TDDE16 Project - Classification of hate speech and offensive language on Twitter

Filip Cornell, filco306

December 29, 2018

Abstract

This article covers the topic of tweet classification, with the intent of classifying different tweets into either being offensive, using offensive language or not being offensive at all. For this, a convolutional neural network is applied.

1 Introduction

The internet's ability to permit everyone to speak their mind and share their knowledge opens up tremendous possibility. The amount of information possible to consume has never been greater for the ordinary person, and it has had a great impact on everyone's lives since it came to be. Social medias such as Facebook, Twitter and Instagram allow each and everyone to live a social life not only in real life, but also online. The internet is thus for most people an enriching experience, and some might even argue that it is the greatest invention of all time [2].

However, not everything on the internet can be considered completely positive. The possibility to speak freely also opens up for the less charming sides of the internet; harassment, hate speech and cyberbullying. This is a rising problem, and the possibility to control this by manually checking harassing posts is not only becoming increasingly difficult, but rather practically infeasible. As of November 2018, more than 500,000 posts were posted each minute on Facebook, yielding a massive amount of information [1]. If only 1 % of these were to be reported as offensive, this would lead to Facebook's employees having to manually check more than 2.6 billion posts each year - a huge cost. This has been a increasingly pressing issue, and the German government even threatened Facebook to fine € 50 million annually if they would not take serious measures tackle the issue of removing hateful and offensive posts [10]. Manual reviews' inefficiency is not only problematic in terms of resources, but also speed, as the posts are left online until removed, causing the damage it was intended for.

Detecting and identifying offensive and hateful posts on social media in a quick, automatic and efficient way is thus a pressing issue, and a lot of work has been done previously within the area. In this paper, we investigate how some models previously tried can be combined in a new way and see what results this might bring. The intent is to use a Convolutional Neural Network (CNN) together with the `word2vec` embedding and compare this to the performance of models in previous work. Although several papers before has used CNNs together with `word2vec` to classify tweets, few seem to so far have used `word2vec` as word embeddings (although it has been done for sentences before [7]), but rather GloVe [3, 4, 5]. The paper starts with an introduction to the theory of `word2vec` and CNNs, followed by what previously has been done on tweet classification. This is followed by the method explained, and a more thorough explanation of the specific model used in this task. The results are then presented, followed by a discussion and conclusion.

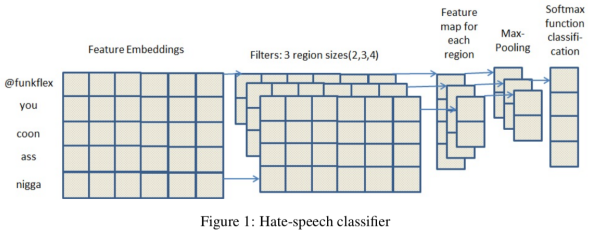

Figure 1: Hate-speech classifier

Figure 1: The layout of the CNN as proposed by Gambäck [5]

.

2 Theory

In this section, the theory behind the models and tools used are presented, and previous work within the field.

2.1 Word2vec

In order to train a neural network to be used for natural language processing, the text must be mapped from text to vectors using a word embedding [8]. In the `word2vec`-package, there exists two types of embeddings; continuous bag-of-words and skip-gram. The continuous bag-of-words

2.2 Convolutional Neural Network (CNN)

A popular approach when analyzing and classifying text data is to use a CNN, originally mainly used in image and sound recognition [3].

When applying a Convolutional Neural Network, the input size of the features has to be fixed. The whole sentence is thus inputted as a 2D-matrix into the neural network, with the words as rows, and their features as columns. In image classification, the squares sent in usually represent images, or compressed such. [3] A CNN usually consists of an input layer, where each node convolutes

When classifying into k classes in a neural network, a popular approach is to have the SoftMax function as activation function to classify into the different classes. The SoftMax function can be written as

$$softmax(z)[i] = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \forall k \in K \qquad (1)$$

where z_i denotes the output of the second last layer, and k denotes class K.

2.3 Related work

Many different approaches has been tried in the field of identifying and classifying hateful speech on the internet before. The first research papers on the topic were published as early as 1997 [9], in which a tree-based classifier managed to reach an accuracy of 88.2 %. The data set was based 720 manually added notes on web page posts from then popular web pages. [9] Over the years, there has been a clear trend of classifying posts on the most popular social medias, such as Twitter and Facebook. In 2016, Waseem and Hovy [11] apply logistic regression combined with character n-gram, achieving an F1-score of 74 %. They are also able to extract the most indicative n-gram features in the set, indicating that features inside "muslim" or "islam" are the most indicative features for racism, while *"bitc"*, *"sex"* or *"xist"* indicate sexual harrassment.

On the same dataset as Waseem and Hovy [11], Gambäck et al. [5] applies a Convolutional Neural Network (CNN) to a set of tweets. To obtain the feature embeddings for the model, `word2vec` and character n-grams were used, while word embeddings were obtained by using `word2vec` and random word vectors. In the neural network, a SoftMax layer calculated the class probability. In between, a pooling layer is utilized to convert each tweet into a vector of fixed length. This pooling layer is followed by a max-pooling layer, used to capture the most important latent semantic factors in each tweet. The model resulted in a precision, at best, of 86 % using random vectors, and an F-score of 78.29 %, obtained through using character n-grams.

A. Gaydhani et al. (2018) [6] uses an n-gram and TF-IDF approach, with n ranging between 1 to 3. TF-IDF is used to be able to reduce the negative

influence of tokens that are not really informative and appear very frequently. Each n-gram feature is then weighted in proportion to their TFIDF value. A final result showed results of an accuracy up to 95 % with logistic regression, 93.4 % for Naïve Bayes and 90.1 % for SVMs after tunings of hyperparameters.

Badjatiya P. et al. [4] investigates the possibility to apply three different neural network architectures on classifying tweets; CNN, Long Short-Term Memory (LSTM) and FastText. On these, word embeddings with GloVe- or random beddings are applied. [4] On these, gradient boosting (GBDT) is also applied as learning method. The best result is achieved with LSTM, random embeddings and GBDT, achieving an F-score of 93 %. The GBDT makes a significance difference, seeing that the F-score of LSTM and random embeddings achieves an F-score of 84.8 %.

3 Data

To make this comparable to some previous work done, the same public datasets are used. With this, we can compare this implementation to the performance of logistic regression, SVM:s and a Naïve Bayes implementations [6]. The first two datasets are fetched from `data.world`, where the tweets has been determined as offensive, hate speech or not offensive by different users.[1] [2] The second dataset[3] is a dataset used in three previous works [6, 5, 11], where the data is classified as hateful or not, but rather divided into categories

Investigating the data further, the two data sets have a skewedness towards neither offensive nor hateful and offensive tweets, as seen in Table 1.

The first dataset contains 20 columns. However, only 2 are significantly interesting for the analysis.

- `does_this_contain_hate_speech` is classified into "The tweet uses offensive language but not hate speech", "The tweet contains hate speech"

[1] `https://data.world/crowdflower/hate-speech-identification`

[2] `https://data.world/ml-research/automated-hate-speech-detection-data`

[3] Available at `https://github.com/ZeerakW/hatespeech`

Table 1: Distribution of types of tweets in dataset1

Class/Dataset	**1**	**2**	**3**	$\sum$
Normal	7274	4163	8233	19670
Offensive	4836	19190	0	24026
Hate speech	2399	1430	2922	6751
Total	14509	24783	11155	50447

- `tweet_text` contains the actual tweet.

The other columns display information such as the confidence of the labelling, based on the judgements from the manual labellers who has labelled the data, as some might have labelled some tweets differently. These would be interesting to incorporate in future work, but is left in this paper for simplification. Instead, the column `does_this_contain_hate_speech`

The second dataset contains only 7 columns, with two columns used for actual analysis.

- `class` - the class as judged by majority of users.
- `tweet` - a string containing the actual tweet to be analyzed.

The data has been labelled by users on CrowdFlower, who has made judgments on whether the tweets are offensive, contains hate speech or neither. Other columns includes information such as number of votes on each class, tweet id and such.

The third dataset only contains two columns; `class` and `id`, where class refers to "racism" or "sexism", indicating hate speech, or "neither?, indicating a normal tweet. Just like Gaydhani A. et al [6], we consider both "racism" and "sexism" to be hatespeech, and "neither" to be a normal tweet. It is important to note that not all tweets from this dataset were possible to be fetched. This was due to two reasons. First of all, some tweets did not seem to remain on Twitter, as they were not fetched. Secondly, some tweets were labelled as both "neither" and either "sexism" or racism in the dataset, making it impractical to use these as neither. These were thus dropped, and out of the 16907 original tweets, only 11155 were retrieved and used. For example,

out of the 1970 tweets marked as racist, only 12 remained fetchable. It is probable that these have been removed due to its racist nature.

3.1 Preprocessing

First, the datasets are merged. The following preprocessings are then made (in given order).

1. All characters are set to lowercase.
2. All usernames in the tweets are changed to "U" to mark a username. A capital U is chosen to ensure that user is recognized that another user is mentioned. This is from the author's personal belief that other users might be mentioned in offensive tweets, but not the ones containing hate speech necessarily, and it thus might have useful information.
3. All escaped characters, such as "&gt" and "&lt" are removed.
4. All url:s are removed.
5. The words are stemmed using the Porter Stemmer from the package `nltk`.
6. Some common words are changed, such as "im" to "i'm", "lil" to "little" and so on.
7. All stopwords are removed.
8. All non alpha-numeric character are removed.

4 Method

To briefly describe the method, each tweet is converted to be represented as a matrix in numerical form using the word embedding `word2vec`. If a tweet is shorter than the most amount of words found in the set after preprocessing, padding is added to have the same dimensions of every tweet. The data is then split up using different seeds (a total of 10 seeds will be used) and then CNN is trained using 80 % of the data in X epochs, and is then tested on the remaining 20 % (the test set). This is repeated on a number of seeds. The metrics used to measure the performance are then calculated based on the confusion matrix retrieved from the test sets, and the average of each metrics will be reported.

4.1 Word embedding

The word embedding `word2vec` is applied to the whole dataset. We use a dimension of 150 on the word vectors and set a minimum count of 2 for a word to be included; in other words, if a word does not exist at least twice, we will not include it as a vector, and it will be considered non-informative and the corresponding vector will be set to a 0-vector. This will yield loss of information, and one might argue that `char2vec` might be useful for tweets as the number of characters is limited in a tweet to 140 characters (although it is discussed whether it is about to be doubled soon [**?**]).

4.2 CNN

The CNN used in this paper is quite simple. As input, we will have a concatenation of the

4.3 Critique of method

One must be critical to the method used.

Finally, it is worth mentioning that initially, the third dataset was not planned to be used. This was later added due to poor results, with the belief that the skewedness of the data and low amount of hate speech was the cause of this. Adding this dataset more than doubled the amount of hatespeech tweets, allowing more data to be used.

4.4 Metrics

To make this comparable to previous works [5, 4, 6], the same metrics used in those will be used here. That is, the precision, recall and the F1-score, three measures commonly used in classification problems. As we have three classes here, it should be clarified that the measures are calculated as follows.

$$P_k = \frac{M_{ii}}{\sum_j M_{j,i}} \tag{2}$$

$$R_k = \frac{M_{ii}}{\sum_j M_{i,j}} \tag{3}$$

$$F1_k = \frac{2 \cdot P_k \cdot R_k}{P_k + R_k} \tag{4}$$

The average scores are then calculated by multiplying the classes' scores with their corresponding percentual representation in the test set.

5 Results

The confusion matrix can be seen in table 3. Table 2

Table 2: Explanation of classes

Type of tweet	**Class**
Hate speech	0
Offensive	1
Neither	2

Table 3: Confusion matrix of result on test set.

Class	**0**	**1**	**2**
0	**0.811**	0.052	0.065
1	0.092	**0.865**	0.137
2	0.097	0.082	**0.800**

6 Discussion

The overall accuracy achieved is comparable to previous works, although it does not seem to match state-of-the-art. The TF-IDF approach

As we can see in the accuracy matrix, few hateful or offensive tweets are classified as normal, but rather as offensive or hateful. However, the distinguishing between hateful and offensive seems a difficult task, and an accuracy of only 60 % is achieved for tweets containing hate speech. This might be due to a few reasons. First of all, the data set is unbalanced, with significantly fewer data points for hate speech tweets than the other two classes. The fact that twitter has removed a lot fo the tweets for the third data set, in particular all except for 12 out of the $\tilde{3}000$ racist tweets does definitely affect the result, and makes it harder to compare with. Secondly, the manual labelling cannot be considered completely accurate. Some loss of accuracy might also be caused by the preprocessing stages causing loss of semantics. The preprocessing is also a bit different from previous works.

7 Conclusion

8 References

References

[1] Facebook statistics. `https://zephoria.com/top-15-valuable-facebook-statistics/`. Accessed: 2018-12-28.

[2] List of the greatest inventions of all time. `http://shortsleeveandtieclub.com/the-top-10-inventions-of-all-time/`. Accessed: 2018-12-28.

[3] Hate speech detection using natural language processing techniques. 2018.

[4] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. *CoRR*, abs/1706.00188, 2017.

[5] Björn Gambäck and Utpal Kumar Sikdar. Using convolutional neural networks to classify hate-speech. 2017.

[6] Aditya Gaydhani, Vikrant Doma, Shrikant Kendre, and Laxmi Bhagwat. Detecting hate speech and offensive language on twitter using machine learning: An n-gram and TFIDF based approach. *CoRR*, abs/1809.08651, 2018.

[7] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics, 2014.

[8] Marco Kuhlmann. Lecture, text mining course tdde16, 2018.

[9] Ellen Spertus. Smokey: Automatic recognition of hostile messages. pages 1058–1065, July 1997.

[10] Emma Thomasson. German cabinet agrees to fine social media over hate speech. https://uk.reuters.com/article/uk-germany-hatecrime-facebook/german-cabinet-agrees-to-fine-social-media-over-hate-speech-idUKKBN1771FK, April 5, 2017. Accessed: 2018-12-28.

[11] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *SRW@HLT-NAACL*, 2016.