Hate speech detection framework for multi-language textual data

Ajith Murugaian
PGP in Data Science
Praxis Business School
Bangalore, India
ajith.murugaian@praxis.ac.in

James Ajeeth J
PGP in Data Science
Praxis Business School
Bangalore, India
james.ajeeth.j@praxis.ac.in

Harsha Chinnamulagund
PGP in Data Science
Praxis Business School
Bangalore, India
harsha.chinnamulagund@praxis.ac.in

Mohammed Arshan Khan
PGP in Data Science
Praxis Business School
Bangalore, India
mohammad.arshan@praxis.ac.in

Abstract— **Over the past few years, social media has exploded in popularity. 3.81 billion people use social media actively, nearly half of the world population. It shows no signs of slowing down too. Unfortunately, this increase in popularity has been accompanied by an equal amount of increase in hate speech messages. People take advantage of the anonymity of the internet and post racist and offensive content online which they wouldn't have said in real life. Twitter is the social media affected most by this issue. Hence if there was a way to find out which tweets which are offensive or racist and who had posted that content, it would be very useful. We are developing a tool which helps us to filter the offensive tweets and displays only the clean non offensive content. There is also an option to report users if necessary.**

Keywords—text classification, hate speech detection, mining social media, language classification.

I. Introduction

In recent times twitter has become a very unsafe place due to the large amount of offensive content on the site. The tweets could be targeted racism towards a particular community or it could be hateful messages directed at a particular person. In most of the cases, the perpetrators are accounts exclusively dedicated to posting only such content.

The definition of hate speech itself varies from person to person. One person may feel a particular tweet is offensive while the next one may feel it is not. So, to avoid such biasness during labeling, we had to create a general framework of rules to follow while labeling the data.

There has been works done in hate speech classification previously. Some of them have been mentioned in Literature review section. But most of them are done on a single language. We know that, with typing in native languages very easy nowadays, the number of non-English tweets has been rising. So, its important to consider the language of tweet first before we classify them as offensive or not.

Therefore, it becomes a multi-level multi-label classification problem. At the first level, we find out to what language a particular tweet belongs to. Then following that, we find out if the tweet was offensive.

II. Literature Review

A. *Social Media Text Mining*

The first major task is the extraction of tweets from twitter. For this purpose, we used the concepts from Mining the Social Web: Data Mining Facebook, Twitter, Linkedin, Instagram, Github, and More written by Matthew A.Russel and Mikhail Klassen[1]. He mentions two ways of getting tweets – one from a user's timeline and other using hashtags. We had to use both the ways in our work. There had been some changes in Twitter API rate limits from the time of publication of the book and now, so we had to take those into consideration.

B. *Internet slang Conversion*

A lot of acronyms and shortened form of words have become part of the current generation's online slang. So, we could not afford to ignore those words. Around 7000 such slang words and acronyms have been collected and their proper English forms have been prepared in [2]. We used to that to create a slang word dictionary to replace such occurrences in our collection of tweets.

C. *Feature Extraction*

Work done by Koushik, Rajeswari and Muthusamy on hate speech detection [3] showed experimentally term frequency-inverse document frequency approach for feature extraction gives very good results. Our experiments with different feature extractions methods and word embeddings also showed the same.

D. *Usage of N-grams*

A paper published on *Abusive Language Detection in online user content* [4] suggested usage of n-grams to better capture patterns of hateful words. Some words in isolation may not be hateful but when used in tandem with other words they are hateful. We have used word n grams in our work.

III. METHODOLOGY

Fig. 1 gives the architectural overview of our hate speech detection system. The inputs to the system are the timeline tweets. The outputs are the offensive tweets present.

The major processes involved in our work are: (1) mining of tweets from twitter; (2) labelling the tweets as offensive or not offensive; (3) text preprocessing; (4) language detection (5) finding if the tweet was offensive or not.

A. *Extraction of tweets*

We have used two approaches to extract tweets for building our dataset. First approach extracts it through hashtags. The second one extracts it through a particular user's timeline. We had to user both approaches as through approach (1), we won't be able to extract a sufficient number of offensive tweets. If we explicitly search for offensive word hashtags, that will make our model more biased towards particular words. So, for that reason, we find out people who post offensive content frequently and extract tweets from their timeline.

Some of the longer retweets were getting truncated and a '…' was getting added at the end. For that, we had to use a different key *full text* in the Json output. The original tweets didn't face this issue. The user id, user name tweet id, text an hashtag were extracted for each tweet.

In total, around 4000 tweets were extracted.

Since there is a lot of offensive content in the replies section too, effort was made to extract the replies for the tweets. But we ran into some issues, most importantly, twitter allows us to extract only first level replies and even those replies had a hard rate limit. So, we did not proceed with extraction of replies.

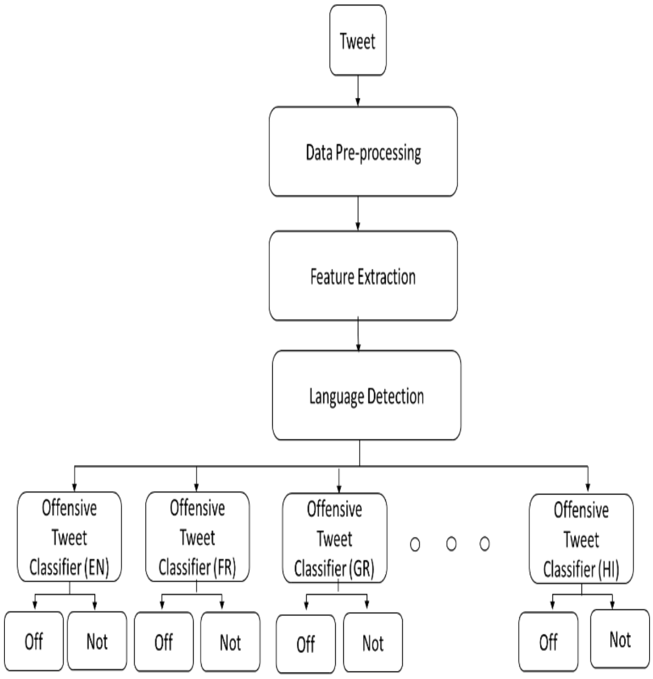
Tweet
Data Pre-processing
Feature Extraction
Language Detection
Offensive Tweet Classifier (EN)
Offensive Tweet Classifier (FR)
Offensive Tweet Classifier (GR)
Offensive Tweet Classifier (HI)
Off
Not
Off
Not
Off
Not
Off
Not

Fig. 1. Hate speech detection framework

B. *Labelling of data*

As mentioned earlier, everyone's definition of offensive or not varies. Therefore, it was important to create a rule set which we could follow to avoid any individual bias. Following were the rules in our labelling framework.

(1) Fuck has become a very commonly used word in online slang. We shouldn't label a tweet as offensive just because it contains the word. We need to look at the word following *fuck* too.

(2) Tweets containing words like bitch, whore, nigger are offensive no matter what the context is.

(3) Sometimes, words in isolation may not be offensive but as a whole it might be offensive. Eg. I hate gays/ I hate black people.

(4) Stop shouldn't be removed. They can change the meaning of a sentence. Eg: You are a monkey – offensive; you have a monkey – non offensive.

(5) Death threats – offensive

(6) Racism towards a particular community – jews, blacks, gays is offensive.

Offensive tweets were labelled as 1 and non-offensive tweets were labelled as 0. Foreign language tweets were labelled as 2. Word clouds were plotted for both type of tweets and analyzed. Results observed have been mentioned in Findings section.

C. *Text preprocessing*

Twitter handles (@) were removed as they did not give any information regarding nature of the tweet. All words were converted into lower case. Images in tweets were getting extracted as https links. So, we used a regex pattern to match and remove the links. Hashtags greater than length 6 were removed. We found out that some people were posting with a # before every word. Eg - #i #am #a #boy. That's why smaller hashtag words were retained and then the hashtag character alone was removed.

There is a lot of internet slang words and shortened version of words in use nowadays. Eg – lol, tis, 2nite, fu etc. We will have to convert these into their normal full verbal form. For that we used [2], which has a collection of 2867 abbreviations and their full texts. We created a dictionary with abbreviations as keys and full texts as values, then replaced the abbreviations present in each tweet in our dataset by its full form. Some of the emoticons were parsable but some of them weren't. So, emoticons were also removed using a regex pattern.

D. *Language Detection*

While for English, we extracted all the tweets through twitter API and built the dataset from scratch, for other languages we used the datasets collected by hatespeechdata.com [5]. The languages we were able to get data on were – Danish, Greek, Arabic, Indonesian, Hindi, French, Turkish. So, our target label in language detection model will be language and it takes 8 different values.

First, we tried with langdetect [6] a package ported from Google's language detection library. Surprisingly we didn't get very good results. On closer investigation, we found that langdetect has 55 languages as its target label values. So, for a restricted 8 language data like ours, it didn't do well.

The Naïve Bayes model we built on the data was performing very well. Some of the languages had a greater number of tweets, so initially results were very skewed towards those languages. After balancing the data, we were able to get rid of the skewed results.

E. *Offensive/Not Offensive Classification*

After the language has been detected, based on the language, the incoming tweets gets sent to the corresponding language model. Most of the cases, the data was imbalanced. We did under sampling to make the data more balanced. For each language, we built a separate model. Feature extraction from text was done using tf-idf with word ngrams. For each language, the best performing model was chosen and added to the pipeline.

F. *Application framework*

We have created implementations at 3 levels – 1) tweet level, 2) user level, 3) hashtag level. Tweet level simply tells us if a particular tweet is offensive or not. User level implementation is useful for finding if a particular person is racist. It goes through the user's timeline, extracts all tweets there and classifies each of them as offensive or not. So depending on the number of offensive tweets posted by him, we have an option to report that user. In the hashtag level, we extract n number of tweets on a particular hashtag, filter the offensive tweets and display only the non-offensive ones. This serves as a kind of hate speech censor.

IV. FINDINGS

In order to find out which groups are targeted the most, we took the nouns in each tweet and did a word cloud to find out the density of each noun. Below are the results for both types of tweet.

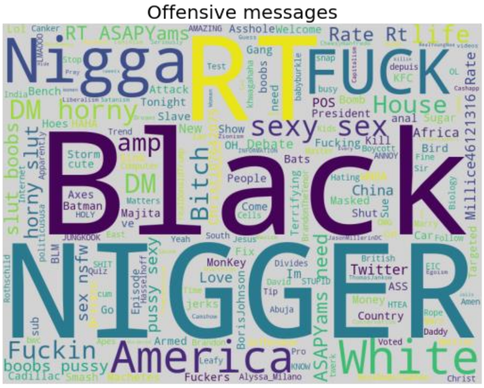

Fig. 2. Word Cloud of Nouns in Offensive tweets

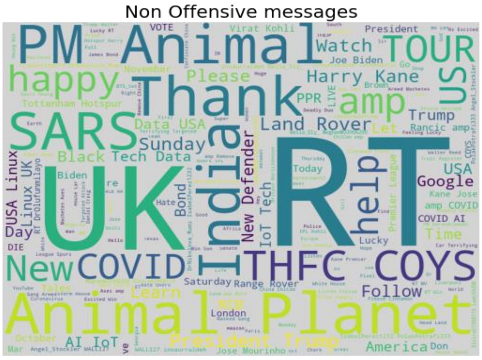

Fig. 3. Word Cloud of Nouns in Non-Offensive tweets

For the language detection model, Google's langdetect delivered an accuracy 69.3% while our Naïve Bayes model got an accuracy of 93.8%. Reason for langdetect's low accuracy – even though languages in our training data were only en, fr, gr, ar, id, tu, da , langdetect predicted 32 different languages.

For feature extraction, use of tf-idf seemed to be the best choice. We experimented with word embeddings like doc2vec, we still got an accuracy of 92.5. So, it didn't have any improvement over tf-idf method.

The best model and its accuracy for each language has been tabulated below.

Language	Model	Validation Accuracy
English	SVC	93.1
French	Random Forest	68.7
Greek	SVC	77.2
Arabic	SVC	74.1
Indo	SVC	89.9
Turkey	SVC	81.5
Danish	Bagging	89.5
Hindi	SVC	70

Table. 1. Languages, best fitting models and correspon

V. CONCLUSION

In this paper, we have proposed a multi-level classification model which helps us detect offensive content in Twitter irrespective of language. In this work, we have considered 8 languages. We aim to add more languages in the future. We have developed an application with a UI which can be used in a variety of ways to help us overcome the problem of hate speech.

VI. REFERENCES

1. Mining the Social Web : Data Mining Facebook, Twitter, Linkedin, Instagram, Github, and More written by Matthew A.Russel and Mikhail Klassen
2. NetLingo: Every Texting Acronym & Online Abbreviation You'll Ever Need to Know https://www.netlingo.com/acronyms.php
3. G. Koushik, K. Rajeswari and S. K. Muthusamy, "Automated Hate Speech Detection on Twitter," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, pp. 1-4, doi: 10.1109/ICCUBEA47591.2019.9128428.
4. C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad and Y. Chang, "Abusive Language Detection in Online User Content", Proceedings of the 25th International Conference on World Wide Web - WWW '16, 2016.
5. Directions in Abusive Language Training Data: Garbage In, Garbage Out https://arxiv.org/abs/2004.01670
6. Language detection library ported from Google's language-detection -https://pypi.org/project/langdetect/