Hate Speech Detection

Omar El Malak

Università degli studi di Milano omar.elmalak@studenti.unimi.it

Abstract. The abstract should briefly summarize the contents of the paper in 15–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

The proposed technique applies traditional machine learning techniques after the tweets are pre-processed and described as bag-of-words (BoW).

Given a set of datasets available online the technique classifies the hate speech with an F1-score of 88%.

2 Research question and methodology

The goal of this project is to define a proper technique to classify tweets as hateful or not. Is a challenging text classification problem since the meaning of sentences change with the context and can be considered hate speech to some and not to others based on their definition. Thus the annotated dataset is the key to produce good results.

There are two reasons to approach this problem. First, to reduce the possible harm that is inflicted to individuals and groups. This harm has been shown to have long-term consequences on the well-being of the individual. Second, the goal is to be able to remove automatically illegal content as many countries legislate against hate speech.

Another goal is to extract what is the most relevant terminology for the hateful category.

3 Experimental Results

3.1 Dataset retrieval

Choice of the dataset To build a system able to detect and classify the hate speech is needed an appropriate training set with which training it.

Vidgen et. al. reviews [3] a collection of online available datasets. The most visible distinction between the analysed datasets is the granularity of the annotations. The majority of the available datasets has a binary classification (e.g.

hate / not) even thought are least useful than the multi-class classification (e.g. racist, sexist and not-hateful).

With the intent to approach a multi-class classification problem, for the project it was chosen the Waseem et al. [5] dataset. In particular, to have more sample, the dataset has been joined with another one [4] provided by the same author.

Merging the dataset The merged dataset is composed of two columns, respectively the TweetID and annotation. Duplicates were coming from the two datasets so the first occurrence is kept while any other was deleted.

The second dataset had for each sample a set of annotations, due to the goal of the associated problem. In particular, for each tweet, there are at least four annotations from distinct annotators and the labels belong to the set {neither, racism, sexism, both}. For each entry the majority was taken from the set of annotations, and mapped them in the following way:

- neither → none
- racism → racism
- sexism → sexism
- both → chosen annotation between racism and sexism based on the sentence.

Using the TwitterAPI we obtained the text of only 12873 tweetIDs from the initial set of 19639 entries.

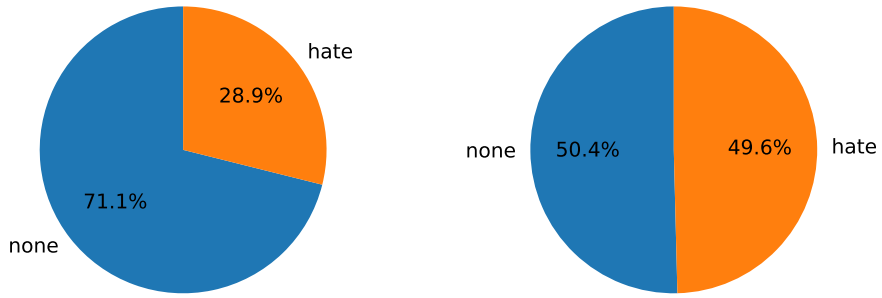
3.2 Dataset analysis

The data obtained is heavily unbalanced with about the 71% of the tweets labelled as "none", while the "racism" and "sexism" tweets are about 28% and 1% respectively. Since the tweets annotated as sexism are not enough represented, from a multi-class text classification problem has shifted to a binary text classification problem where the labels are hate and none.

As described in the following section, to balance between the hate and none samples, from a third dataset available online [1] the tweets annotated as hateful were taken and added to the dataset. With this addition, the dataset is balanced like shown in Fig. 1b.

Unlike the dataset made available by Waseem [4, 5] that scraped the data following the tweets about a tv show, the data provided by Golbeck et. al. [1] try to represent all the possible harassment content available online, searching through dedicated block list and by keywords.

The following two word-clouds show the most frequent words subdivided by labels. As shown, the tweets labelled as 'hate' (Fig. 2b) are mostly represented by hateful and harassing terms. It is to be considered the presence of words like "Jew", "woman", "Muslim" and "black" that don't have a negative meaning but used the describe negatively a group of people. Instead, the tweets labelled not harassing (Fig. 2a) are mainly described by neutral words and some positive and few negative terms.



(a) Unbalanced dataset (b) Balanced dataset

Fig. 1: Balance of the dataset

(a) Not hate (b) Hate

Fig. 2: WordCloud of tweets divided by label.

3.3 The pre-processing phase

Since there are only two classes, the labels are binary encoded.
For each tweet is applied the following set of operation to clean the text:

1. Lowercase the text.
2. Small spell correction. In the English language words usually have a maximum of 2 repeated characters (e.g. "Hello"). This operation removes any extra repetition (e.g. "Heeeellooo" $\rightarrow$ "Heelloo").
3. Find and mapping contracted form, typically used during informal speaking, with the corresponding long form (e.g "she'll" $\rightarrow$ "she will").
4. Remove the presence of links and URLs.
5. Remove any occurrence of numbers.
6. Translate the emojis to the textual description.
7. Tokenize the tweet with a Twitter-aware tokenizer.

8. Discard the tokens that have length lower than three characters.
9. Remove any occurrence of punctuations. This is useful to consider as same any tokens that would be distinguished by a symbol.
   An example is "@bradpitt" → "bradpitt" or "#dummy" → "dummy".
10. Remove stopwords. This operation was done taking a standard list of stop words and adding custom words that were too frequent in the dataset.
11. Stemming of the tokens. As stemmer the technique uses the Snowball algorithm [2].

It was considered to remove any occurrence of user tag inside the tweets but the classification results were poorer. This could be a characteristic of the dataset chosen or possible bias of the annotators.

Once all the data samples have been cleaned, they are used to compute the feature vectors used as input to the classification algorithm. The feature vectors are computed counting, for each sample, the occurrence of each word in the vocabulary. This technique doesn't provide any semantic or relational information but works well with the used dataset.

3.4 Classification

For the classification task, a Logistic Regression classifier was used. The results were validated with the k-Fold cross-validation technique with $k = 5$ and before splitting all the data was shuffled.

The following tables and plots describe the results obtained with the unbalanced dataset and the balanced dataset.

Table 1: Unbalanced dataset scores.

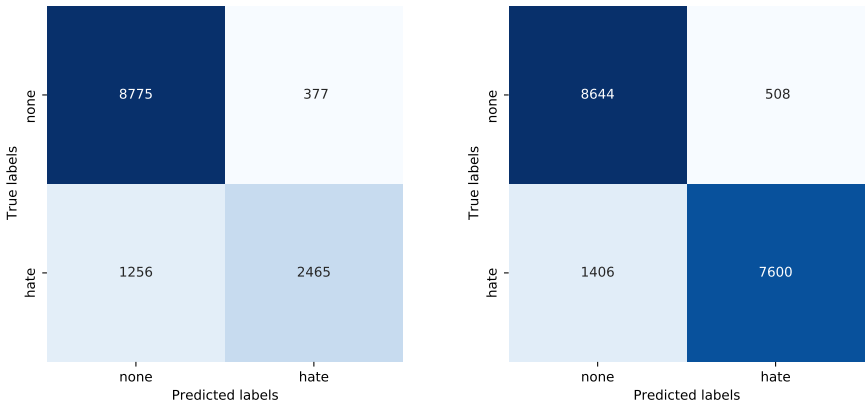
Label	**Precision**	**Recall**	**F1 score**	**Support**
None	0.875	0.959	0.915	9152
Hate	0.867	0.662	0.751	3721

Table 2: Balanced dataset scores.

Label	**Precision**	**Recall**	**F1 score**	**Support**
None	0.860	0.944	0.90	9152
Hate	0.937	0.844	0.888	9006

As shown in Table 1, while the scores for the "none" class are pretty good, the performances are slightly worse with the "hate" class. The reasons are two; first, the class is not well represented and indeed the results in Table 2 shows

improvements in the performance. Second, as shown in the confusion matrix (Fig. 3a) there are a lot of false positive.The Fig. 4 represent the most frequent words that occur in the tweets classified as "not hate" but labelled as "hate". The majority of those words have a neutral sentiment, while others have a positive meaning like "good", "pretty", "better".



(a) Unbalanced data (b) Balanced data

Fig. 3: Confusion matrix

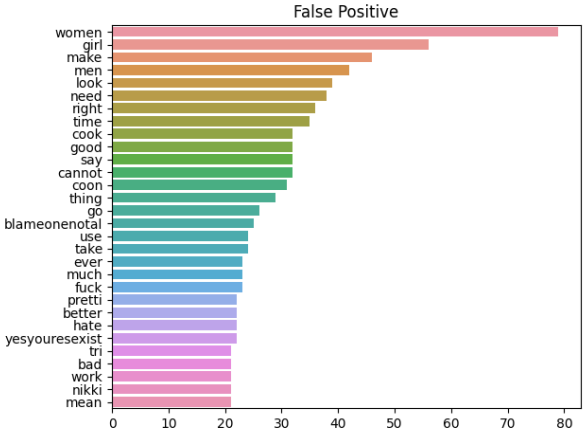
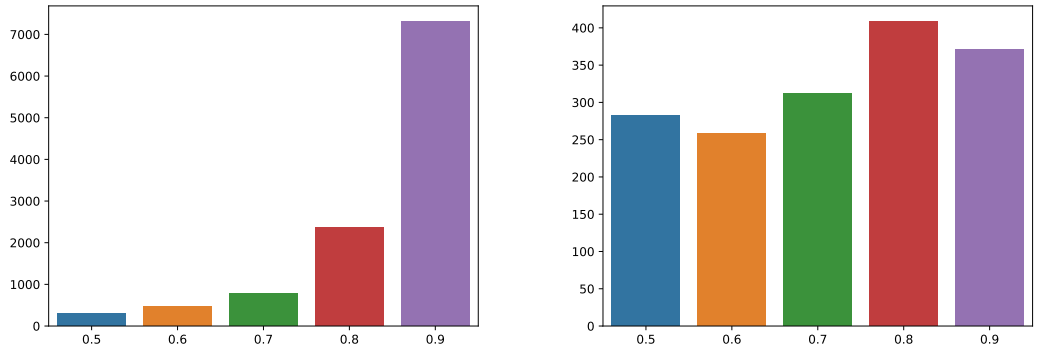


Fig. 4: Most frequent words in false positive tweets.

Once the dataset is balanced, all scores about the hate label have improved toward an F1-score of 88%. Also, the Recall which was the lowest score in the unbalanced dataset, from a value of 62% now is 84%, meaning there are less false positive samples in proportion to the total number of samples.

3.5 Confidence

The bottleneck of this problem is the availability of a large, complete and annotated dataset. Studying the confidence that the model has during the classification the tweets can help understand if this problem could be approached with semi-supervised learning techniques like the self-learning or, in case of a crowdsourcing system, active learning. The classifier, for each classification that makes, returns a probability distribution, one probability for each label. The predicted label is the one with the highest probability.



(a) Right prediction

(b) Wrong prediction

Fig. 5: Confidence

The Fig. 5a shows that the model classify correctly and with high confidence the majority of the tweets. This is a result that work well with the concept of self-learning. Self-learning is a semi-supervised technique in which the model, starting with a small dataset, increases the data size adding samples that the model classified. This technique has the disadvantage that errors made during the classification are propagated on the dataset, and Fig. 5b shows that sometimes, even if the probability if very high, the model make mistakes.

4 Concluding remarks

References

1. Golbeck, J., Gnanasekaran, R., Gunasekaran, R., Hoffman, K., Hottle, J., Jienjitlert, V., Khare, S., Lau, R., Martindale, M., Naik, S., Nixon, H., Ashktorab, Z., Ramachandran, P., Rogers, K., Rogers, L., Sarin, M., Shahane, G., Thanki, J., Vengataraman, P., Gergory, Q.: A large labeled corpus for online harassment research. pp. 229–233 (06 2017). https://doi.org/10.1145/3091478.3091509
2. Porter, M.F.: Snowball: A language for stemming algorithms (2001)
3. Vidgen, B., Derczynski, L.: Directions in abusive language training data: Garbage in, garbage out. arXiv preprint arXiv:2004.01670 (2020)
4. Waseem, Z.: Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In: Proceedings of the First Workshop on NLP and Computational Social Science. pp. 138–142. Association for Computational Linguistics, Austin, Texas (November 2016), http://aclweb.org/anthology/W16-5618
5. Waseem, Z., Hovy, D.: Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In: Proceedings of the NAACL Student Research Workshop. pp. 88–93. Association for Computational Linguistics, San Diego, California (Jun 2016). https://doi.org/10.18653/v1/N16-2013, https://www.aclweb.org/anthology/N16-2013