University of Ljubljana
Faculty of Computer and
Information Science

**FRI Natural language processing course 2021**
**Project report**

# Offensive language exploratory analysis

Tomaž Martinčič, Dimitar Stefanov and Žiga Trojer

**Abstract**

Offensive speech is often present on the Internet. Many times this is not desirable on the pages, or it is forbidden, so we want to detect it automatically. This task has attracted a lot of attention recently, and as a result, many language processing techniques and algorithms have been developed. In this report, we present the capabilities of a number of them. Furthermore, as our end goal we try to produce a meaningful visualization of the distances among the different classes, or better say types of hate speech.

**Keywords**
hate speech, offensive language, exploratory analysis

*Advisors: prof. dr. Slavko Žitnik*

## Introduction

Our first task was to find as many offensive speech data sets as possible. Out of the 65 datasets for which we were given descriptions and references, we decided to focus on the 25 English language datasets. We had a lot of problems with the protection of personal data - some data sets were not accessible. Many data sets consisted of Twitter messages, but only tweet IDs were given - scrape data is required to analyze these tweets.

Another problem that we encountered was that we found some data sets and most of them had only binary class - hate speech or not. We also found some others, but we can not merge those data sets, because the classes are different. In addition, the datasets were obtained from different media, so formatting issues need to be taken into account in some cases as well.

We have documented our whole analysis of the quality and suitability of the datasets in the Github repository[1]. Additionally, we have come across one useful dataset on Kaggle (Twitter), but it has not been added to Github due to its size (6 GB). This dataset is rather abundant, so we used it for one of our initial explorations.

We also found some methods that we could use. We will list some of them, so the decision which one to choose will be easier.

1. Non-contextual pre-trained word embeddings: Word2Vec, Glove, fastText
2. Contextual pre-trained word embeddings: BERT, ELMo
3. Support Vector Machines (SVM) over Bag-of-Words vectors
4. Recurrent Neural Networks with Long Short-term Memories [1]
5. Convolutional Neural Networks (CNN) [2]
6. Logistic regression (LR), and multi-layer perceptrons (MLP) - using bag-of-words representations based on either word or character-level n-grams [3]
7. Lexical Syntactic Feature (LSF), support vector machine with different kernels: Polynomial, Dot, Radial, ANOVA and Epachnenikov. [4]
8. Ensemble models (combination of context-aware logistic regression and context-aware neural networks) [5]

Our plan is that we start with some simple approaches such as TF-IDF (Term Frequency - Inverse Document Frequency) and BoW (Bag of Words). Then we will try to implement some of the other approaches suggested in the project description (Word2Vec, BERT,...), that best fit our problem.

For the visualization part, we plan on performing PCA (Principal Component Analysis), MDS (Multi-Dimensional Scaling) or representation of the embeddings with t-SNE.

[1] https://github.com/tm1897/nlp_offensive_lang

### Twitter dataset

We already tested [2] some basic methods on one of the datasets. The dataset contains 149,823 tweets, which are categorized into 5 categories of offensive language. First, we had to clean the tweets by removing URLs, and user tags. After that, we encoded tweets by the TD-IDF method. So each document was represented with a vector. We then performed K-Means clustering with a different number of clusters (3, 5, 10, 15) and checked the most frequent words in each cluster. With even that simple methods we were able to notice that most of the clusters do make sense.

We noticed that the current stemmer and stop words list don't work well with this kind of text, as the tweets contain a lot of grammatically incorrect words and language variations, such as slang.

### Reddit dataset

We also tested the TF-IDF method on a Reddit dataset containing 5000 conversations (each conversation has approximately 4 replies). We first preprocessed the conversations by removing tags and unnecessary characters. Then, we considered a corpus of English stopwords, so that we don't have to search for their TF-IDF value. Lastly, we performed the TF-IDF vectorization. By looking at the 50 most frequent words in this dataset, we can see that words representing offensive language are present there. Unfortunately, we came across the same problem as above: we still track a lot of common everyday words, short forms and slang which don't help in our analysis.

In the future, we will combine all of the datasets, and explore the main characteristics and interrelationships of the following types of hate speech:

- Abusive
- Hateful
- Spam
- Harassment
- Personal attack
- Racist
- Sexist
- Homophobe
- Religion
- Other hate

### Word2Vec Model

We utilized the above described Twitter dataset to train our own *Word2Vec* model on it. The embedding size of the words we chose was 100, and during the training we considered a window of size 5. In terms of clustering, we tried to divide the words into 5 categories: homophobe, sexist, racist, religion and hateful, as these classes were also considered in the dataset itself. By having a dataset specific to hate speech recognition, we thought a reasonable Word2Vec model could be trained. Unfortunately, it turned out we needed additional text, additional tweets to capture deeper patterns. Still, as it can be observed in Figure 2, we obtain well defined clusters for each of the terms, although there remains one mixed cluster.

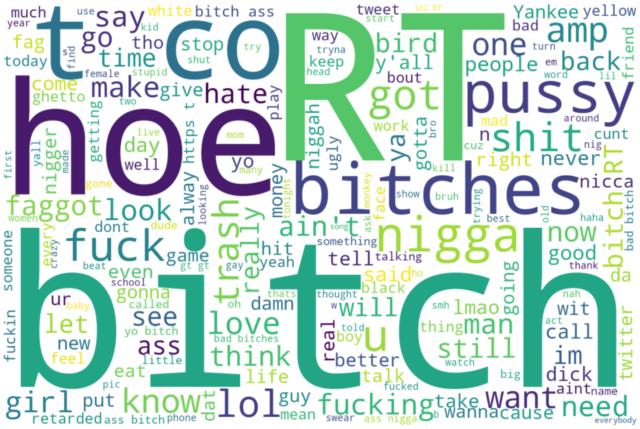

**Figure 1.** Word cloud from tweets with offensive language.

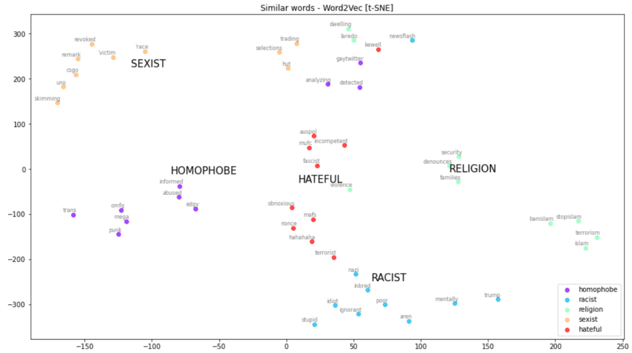

**Figure 2.** A Word2Vec model was trained on a hate speech dataset from Twitter. The obtained word clusters after utilizing t-SNE for dimensionality reduction are presented in this figure.

## References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[2] 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, Doha, Qatar, October 2014. Association for Computational Linguistics.

[3] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1391–1399, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[4] Uwe Bretschneider and Ralf Peters. Detecting cyberbullying in online communities. 06 2016.

[2] These initial analyses have been added to the *develop* branch of our repository.

[5] Lei Gao and Ruihong Huang. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 260–266, Varna, Bulgaria, September 2017. INCOMA Ltd.