Distant Supervision for Detecting Hate Speech

Diana Petrescu

School of Computer and Communication Sciences
Semester Project

Supervisor
Prof. Karl Aberer
EPFL / LSIR

Supervisor
Hamza Harkous
Rémi Lebret
EPFL / LSIR

June 09, 2017

Contents

1 Abstract

The hate speech has become a major problem affecting the quality of communication through social media. This phenomenon has probably affected Twitter the most among all social networking services. Therefore, the ability to create a knowledge base about hateful tweets could be very helpful in protecting potentially vulnerable users such as children or teenagers.
The goal of our research was precisely to find and implement a scalable methodology to identify hate speech in the tweets based on the hashtags they contained. The search for such hashtags was based on a dictionary of hate words and began with a set of initial hashtags considered themselves as potentially hateful. We then started an iterative process that used the tweets - obtained from Twitter - in order to highlight new hashtags that are more likely to be associated with hateful speech.
The results are promising and we have identified possible improvements for the proposed method.

Keywords *hate speech, tweet, hashtags, social media, supervised learning*

2 Introduction

Cyberbullying has become a common problem nowadays. It is sometimes said to be even more harmful than traditional bullying as it is easier to attack a person and remain anonymous without any physical confrontation. The information spreads more quickly on the Internet than in real life and victims may have lower self-esteem, increased suicidal ideation, and a variety of negative emotional responses. This is a problem especially for teenagers who have less perspective and protection against it [1]. There has always been a lot of controversy concerning hate speech on Twitter. Twitter is considered the worst offender among big media giants in policing hate speech [2]. Hate speech can be defined as speech that attacks, threatens, or insults a person or group on the basis of national origin, ethnicity, color, religion, gender, gender identity, sexual orientation, or disability [3]. Despite recently taken measures, we still find a lot of hate speech on Twitter. Therefore we found this problem interesting and important. Our approach to solving it was detecting hate speech using supervised learning. The main limitation of supervised learning is that we need to access large labeled training data. As social media is noisy and tweets are short, the contexts of the tweets are not always obvious. We can find a lot of spam or sarcasm so it is difficult to get positives samples. In this paper, we propose a scalable methodology for collecting large datasets.

3 Proposed Methodology

Our approach involved the following steps: (1) data collection (with Snowball sampling / Breadth-first search), (2) coupling hashtags with swear terms, (3) cleaning the tweets and the hate words, (4) hate word weighting.

Data Collection. Snowball sampling uses a small pool of initial informants to nominate, through their social networks, other participants who meet the eligibility criteria and could potentially contribute to a specific study. The term "Snowball sampling" reflects an analogy to a snowball increasing in size as it rolls downhill [4]. Therefore to collect our data we have used the same principle with hashtags on Twitter. More precisely, we have made some research to choose several initial hashtags which were likely to be related to hate speech. We then made queries to collect tweets containing those initial hashtags. For each tweet collected, we looked at the other hashtags it contained. We then used those new hashtags to search for the next tweets. We did this several times in order to find more specific hateful hashtags and thus increase the ratio of hate speech among all the tweets containing them. In this way, we could also find hashtags that couldn't have been found by other means because they are too specific.

Coupling hashtags with swear terms. In order to have a heuristic to evaluate the level of hate speech of a tweet and thus determine which hashtag should be kept for the Snowball sampling, we have used a dictionary of hate words (also named hate base) found on the Internet. More precisely, we kept track of the number of hate words occurrences in the collected tweets. Each time we found a hate word in a tweet, we increased this number of occurrences for all the hashtags it contained. We also recorded the number of occurrences of every hashtag in the tweets. At the end, we only kept the hashtags with the highest ratios of hate speech to generate the next samples. We also decided not to keep the hashtags which themselves contained hate words.
It should also be noted that the hashtags which had a high ratio of hate speech but which appeared only in a small number of tweets should be handled separately. For example, a hashtag with ratio 1 could be a hashtag found in only one tweet containing a hate word. We decided to leave those cases aside because they weren't meaningful.

Cleaning the tweets and the hate words. To compare the words of the tweets with the words of the hate base we needed to tokenize the tweets. In order to tokenize a tweet and more specifically the hashtags, we separated the words of that tweet with spaces/punctuation (or group of punctuations). We also chose to separate each upper case letter (or group of upper case letters) followed by lower case letters as different words. We then converted every word to lower case and did the same trick with the words of the hate base. Let's consider for example the following tweet:

Stand! Fight! Win! Founders wrote #2A for self protection. Europe should demand right to bear arms!! #Trump #LondonAttacks #MAGA

which was separated like this:

'stand', '!', 'fight', '!', 'win', '!', 'founders', 'wrote', '#', '2', 'a', 'for', 'self', 'protection', '.', 'europe', 'should', 'demand', 'right', 'to', 'bear', 'arms', '!!', '#', 'trump', '#', 'london', 'attacks', '#', 'maga'

Another example:

I'm an #ExMuslimBecause civilized human beings don't kill innocent people just because they feel offended #MuslimBan

which gave us:

'i', "'", 'm', 'an', '#', 'ex', 'muslim', 'because', 'civilized', 'human', 'beings', 'don', "'", 't', 'kill', 'innocent', 'people', 'just', 'because', 'they', 'feel', 'offended', '#', 'muslim', 'ban'

Hate word weighting. We noticed that the hate words from the hate base or more generally from any dictionary found on the Internet were not always associated with hate speech. Indeed some of them appeared more frequently than others in different contexts but not always with hate speech and therefore they advantaged the tweets containing them. To balance it, we decided to weight the hate words to give the less frequent hate words more importance. Consequently, when hate words which did not appear often were found, they were given a higher weight in order to compete with the hate words appearing more often.

4 Implementation

The methodology proposed in the previous chapter was implemented in Python.

4.1 Data collection

The bottleneck of our approach was collecting the tweets from Twitter. We collected approximately 1000 tweets per hashtag over the last two years. Our initial hashtags were *'trump'*, *'notmypresident'* [5], *theresistance* [6], *'londonAttack'*, *'reclaimtheinternet'* [7], *'muslimban'* [8], *'BlackLivesMatter'* [9]. At first sight, we thought that it was a good idea to implement the Snowball sampling by generating for every hashtag given as input a certain number of new hashtags. However, if we used this strategy the number of tweets requested would increase exponentially. To deal with this, we decided to do the search for the new hashtags over all the tweets found with all the hashtags given as input and to keep the 5 best of all of them to continue our search. In this way, we kept a constant complexity. In order to be sure that we don't miss some good hashtags (that were generated by the initial ones), we repeated the Snowball sampling several times with the same initial seeds but taking as next hashtags the ones that hadn't already been looked at.
We first tried to collect the tweets with the *Twitter API*, *Tweepy* and the *StreamListener* provided in Python. With those libraries, we were able to query tweets containing certain hashtags and to add different parameters to the queries (such as the language of the tweets, the retweeet parameter, etc.). However, due to the restrictions of Twitter and the latency to collect the tweets, this approach wasn't scalable for us. Therefore we decided to use another approach namely data scrapping. We automatically generated the URLs for the specific hashtags and the English

language. We then scraped the tweets using *Selenium (webdriver)* and *BS4* (*BeautifulSoup*). In this way, we got the IDs of the tweets that we needed and could then make the query directly with the IDs (which is a less expensive operation). This approach had also the advantage of being able to query tweets using different chosen time periods. The queries results were given in a JSON format and we only kept the IDs, contents, and hashtags. As we could do the scraping in parallel on several browsers (for different time periods), the latency was significantly reduced. This approach is scalable for collecting thousands of tweets. To give a rough estimate, the collection of thousands of tweets with the API took hours while with the second method it only took several minutes.
It should also be noted that during the iterations for the Snowball sampling, we took care to avoid "cycles". More precisely, if we had already queried tweets for a hashtag we would not do it again even if we found this hashtag from another hashtag afterwards. Moreover, the processing of the tweets was also done in parallel.

4.2 Coupling hashtags with swear terms

For our hate words dictionary, we used the *hatebase.org* words [10]. This site classifies hate words in several categories such as *ethnicity*, *nationality* or *religion*. We used all those filters to create our dictionary. To query the hate base above mentioned we needed to create a special URL containing the filters (i.e. categories) above mentioned and the English language. The response to the query is a web page and we had to read it to get the vocabulary words in a JSON format. We decided to use all the filters combined together in one single hate base to increase the likelihood of finding hate words in the tweets because if we took only one category at the time we couldn't find enough hashtags to continue our Snowball sampling.

4.3 Cleaning the tweets and the hate words

To do the tokenization of the tweets we used the regular expression (the *re* library) and also the existing *word_tokenize* method (*nltk.tokenize* library). We wanted the search of the hate words in the tweets to be fast. Therefore we used dictionaries because the search in a Python dictionary is done in O(1) time. We thus had to do a "normalization" operation. In tweets and compound hate words, we added a '_' separator in front of and after every word. This way we could just use the *in* functionality of the dictionary without needing to use nested loops.

4.4 Hate word weighting

We have tried to use another dictionary (i.e. noswearing words [11]) for our hate base and noticed that the results obtained greatly depended on the choice of the dictionary. The second dictionary gave us worst results so we decided to keep the first one.
Moreover, we noticed that even with our first dictionary (i.e. the hate base) some

of the words appeared more frequently but weren't necessary associated with hate speech. Those words greatly affected the Snowball sampling.
To solve this problem we first tried to delete from our hate base the words which were frequent in common English. However, depending on the case either we could not continue to do the Snowball sampling (as no new hashtags were found with the remaining hate words) or this didn't improve at all the results. We thus decided to keep all the words. Nevertheless, we added to the words with higher frequency a small weight and a greater weight to the ones with a smaller frequency. More precisely, we used two formulas. The first one for the frequency of a hate word f(x):

$$f(x) = \frac{number\ of\ tweets\ containing\ this\ hate\ word}{number\ of\ tweets\ analysed}$$

And the second one for the weight of a hate word g(x) (where f_{max} is the highest frequency among the hate words and f_{min} the smallest one):

$$g(x) = 2^{6 \cdot \frac{f_{max} - f_x}{f_{max} - f_{min}}}$$

We supposed f_{max} always different from f_{min}. The values found for g(x) are thus between 1 and 64. We considered this formula good as there was a big "discrimination" between the values close to f_{min} and the values close to f_{max}. f_{max} has the weight 1 so it is indeed still taken into account.

4.5 Limitations

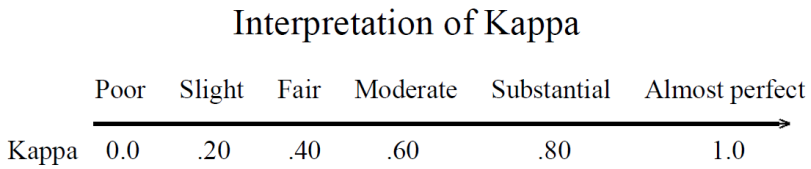
One objective limitation of our method is that after a certain time a "Session exception" occurs while querying tweets for our research. We found no way to determine precisely when this happens but we just catch this exception and continue the execution.
Another limitation is that as Twitter contains a lot of tweets, some of the tweets can be differently written from what we considered.
Furthermore, there is no theoretical proof for the "convergence" of our method. For example, it is not possible to specify some necessary or sufficient conditions that would ensure the best hashtags to be obtained after a certain number of iterations. Therefore we have done an empirical research. The choice of the exponential function used for the weighting of the hate words still needs to be verified on a larger dataset. This distribution should ideally be adjusted given the occurrences of the hate words during the execution (using the standard deviation for example).

5 Evaluation

As mentioned above, we started the sampling with these initial hashtags: *'trump'*, *'notmypresident'*, *theresistance*, *'londonAttack'*, *'reclaimtheinternet'*, *'muslimban'*, *'BlackLivesMatter'*. We did the sampling 3 times from the initial hashtags with 4 iterations each time (an iteration means here finding new hashtags and analyzing them). We kept 2 hashtags found by our algorithm for evaluating our approach. It is worth mentioning that we found that the excessive increase in the number of iterations did not necessarily lead to the discovery of better hashtags. As detailed in the "Conclusions", without taking into account the context of the tweets, there is the risk of seeing the iterative process evolving more and more towards hashtags concerning less hate speech and rather spam or pornographic content.
Hence, two annotators labeled the tweets of 3 hashtags. One initial hashtag (i.e. *'notmypresident'*) and two hashtags that had been obtained with the Snowball sampling (*'triggered'* and *'np'*).
They used 4 categories to label the tweets, namely: *hate speech*, *normal tweet*, *vulgar language but no hate speech*, *spam*.
We used Cohen's kappa coefficient to measure the inter-annotator agreement. The Cohen's kappa function gave us a score that expresses the level of agreement between two annotators on a classification problem. It is defined as follows:

$$kappa = \frac{p_o - p_e}{1 - p_e}$$

where p_o was the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio) and p_e was the expected agreement when both annotators assign labels randomly. p_e was estimated using a per-annotator empirical prior over the class labels [12]. To compute this we used a library in Python, namely *sklearn.metrics*. The Cohen's kappa score can be interpreted as follows:



Kappa	Agreement
< 0	Less than chance agreement
0.01–0.20	Slight agreement
0.21– 0.40	Fair agreement
0.41–0.60	Moderate agreement
0.61–0.80	Substantial agreement
0.81–0.99	Almost perfect agreement

Figure 1: Cohen's kappa score interpretation [13]

Results

The results of the labeling can be seen in the two following tables.

Annotators / Categories	Annotator 1	Annotator 2
Hate speech	0.06	0.06
Normal	0.62	0.74
Vulgar	0.22	0.15
Spam	0.10	0.05

Table 1: *'notmypresident'* hashtag

We chose to label the hashtag *'notmypresident'* as it had obtained the best ratio from the initial hashtags. The kappa's score was 0.44.

The *'np'* hashtag (*now playing*) discovered with Snowball sampling was clearly a *spam/advertisement* hashtag. Both annotators agreed with that. The reason why this hashtag had a good ratio was that the tweets for it contained almost every time either *'prod'*, *'crow'* or *'bitch'* which are words in our hate base.

Annotators / Categories	Annotator 1	Annotator 2
Hate speech	0.06	0.17
Normal	0.90	0.74
Vulgar	0.04	0.09
Spam	0.00	0.00

Table 2: *'triggered'* hashtag

For the second hashtag found (i.e. *'triggered'*), we noticed a small improvement of the percentage of hate speech (compared to the initial one, especially if we look to the second annotation). The ratio obtained for *'triggered'* by our program was 0.11. The kappa's score was quite low, indeed it was 0.28. This could be explained by the fact that it was difficult to label the tweets because they could be considered as ironical most of the time and it was difficult to put them in the right context. Another remark that can be done is that as we hadn't kept any photos or videos in the tweets, the contexts of the tweets were even less obvious. Moreover, some of the tweets were in fact replies to others and were too short to know what they were referencing. In the next chapter, we will give more details concerning possible ways to improve the quality of the obtained hashtags.

6 Conclusion and Future Work

Our work's objective was to obtain new hashtags contained in hateful tweets from a set of initial hashtags. To achieve the task, we proposed a methodology which was then implemented using the Python language. In certain steps of the methodology,

we had to test several variants before choosing the most appropriate. As for the implementation, we paid particular attention to performance (for example, testing two ways to collect tweets, computing in parallel, reduce the complexity of our algorithm, etc.). Generally speaking, there is still room for improvement but as we started our research from scratch, the work we carried out allowed us to highlight some important aspects for the next stage of this research.
The initial set of hashtags was acquired using social engineering in order to choose hashtags about controversial subjects (with a high probability for hateful tweets). A tweet was considered hateful if it contained at least one word from our list of hateful words. This list was obtained automatically from a web site specially used for this kind of work.
It is worth mentioning that some of those words may also appear in non-hateful contexts. For example, "queen" and "queer" are typically widely used words in tweets without necessarily having a negative connotation. Thus, we should not disadvantage words that appear globally less often in tweets, but more likely in a hateful context. Therefore, we tried several "discrimination" functions. However, this aspect should be further investigated. It could be done by taking into account the distribution of the hate words frequencies in the analyzed tweets for example. Improving the dictionary of hateful words can make the selection of good hashtags even more relevant. For example, a word like "bitch" should not be included in such a dictionary because of it's general use.
The degree of hate associated with a hashtag was measured by the ratio between the number of hateful tweets and the total number of tweets - both containing the said hashtag. This approach proved to be appropriate. Indeed, among the hashtags our program highlighted, were "good" hashtags. A hashtag is considered "good" if the majority of the tweets mentioning it can be considered hateful by a human - after reading the said tweets.
Despite all the precautions taken in our research, we have nonetheless observed that the hate ratios of these "good" hashtags sometimes were inferior to other "less good" hashtags. As a matter of fact, our Snowball sampling left good hashtags aside and chose non-hateful tweets - and hashtags - often pornographic or offensive tweets, spams or advertisement. This can be explained by the fact that we do not take into account the context of the tweets nor the users who publish them. The particularities of the users or the tweets (similarity between tweets for the same hashtags, number of retweet, number of followers of a user, user who rarely publishes, user who always publishes the same messages / hashtags, total number of tweets published by a user, etc.) could help us eliminate spam and tweets sent by trolls or manage to "steer" our Snowball sampling in the desired direction.
We think that the ratio mentioned above is a good indicator in order to qualify the degree of hate speech of the hashtags. However, if we also take into account other parameters of the tweets or if we select our dictionary differently we could probably get better results.
In the future, other criteria should complete the approach in order to qualify the treatment of hashtags. Moreover, we could try to repeat the same process for other initial hashtags concerning other domains and compare the results. Furthermore, all the fetched tweets in our study were in English, so one of the things that can still

be done is to use the same algorithm and fetch tweets in another language with an appropriate dictionary for the language in question. Finally, after having found a large dataset, the next step would be to label some tweets using crowd sourcing (for example *CrowdFlower*) and then train a model in order to recognize hate speech.

7 Acknowledgments

During my semester work, I learned a lot about a new topic that I find very interesting. I am very grateful for having had the opportunity to be supervised by experienced researchers in machine learning as well as to improve my knowledge (the initiation to the "really" research, the discovery of the Python language and its libraries, all the new tools used during this project, etc.). Therefore, I would like to thank Professor Karl Aberer, the head of the Distributed Information Systems Laboratory, without whom this project would never have been realized. I am grateful to Messrs. Hamza Harkous and Rémi Lebret for leading my work with professionalism and patience. Their competence and their availability have helped me enormously.

References

[1] https://en.wikipedia.org/wiki/Cyberbullying.

[2] http://forward.com/news/352133/twitter-gives-online-hate-speech-its-biggest-platform-why/.

[3] http://www.dictionary.com/browse/hate-speech.

[4] https://en.wikipedia.org/wiki/Snowball_sampling.

[5] http://www.nydailynews.com/news/election/social-media-reacts-president-trump-viral-hashtags-article-1.2866369.

[6] https://www.yahoo.com/news/one-hashtag-uniting-americans-fight-020252553.html.

[7] http://www.telegraph.co.uk/technology/2017/05/22/mps-targeted-thousands-abusive-tweets-study-shows/.

[8] http://www.uexpress.com/on-religion/2017/2/8/complex-facts-beyond-that-muslim-ban.

[9] http://www.pewinternet.org/2016/08/15/the-hashtag-blacklivesmatter-emerges-social-activism-on-twitter/.

[10] https://www.hatebase.org/.

[11] http://www.noswearing.com/dictionary.

[12] http://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html.

[13] https://stackoverflow.com/questions/11528150/inter-rater-agreement-in-python-cohens-kappa.