data when there is no specified output. The learning algorithm is fed with just the input data out of which the similar data is grouped into clusters based on the similarities found within the data [9][5].

iii. Semi-supervised learning: It uses a combination of labeled and unlabeled data to build a classifier. A classifier built for labeled data is used to make the classifier learned for unlabelled data perform better (self-training). Although their work is of unsupervised nature, Njagi Dennis Gitari et al. [1] used labeled data for evaluating their classifier. Bootstrapping involves a classifier from a small quantity of labeled data [5] later used to classify unlabeled data. This is done iteratively till the classifier teaches itself. Mainly, the accuracy of labeled data remains the focus of most research work.

I. **Lexicon based approach**: Building a lexicon is based on two approaches.

i. Dictionary-based approach: This involves a static dictionary of words tagged against a polarity label and a semantic score [8]. It does not contain domain specific knowledge.

ii. Corpus-based approach: It is a data-driven, a rule-based approach which has access to a context in addition to the sentiment labels. It has domain specific knowledge.

Proposed Methodology

An innovative approach towards detection of hateful content on social media is presented. It's one of a kind as it's the pioneer work in hate speech detection on social media which involves employment of semantic features. The novelty of the paper lies in the fact that the classifier is aimed to be a general purpose hate speech classifier and is built using morphological, contextual, semantic as well as lexical features. Based upon the content of the post, the generalized multi-class hate speech classifier classifies the post into any number of classes that are provided to it. The following figure 2 portrays the overall working of our system.

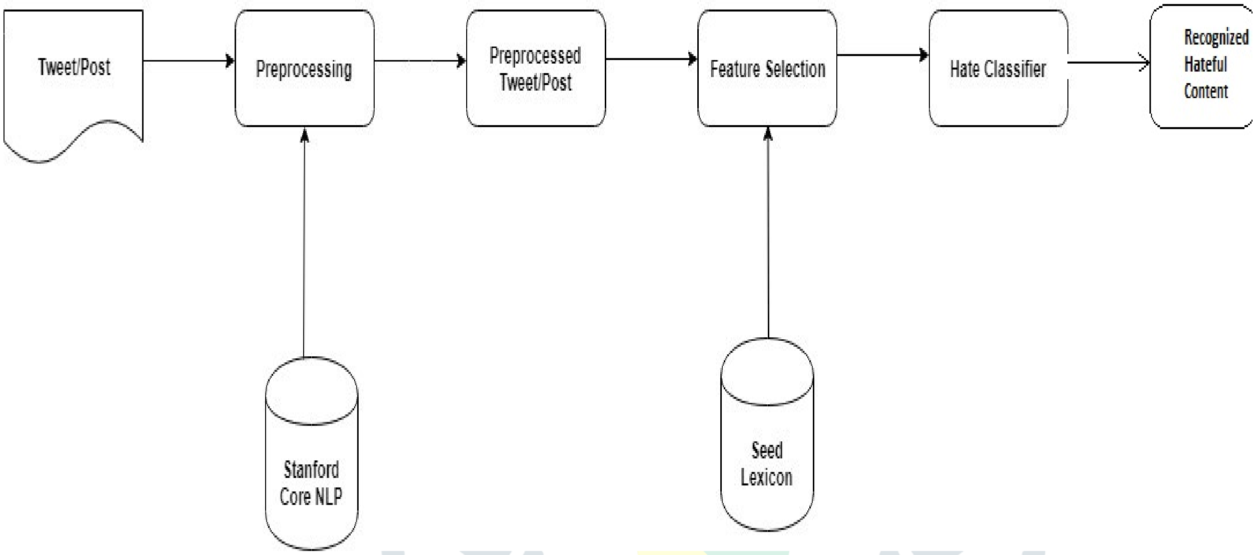

Figure.2. Overall Methodology

Hate Datasets

- Ritesh Kumar et al. (2018) presented a dataset comprising of 15,000 Facebook posts. Each post labeled against labels namely “Covertly Aggressive”, “Overtly Aggressive” or “Non-Aggressive”.
- Another dataset consisting of 16907 instances, out of which 1970, 3378 and 11559 labeled as “Racism”, “Sexism” and “None” respectively is used [11].
- Third dataset comprising of 20001 tweets is also used [13]. The tweets are humanly annotated; those which are “Cyber-aggressive” are labeled as 1 and those which are “Non Cyber-aggressive” are labeled as 0.

Automatic Hate Speech Classification

Based on supervised learning we learned a hate speech classifier. The proposed classification algorithm is carried out with the aid of multi-class Support Vector Machine (SVM). It is composed of the following parts:

a. Hate Dataset[2]
b. Pre-processing
c. Lexicon Generation

d. Feature Selection and Extraction
e. Vectorization
f. Automatic Hate Classifier

Preprocessing Module

Crucially, preprocessing involves cleaning of data, analysis of features, annotation of data, and normalization of data. It involves removing upper case letters and stop words, tokenization, Part Of Speech tagging, stemming, lemmatization etc

a. In order that the tweets/posts are tokenized, Stanford Core NLP is used.
b. Stemming is performed using Porter Stemmer.
c. Stop words are removed.
d. Usernames and URLs are removed.
e. Uppercase letters are removed.
f. Using dictionary module WordNet and replacer module Netlingo, slangs and abbreviations are removed from the post/tweet. The former retrieves proper meanings of words passing onto the latter which replaces words that have no proper meanings.

Figure 3. illustrates preprocessing of posts in detail.

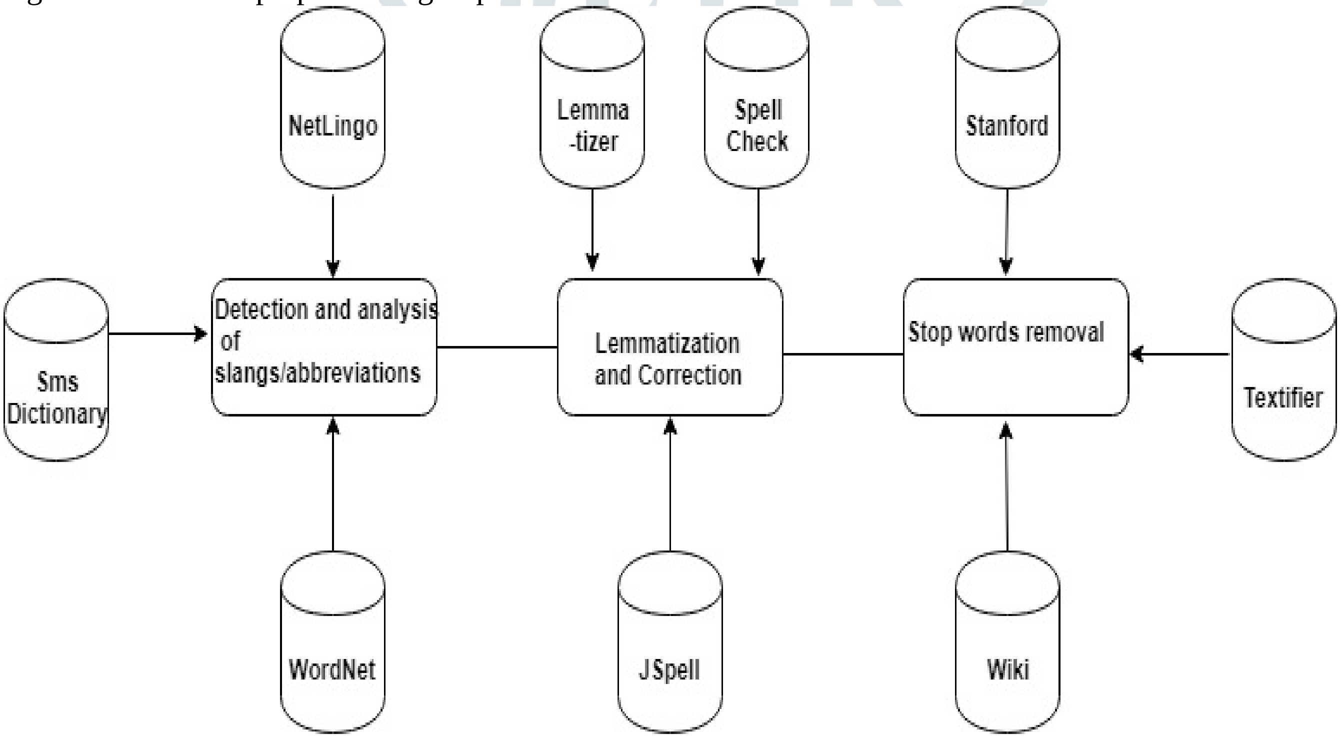

Fig.3. Preprocessing of Posts.

Lexicon Generation

To annotate each sentence of the corpus a seed lexicon from www.hatebase.org was used. In order to keep check of hate speech, hatebase makes use of a wide vocabulary in varied languages pertaining to disability, sexual discrimination, ethnicity, caste etc, throughout nations which amount exceed 200 in number [14]. The initial seed set consisting of 1034 words as obtained from hatebase was then extended with the help of distributional semantic models resulting in 1559 words.

Feature Selection

So that the prediction performed turns out to be optimal features are extracted. The features that we selected for our task are extracted using Java and are listed below:

a. Unigrams: Unit tokens out of a corpus in the form of nouns, verbs, adjectives etc regardless of order.
b. Bigrams: Two random unigrams taken together following a particular order. The word following the seed word is the most probable word that would follow that word in the corpus.
c. Personal Pronouns: The gender specific pronouns add to information pertaining to the context of words.
d. Hate words: Those words which are included in the hate lexicon related to our task.

Vetcorization

Since the features that must be input to the SVM classifier must be in vector representation, we vectorized the feature dictionary by performing indexing on the same. Each feature is associated with an index number which represents it's presence and thereafter those which reoccur are rose in weight. This is a simple method of vectorization which assigns each feature against its index number. These vectors are then fed to the multi-class SVM classifier and it is trained based on these.

Automatic Hate Classifier

We employed multi-class SVM kernel to implement our hate classifier. SVM makes use of the seed words as acquired during data normalization after these are preprocessed i.e. brought into number form. The SVM multi-class formulation as conducted by [15] is used but along with an algorithm that improves the case of linear SVM as its faster. For instance, taking training examples (x_1,y_1) ... (x_n, y_n) are labeled as y_i in [1..k], the following problem of optimization while the process of training is carried out, is solved.

$\min 1/2\ \Sigma_i=1..kw_i*w_i+ C/n\Sigma_i = 1..n\xi i$

$\forall$ y in $[1..k]$: $[x_1\bullet wy_i] >= [x1\bullet wy] + 100*\Delta(y1,y) - \xi 1$

$\forall$ y in $[1..k]$: $[x_n\bullet wy_n] >= [x_n\bullet wy] + 100*\Delta(yn,y) - \xi n$

C is the regularization parameter that trades off margin size and training error. $\Delta(y_n, y)$ is the loss function that returns 0 if y_n equals y, and 1 otherwise. [15]

The hate classifier is given the posts as an input along with the labels. It outputs the posts each labeled against the provided label. Based on the features the hate classifier learns a model with the aid of SVM which then predicts labels pertaining hatred against ambiguous data.

Algorithm

The propounded algorithm of hate classification as suggested by us is presented in the following steps:

a) Choosing a set of seed words vis-à-vis hate speech namely, hatebase.org.
b) Augmentation of the seed lexicon in (a) with the pre trained Word2Vec model. This results in the expanded seed set to get the hate lexicon.
c) Carry out pre-processing of training datasets.
d) Extraction and selection of features for hate detection from the preprocessed dataset.
e) Vectorization of features into numbers so as to be trained using SVM.

f) Assigning weights to each feature using indexing of the feature dictionary such that each feature is represented as “Feature (Number):Weight”.
g) Training the classifier in order to predict hatred.

Evaluation, Validation and Results

The hate classifier is given the dataset [2] as an input. Given the chosen classes namely CAG, NAG and OAG the classifier produces the post along with the label associated with it. We tested our classifier upon the same dataset and reached an accuracy of 68%. The measures that we used to evaluate our hate classifier include confusion matrices, accuracy, f1-score, recall and precision. In case of multi-class classification, confusion matrices prove to be a good measure of efficiency in particular. A general appearance of a confusion matrix is depicted by table 4. While misclassified data is represented by false negatives and false positives, the correctly classified data in the table stands either as true positive or true negative.

Table 4.GeneralStructure of Confusion Matrix

	Class A Prediction	Class B Prediction
Class A Real	True Positive	False Negative
Class B Real	False Positive	True Negative

Precision is the measure of exactness of the model. From among the total number of records that exist in the dataset, it is calculated as the relevant records that we find. Mathematically,

$$Precision = \frac{tp}{tp + fp}$$

Recall is the measure of completeness of the model. From among the total number of relevant records that exist in the dataset, it is calculated as the relevant records that we find. There exists an inverse relation between precision and recall. Mathematically,

$$Recall = \frac{tp}{tp + fn}$$

F-score is given by the harmonic mean of Recall and Precision.

$$F1 - Score = \frac{2(Precision * Recall)}{Precision + Recall}$$

Considering its benefits and efficiency the validation technique that was used is Leave One Out Cross-Validation where training is performed on the entire dataset, but one small part is set aside over which it iterates. It is less biased but leads to higher variance and therefore, overfitting. Figure provides an insight into LOOCV technique [16].

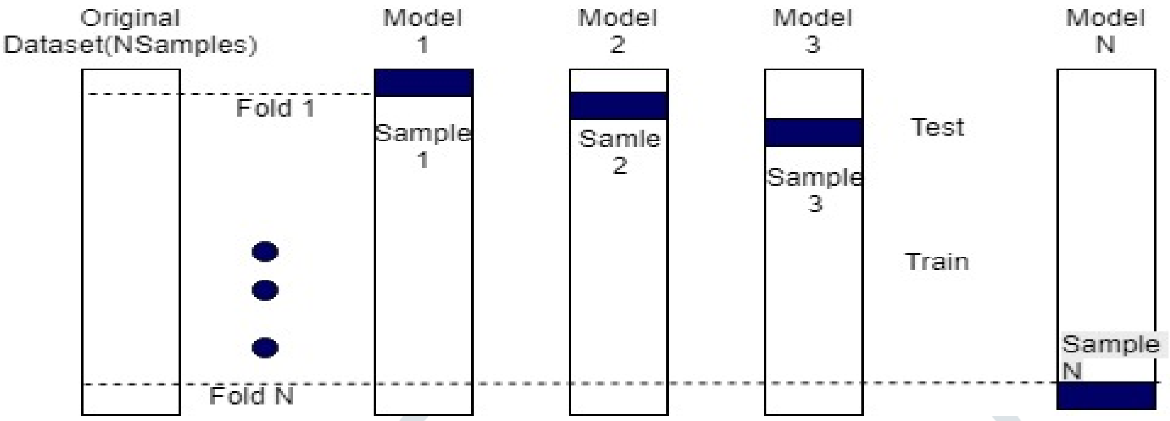

Fig .4. Leave One Out Validation.

Comparative Analysis

We present the outcome that our algorithm produces and contrast it against the like techniques that have been used for hate detection. We achieved an F1-Score of 76.6.

Table 7. Comparison of our hate classifier with similar techniques.

Technique	Accuracy
Proposed Algorithm	68%
Waseem Z, Hovy D. Hateful symbols or hateful people? predictive features for hate speech detection on twitter	73.89(F-measure)
Waseem Z. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter.	53.43(F-measure)

Conclusion

With the incorporation of certain features like unigrams, bigrams, personal pronouns and hate words from the hate lexicon, we reached an accuracy of 68%. We intend to improve accuracy by additionally incorporating semantic features in order to train our supervised classifier. Semantic features are observed to produce optimal results for all tasks related to sentiment analysis and thus, we expect it to work well with hate detection as well.

References

1. Hosseinmardi H, Mattson SA, Rafiq RI, Han R, Lv Q, Mishra S. Detection of cyberbullying incidents on the instagram social network. arXiv preprint arXiv:1503.03909. 2015 Mar 12.
2. Nobata C, Tetreault J, Thomas A, Mehdad Y, Chang Y. Abusive language detection in online user content. InProceedings of the 25th international conference on world wide web 2016 Apr 11 (pp. 145-153). International World Wide Web Conferences Steering Committee.
3. Verma R, Nitin N, Srivastava A. On Behavioural Responses and Different Shades of Flaming in Social Media and Computer Mediated Communication. International Journal of Human Capital and Information Technology Professionals (IJHCITP). 2016 Oct 1;7(4):33-49.
4. Sood SO, Antin J, Churchill E. Using crowdsourcing to improve profanity detection. In2012 AAAI Spring Symposium Series 2012 Mar 23.
5. Warner W, Hirschberg J. Detecting hate speech on the world wide web. InProceedings of the second workshop on language in social media 2012 Jun 7 (pp. 19-26). Association for Computational Linguistics.
6. Chen Y, Zhou Y, Zhu S, Xu H. Detecting offensive language in social media to protect adolescent online safety. In2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing 2012 Sep 3 (pp. 71-80). IEEE.
7. Davidson T, Warmsley D, Macy M, Weber I. Automated hate speech detection and the problem of offensive language. arXiv preprint arXiv:1703.04009. 2017 Mar 11.
8. Kwok I, Wang Y. Locate the hate: Detecting tweets against blacks. InTwenty-seventh AAAI conference on artificial intelligence 2013 Jun 29.
9. Park JH, Fung P. One-step and two-step classification for abusive language detection on twitter. arXiv preprint arXiv:1706.01206. 2017 Jun 5.
10. Valletta JJ, Torney C, Kings M, Thornton A, Madden J. Applications of machine learning in animal behaviour studies. Animal Behaviour. 2017 Feb 1;124:203-20.
11. Gitari ND, Zuping Z, Damien H, Long J. A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering. 2015 Apr;10(4):215-30.
12. Neviarouskaya A, Prendinger H, Ishizuka M. SentiFul: A lexicon for sentiment analysis. IEEE Transactions on Affective Computing. 2011 Jan;2(1):22-36.
13. Kumar R, Reganti AN, Bhatia A, Maheshwari T. Aggression-annotated Corpus of Hindi-English Code-mixed Data. arXiv preprint arXiv:1803.09402. 2018 Mar 26.
14. Waseem Z. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. InProceedings of the first workshop on NLP and computational social science 2016 (pp. 138-142).
15. Gupta, M. (n.d.). [Blog] *Dataturks*. Available at: https://dataturks.com/ [Accessed 8 Mar. 2019].
16. https://hatebase.org. (2019). *How It Works*. [online] Available at: https://hatebase.org/how_it_works [Accessed 14 Mar. 2019].
17. Pak A, Paroubek P. Twitter as a corpus for sentiment analysis and opinion mining. InLREc 2010 May 19 (Vol. 10, No. 2010, pp. 1320-1326).