International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 09 | Sep 2019 www.irjet.net p-ISSN: 2395-0072

A Pragmatic Supervised Learning Methodology of Hate Speech Detection in Social Media

G. Priyadharshini[1]

[1]*Research Scholar, School of Computer Science, Engineering and Applications, Bharathidasan University, Trichy*

Abstract - In recent decades, with the excessive involvement of social networks and social media platforms in day to day life, the digital interaction between its users has become rapid, easy and time convenient. Inspite of its numerous advantages, the anonymity associated with these interactions often leads to the emergence of more hateful, offensive and aggressive communication styles. These intrude at a fast and uncontrollable pace and apparently causes severe damage to the targets, being vital that governments and social networking platforms are successful in detecting and regulating aggressive and hateful behaviors occurring on a regular basis on multiple online platforms. The hate speech detection due to its abstractness considered, its far from being trival. Therefore this paper is proposed to deliver and complement current methodology and solutions on the detection of hate speech online, focusing on social media.

***Key Words*: Text Preprocessing, Feature Extraction, Machine Learning, Classification.**

1. INTRODUCTION

Hate speech is a language that violates people by considering their specific characteristics such as physical appearences, sexual orientation, gender identity, national or ethinic origin, descent, religion or other. These hate language is expressed in different format, styles and styles targeting different groups and minorities which can occur with different linguistic forms such as in subtle forms or even through humour. In order to capture the frequency of hate speech occurring in social media platforms, a large scale systemic measurement study was executed on commonly social sites such as Twitter and Youtube. This paper provides a summarized overview of pragmatic approach of automatic hate speech detection that is in present existence. It would be in need for freshers of NLP research who wanted to keep themselves aware of the actual state of art.

2. TEXT PREPROCESSING TECHNIQUES

In order to maximize the efficiency of the machine learning algorithms used in the classification processes, it is necessary to have clean data. Consequently, there's a set of techniques that can be applied in text mining that reduces the amount of noise in the data that is substantial due to the comments' shortness and informality, usually containing useless or unknown characters, emoticons, among other things and to make the data clean:

2.1 Tokenization

It is defined as slicing a stream of text into pieces, denoted as tokens. The tokenization varies from language to language but lexical characteristics such as colloquialism (e.g. "u" instead of "you"), contractions (e.g. "aren't" instead of "are not") and others (e.g. "O'Neil) make the task harder. Sometimes also removal of less frequent tokens of the data is included.

2.2 Filtering

This involves removal of punctuation marks and irrelevant and/or invalid characters, (e.g. "?|%&!"), removal of stop words that are frequently used words that carry no useful meaning whose commonness and lack of meaning makes them useless. These filtering is very necessary since they do not contribute to the classification task.

2.3 Stemming

It is the process of reducing inflected words to a common base form(e.g. "ponies" turns into "poni" and "cats" into "cat"). Stemming also improves performance by reducing the dimensionality of the data, since the words "fishing", "fished", and "fisher" are treated as the same word "fish".

2.4 SpellChecker

Misspelling is common in online platforms due to their informal nature. A spell checker is needed to avoid unidentified or intentionally camouflaged words (e.g. "niggr", "fck").

2.5 Lemmatization

Although very similar to stemming, lemmatization considers the morphological analysis of the words. While stemming would shorten the words "studies" to "studi" and "studying" to "study", lemmatization would shorten both to "study".

2.6 PoS Tagging

Part of speech tagging, is a technique to extract the part of speech associated with each word of the corpus, grammatically wise which might be common to remove words belonging to certain parts of speech that might end up not being so relevant(e.g. pronouns).

2.7 Lowercasing

Lowercasing is converting a stream of text to lowercase which improves the performance of the classification since it reduces the dimensionality of the data. Not applying this technique may raise problems such as "tomorrow", "TOMORROW" and "ToMoRroW" being considered different words.

3. FEATURE EXTRACTION TECHNIQUES

Feature extraction techniques accumulates derived values (features) from the input data (text in this specific scenario) and generates distinctive properties,usually informative and non-redundant, that paves way to improve the learning and generalization tasks of the machine learning algorithms. On extraction of features there happens a subset of features that will have more relevant information. Some of the frequently used feature extraction approaches is presented here.

3.1 N-Grams

N-grams are one of the most used techniques in hate speech automatic detection and related tasks [1,3,14]. The most common n-grams approach consists in combining sequential words into lists with size N. In this case, the goal is to enumerate all the expressions of size N and count the occurrences of them. This allows to improve the classifiers' performance because it incorporates at some degree the context of each word. Instead of using words it is also possible to use n-grams with characters or syllables. This approach is not so susceptible to spelling variations as when words are used. In a study character n-gram features proved to be more predictive than token n-gram features, for the specific problem of abusive language detection [2].

3.2 Bag of Words (BoW)

Bag of words is a form of representation of words by disregarding grammar and the order of the words in sentences, while keeping the multiplicity. Like in n-grams, BoW can been coded using tfidf, token counter or hashing function. Usually it is typically used to group textual elements as tokens, but it can also group other representations such as parts of speech.

3.3 TF-IDF

Term frequency-inverse document frequency is a numerical statistic that measures the importance and need of a certain word in a data corpus. This helps in understanding the importance of certain words to express specific types of speech (e.g."hate")[29].

3.4 Word Embeddings

It is a representation of text where words that have the same meaning have a similar representation. It is a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values seem to be a neural network. One of the word embedding technique that gained maximum interest by researchers in text mining is Word2vec.

- **Word2Vec:** The granularity of the embedding is word wise, generating a vector for each word of the corpus. There are 2 different possible models: **CBOW** (continuous bag of words), that learns to predict the word by the context, and skip-grams, which is designed to predict the context itself. According to [22], CBOW is faster to train and has slightly better accuracy for the frequent words. On the other hand, **Skip-grams** work well with a small amount of training data and represent well even rare words or sentences. Most of the approaches that used Word2Vec[20] apply the skip-gram model.

3.5 Sentiment Analysis

It is much necessary to understand the sentiment behind the message, or else its actual real meaning will perhaps be misunderstood or misinterpreted. Usually in social media, sentiment analysis approaches tend to focus on identifying the polarity (positive or negative connotation) of comments and sentences as a whole.

3.6 Template based Strategy

The idea behind this strategy is to construct a corpus of words, and for each word in the corpus, collect K words that occurring around. This information can be used as context.

4. CLASSIFICATION ALGORITHMS

Hate speech detection in text is mostly a supervised classification using machine learning algorithms. The usage of Deep learning approaches have increased significantly because of its intense accuracy which caused the emergence of neural networks on large scale for text classification.

4.1 Support Vector Machines

SVM's are widely used in classification problems and the algorithm can be described as an hyperplane that categorizes input data (text in this case).In 2017, SVM's held the best results for text classification tasks, but in 2018 deep learning took over, especially in hate speech detection as described here [24].

4.2 Logistic Regression

logistic regression is a (predictive) regression analysis which estimates the parameters of a logistic model, a statistical model that uses a logistic function to model a binary dependant variable [28].

4.3 Naïve Bayes

This is an algorithm based on the Bayes' theorem with strong naive independence assumptions between the features of the data. It generally assumes that a particular feature in a class is unrelated to any other feature. Naive Bayes is a model useful for large datasets and does well despite being a simple method.

4.4 Random Forest

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [27]. This model requires almost no input preparation, performs implicit feature selection and is very quick to train, performing well overall.

4.5 Decision Tree

This is an algorithm that provides support for decision making, providing a tree-like model of decisions and their possible consequences and other measures (e.g. resource cost, utility). They are often used since their output is usually readable, being simple to understand and interpret by humans. They are also fast and perform well on large datasets, but they are prone to overfiting.

4.6 Gradient Boosting

This is a predicition model consisting of an ensemble of weak prediction models, typically decision trees (that's why it may also be called gradient boosted trees), in which the predictions are not made independently (as in Bagging), but sequentially. The sequential modeling allows for each model to learn from the mistakes made by the previous one[23].

Deep learning popularity has been growing significantly over the recent years, especially in text classification. This is partly due to the disclosure of artificial neural networks' architecture, which made it possible and easier to tune the parameters and, consequently, model the behavior of such algorithms. The main artificial neural networks' architectures are:

4.7 CNN (Convolutional neural networks)

A class of deep feed-forward artifical neural networks. A CNN consists of an input and output layer and multiple hidden layers which consist of convolutional layers, pooling layers and fully connected layers[26].

4.8 RNN (Recurrent neural networks)

Unlike CNN's, are able to handle sequential data, allowing to produce temporal dynamic behaviors according to a time sequence. The connections between nodes form a directed graph. RNN's have feedback loops in the recurrent layer, which act as a memory mechanism. Despite this fact, long-term temporal dependencies are hard to grasp by the standard architecture, because the gradient of the loss function decays exponentially with time (vanishing gradient problem). For this reason, new architectures have been introduced.

- **LSTM : Long short-term memory neural networks:** These are a type of RNN that use special units in addition to standard units, by including a memory cell able to keep information in memory for long periods of time. A set of gates is used to control when information enters the memory, when it's output, and when it's forgotten enabling this architecture to learn longer-term dependencies as detailed in [25] and [26].
- **GRU: Gated recurrent unit neural networks:** These are similar to LSTM's, but their structure is slightly simpler. Although they also use a set of gates to control the flow of information, these are fewer when compared to LSTM's.

RNN supports sequential architectures where CNN has a hierarchical architecture.GRU and CNN results can be compared with respect to text size, GRU is better when the sentences are bit longer. Finally, they concluded that deep neural network performance is highly dependable on tuning the hyperparameters.

5. PERFORMANCE METRICS

For evaluating the performance of machine learning algorithm, the metrics are built from confusion matrix where output can be two or more classes The confusion matrix records which samples of the data have been correctly and incorrectly predicted for each class.

Accuracy is a generic performance measure that assesses the overall effectiveness of the algorithm, by computing the number of correct predictions over all the predictions made. Although it is commonly used accuracy doesn't distinguish between different classes. Consequently, this performance metric may be misleading, especially when the classes of the data are unbalanced.

There is a subset of performance metrics that consider classes. These are usually more useful in sets of data that contain unbalanced classes, since the performance of the algorithm can be assessed class wise. This is quite often in hate speech datasets. The most used class wise, performance measures in hate speech detection are:

Recall (R), also known as Sensitivity or True Positive Rate, is defined as the proportion of real positives that are correctly predicted as positive. **Precision (P)** denotes the proportion of predicted positive cases that area actually positive.

F1 score is defined as the harmonic mean of Precision and Recall, and considers class imbalance, unlike accuracy, hence it's wide usage in hate speech detection.

Using these performance metrics, a graphical visualization of the algorithm's predictions can be computed, known as **ROC (Receiver operating characteristic)**. It shows the relation between the sensitivity and the specificity of the algorithm and is created by plotting the true positive rate (TPR) against the false positive rate (FPR). The higher the TPR, the higher the area under ROC, also known as **AUC (Area under curve)**.

6. RELATED WORK

This section presents a comprehensive review on the key works and existing studies related to the area of automatic detection and hate speech in English Language in particular. In English language, hate speech detection has been intensively investigated by more than 14 contributors in all the categories of hate speech (racial, sexism, religious and general hate).Hate speech in other languages such as Dutch, German, Italian, Turkish, Indonesian, Arabic, Portugese was also investigated but in a limited number. This paper surveys on hate speech detection in English language which has majority researches.

6.1 Dataset and Annotation

One of the difficuties in hate speech detection in text is the availability of dataset. Most of the researches done relied on privately collected datasets. [3] claimed to have collected the largest datasets for abusive language by annotating the comments on Yahoo!. The datasets were again used by [2].But the datasets are not publicly available. Currently, the only publicly accessible abusive speech datasets include those employed in [1,4,14,17,21]. All these publicly available datasets are collected from Twitter by crawling for tweets containing frequently occurring words (based on certain manual analysis) in tweets that contain abusive language references to specific entities.

If a data set needs to be annotated manually, either expert annotators or crowd sourcing services, such as Amazon Mechanical Turk (AMT), are employed. Crowd sourcing services has considerable economic and organizational benifits, especially for a task of time-consuming jobs, but annotation quality might degrade from employing non-expert annotators.

In [14] 16,914 tweets are annotated such that 3,383 as 'sexist', 1,972 as 'racist' and 11,559 as 'neither'. It is then also annotated by crowd-sourcing services over 600 users. This dataset in [14] is further expanded in [21], where some 6,900 more tweets are collected, where about 4,000 are newly introduced to their previous dataset. Then two group of users are involved to annotate the dataset in [21] to create two different versions. The two groups consists of domain experts who are either feminist or anti-racism activist; and amateurs that are crowd-sourced. From the results of annotation it is seen that amateur annotators are more likely to label tweets as hate speech than expert annotators. The authors considering the majority vote, give expert annotations double weight and combine both expert and amateur annotations in the datasets of [17]. In [4], a single dataset is created by merging the dataset in [14] with the expert annotations in [21]. The 24,000 tweets in [1] are annotated into three categories such as 'hate speech', 'offensive language' but not 'hate', and 'neither'. It is always a challenging task to differentiate between hate speech and non hate offensive language because hate speech does not always have offensive words while offensive language does not always express hate. In Previous researches the annotation guidelines provided to their annotators did not serve the purpose to the expected level. Despite providing a definition of hate speech to the annotators, they still fail to produce annotation at an acceptable level of reliability.

6.2 Summary and Analysis

The next following two tables present a summary of all the discussed papers in English language in all the categories of hate speech (racial, sexism, religious and general hate). These tables can serve as a quick reference for all the key works done in the automatic detection in social media. All the approaches and their respective experiments results are listed in a concise manner.

Table -1: Summary of the current state of hate speech detection, and their respective results, in the metric: Precision (P), Recall (R), F1-Score (F)

Author	Year	Platform	Feature Extraction Methods	Classification Algorithms	P	R	F1
[4]	2017	Twitter	Character and word2vec	Hybrid CNN	0.71	0.75	0.73
[5]	2017	Youtube, MySpace, SlashDot	Word embeddings	Fast Text	-	0.76	-

[6]	2018	Twitter, Wikipedia, UseNet	Lexical, Linguistics and Word embeddings	SVM	0.82	0.80	0.81
[7]	2011	Youtube	Tf-idf, lexicon, PoS tag, bigram	SVM	0.66	-	-
[8]	2018	FormSpring	Bag of Words	M-NB and Stochastic Gradient Descent	-	-	0.90
[9]	2018	Twitter	Semantic Context	SVM	0.85	0.84	0.85
[10]	2013	Yahoo News Group	Template-based, PoS tagging	SVM	0.59	0.68	0.63
[11]	2013	Twitter	Unigram	Naïve Bayes	-	-	-
Author	Year	Platform	Feature Extraction Methods	Classification Algorithms	P	R	F1
[12]	2014	Twitter	BOW, Dependencies, Hateful Terms	Bayesian Logistic Regression	0.89	0.69	0.77
[13]	2015	Yahoo Finance	Paragraph2vec and CBOW	Logistic regression	-	-	-
[14]	2016	Twitter	Character ngrams	Logistic regression	0.72	0.77	0.78
[15]	2018	Twitter	Sentiment Based, Semantic Unigram,	J48graft	0.79	0.78	0.78
[16]	2018	Twitter	N-grams, Skipgrams, hierarchical word clusters	RBF kernel SVM	0.78	0.80	0.79
[17]	2017	Twitter	Character Ngrams, word2vec	CNN	0.85	0.72	0.78
[18]	2017	Twitter	Random Embedding,	LSTM and GBDT	0.93	0.93	0.93
[19]	2018	Twitter	Word-based frequency vectorization	RNN and LSTM	0.90	0.87	0.88
[20]	2018	Twitter	Word Embeddings	CNN and GRU	-	-	0.94

7. CONCLUSIONS

In this paper, a systemic literature survey is conducted to in the automatic hate speech detection process. From the previous works, it was found that most of the researchers relied on supervised learning methods in this automatic hate speech detection. Instancing, one major factor is the size of the corpus, as some ML algorithms works pretty well with small datasets and others such as Neural Networks needs more intensive and complex training. Recent researches concentrate largely on deep learning to solve complex learning tasks. Researchers prefer these deep learning approaches

because of their powerful capacity finding data representation for classification. Choosing to adopt deep learning needs commitment in both of preparing and training the model with large amount of data and apparently it has a promising future in the field of automatic detection. Usually, there are two main architectures for deep neural networks that are usually utilized for NLP tasks, these models are: RNN and CNN. From the above survey, there were 5 hate speech researches that adopted deep learning, two of them were RNN and the two others were CNN. These researches showed the effectiveness of both approaches. For that reason, more investigation needs to be done to make the appropriate choice of deep learning architecture.

Henceforth this paper was established with the goal to understand the state of the art by presenting a comprehensive study on the methodology in automatic hate speech detection in social networks

REFERENCES

[1] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. arXiv preprint arXiv:1703.04009, 2017.
[2] Yashar Mehdad and Joel Tetreault. Do characters abuse more than words? In Proceedings of the SIGdial 2016 Conference: The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 299–303, 2016.
[3] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 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, 2016.
[4] J. H. Park and P. Fung, "One-step and Two-step Classification for Abusive Language Detection on Twitter," in AICS Conference, 2017.
[5] H. Chen, S. McKeever, and S. J. Delany, "Abusive text detection using neural networks," in CEUR Workshop Proceedings, 2017, vol. 2086, pp. 258–260.
[6] M. Wiegand, J. Ruppenhofer, A. Schmidt, and C. Greenberg, "Inducing a Lexicon of Abusive Words – a Feature-Based Approach," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 1046–1056.
[7] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of Textual Cyberbullying.," Soc. Mob. Web, vol. 11, no. 02, pp. 11–17, 2011.
[8] R. Pawar, Y. Agrawal, A. Joshi, R. Gorrepati, and R. R. Raje, "Cyberbullying Detection System with Multiple Server Configurations," 2018 IEEE Int. Conf. Electro/Information Technol., pp. 90–95, 2018.
[9] M. Fernandez and H. Alani, "Contextual semantics for radicalisation detection on Twitter," CEUR Workshop Proc., vol. 2182, 2018.
[10] W. Warner and J. Hirschberg, "Detecting Hate Speech on the World Wide Web," no. Lsm, pp. 19–26, 2012.
[11] Kwok and Y. Wang, "Locate the Hate: Detecting Tweets against Blacks," Twenty-Seventh AAAI Conf. Artif. Intell., pp. 1621–1622, 2013.
[12] P. Burnap and M. L. Williams, "Hate Speech, Machine Classification and Statistical Modelling of Information Flows on Twitter: Interpretation and Communication for Policy Decision Making," in Proceedings of the Conference on the Internet, Policy & Politics, 2014, pp. 1–18
[13] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, "Hate Speech Detection with Comment Embeddings," in Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 29–30.
[14] Z. Waseem and D. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," Proc. NAACL Student Res. Work., pp. 88–93, 2016.
[15] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection," IEEE Access, vol. 6, pp. 13825–13835, 2018
[16] S. Malmasi and M. Zampieri, "Challenges in Discriminating Profanity from Hate Speech," J. Exp. Theor. Artif. Intell., vol. 30, pp. 187–202, 2018
[17] B. Gambäck and U. K. Sikdar, "Using Convolutional Neural Networks to Classify Hate-Speech," Assoc. Comput. Linguist., no. 7491, pp. 85–90, 2017.
[18] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep Learning for Hate Speech Detection in Tweets," in Proceedings of the 26th International Conference on World Wide Web Companion, 2017, pp. 759–760
[19] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, "Effective hate-speech detection in Twitter data using recurrent neural networks," Appl. Intell., vol. 48, no. 12, pp. 4730–4742, Dec. 2018.
[20] Z. Zhang and L. Luo, "Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter," vol. 1, no. 0, pp. 1–5, 2018.

[21] ZeerakWaseem.2016. Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In Proceedings of the First Work shop on NLP and Computational Social Science. Association for Computational Linguistics, Austin, Texas, 138–142.

[22] Yoav Goldberg and Omer Levy. word2 vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. Computing Research Repository, abs/1402.3722, 2014.

[23] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. Frontiers in neuro robotics, 7:21, 2013.

[24] Hajime Watanabe, Mondher Bouazizi, and Tomoaki Ohtsuki. Hate speech on twitter a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. IEEE Access, 2 2018. ISSN 2169-3536.

[25] João Guilherme Routarde Sousa, Feature extraction and selection for automatic hate speech detection on Twitter, March 25, 2019

[26] Paula Fortuna, Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes, June 23, 2017

[27] Leo Breiman, Random forests. Machine Learning, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324.

[28] Sandro Sperandei. Understanding logistic regression analysis. Biochemia medica, 24(1):12–18, 2014.

[29] Sanjana Sharma, Saksham Agrawal, and Manish Shrivastava. Degree based classification of harmful speech using twitter data. Computing Research Repository, abs/1806.04197, 2018.