

Heidelberg University
Institute of Computer Science
Database Systems Research Group

Report for the lecture Text Analytics
Hate Speech Detection

<https://github.com/fidsusj/HateSpeechDetection>

Mentor: John Ziegler

Team Member: Christopher Klammt, 3588474,
Applied Computer Science
iv249@stud.uni-heidelberg.de

Team Member: Felix Hausberger, 3661293,
Applied Computer Science
eb260@stud.uni-heidelberg.de

Team Member: Nils Krehl, 3664130,
Applied Computer Science
pu268@stud.uni-heidelberg.de

Abstract

Plagiarism statement

We certify that this report is our own work, based on our personal study and/or research and that we have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. We also certify that this report has not previously been submitted for assessment in any other unit, except where specific permission has been granted from all unit coordinators involved, or at any other time in this unit, and that we have not copied in part or whole or otherwise plagiarised the work of other students and/or persons.

Contents

1	Introduction	3
2	Related Work	4
3	Approach	4
3.1	Definition of Hate Speech	4
3.2	Dataset	4
3.3	Preprocessing	4
3.4	Definition of feature groups	4
3.5	Classifiers	4
3.6	Evaluation	4
4	Experimental setup and results	4
4.1	Data	4
4.1.1	Data preparation	5
4.1.2	Corpus building	5
4.1.3	Data analysis	6
5	Analysis	8
6	Conclusion	8

1 Introduction

One current research area in the field of text analytics is hate speech detection. Many approaches from the recent past take use of neural network architectures to deal with such classification problems. One of the most common approaches is to use uni- and bidirectional long short-term memory (LSTM) networks, a recurrent neural network architecture that can process input of arbitrary length and remembers context information [5, 11, 10]. The paper [6] states, that even a simple gated recurrent unit (GRU) architecture can perform as good as more complex units. [1] repurposes the famous bidirectional encoder representations from transformers (BERT) language model to perform classification tasks for hate speech detection. Besides that, other approaches use convolutional neural networks (CNNs) to extract typical hate speech patterns [2, 9, 7] or even deep belief network algorithms [4]. Using neural network approaches means to automatically learn representative features for the classification task. On the other hand, the papers introduced in section 2 use a different approach by solving the classification task with manually extracted features. Nevertheless, none of the papers combines the different achievements of such recent research and compares it to a baseline neural network architecture, which is what this work is dedicated to.

After a definition of the term “hate speech” in section 3 different classifiers will be trained on a holistic, hand-crafted feature set based on recent publications in the field of hate speech detection. The task includes building and preprocessing a training corpus as well as introducing and explaining the different kinds of features. How well the different classifiers perform compared to a neural network approach as a baseline and several statistical insights into typical hate speech artifacts will be presented in section 4. The results of this work in section 5 should show which features work best for which classifier and which problems can be addressed with conventional machine learning methods and which not as opposed to neural network approaches. A summary over the achievements earned will be drawn in section 6.

2 Related Work

3 Approach

3.1 Definition of Hate Speech

3.2 Dataset

3.3 Preprocessing

3.4 Definition of feature groups

3.5 Classifiers

3.6 Evaluation

4 Experimental setup and results

4.1 Data

There are two data sets used for the project. The first one uses data from the *Twitter API* [12]¹. It consists of a sample of around 25k tweets that were identified as hate speech based on a previously composed hate speech lexicon without regarding context information. Subsequently, each document in the corpus got labeled with one of the three categories *hate speech*, *offensive language* or *neutral*. Therefore the data set follows a classical ternary classification style. The workers were instructed to follow predefined definitions of each category and to take context information into consideration. Each tweet was assessed and labeled by three or more workers. The majority of tweets were classified as offensive language (76% at 2/3, 53% at 3/3), only 5% were coded as hate speech. The data is provided offline as a CSV or pickle file.

The second data set uses data from the *White Supremacy Forum* [8]². One document represents a sentence that is either labeled as hate or not hate. In total, 1.119 sentences containing hate and 8.537 sentences being non-hate are provided. Once again the documents were labeled manually by human actors following previously specified guidelines, on request additional context information was provided. The documents are given offline as normal text files with annotations stored in a separate CSV file.

¹<https://github.com/t-davidson/hate-speech-and-offensive-language>

²<https://github.com/Vicomtech/hate-speech-dataset>

4.1.1 Data preparation

To prepare a central dataset, both single datasets had to be transformed into a common format. For the central dataset only the class and the text content of each tweet respectively each forum contribution was considered.

The first dataset *Automated Hate Speech Detection and the Problem of Offensive Language* was entirely given as a .csv file and contains 25.297 tweets, that were either labeled as hate speech, offensive language or neither of both. To determine the right label three independent evaluators classified each tweet, the final label got assigned by the majority vote. As for the first approach, one is only interested in hate speech and neutral tweet classification, all offensive language documents in the dataset were dropped. Some tweets were retweets that were commented additionally by a user. As it could not be distinguished whether the original tweet or the retweet contains hate speech, these documents were filtered out as well. An example is shown below:

```
""@jaimescudi_: ""@Tonybthrz_: ""@jaimescudi_: I swear  
if oomf try talking to me tomorrow.."" @"" @BarackObama""  
pussy"
```

The original tweets can be found in between the "...". Same goes for tweets that cite other users without using the retweet option.

The second dataset *Hate Speech Dataset from a White Supremacy Forum* was not entirely given as a .csv file. Only the document annotations were given in a .csv file, all forum contributions were stored in separate .txt files. Only documents which could not be assigned to a single class (label "idk/skip") or referred to other documents (label "relation") were dropped.

The resulting common dataset was stored in a .csv file. It contains 2.491 hate speech documents and 13.336 non hate speech documents. The dropped offensive language documents make up 17.505 instances. In case the classification results are too poor, additional 2.818 offensive language documents can be added that were labeled as hate speech by one evaluator.

4.1.2 Corpus building

The common dataset is loaded from the .csv file into a pandas dataframe. After doing basic preprocessing like removing emojis and other irrelevant characters, spacy is used to build a tokenized corpus. The language model that spacy brings decides about stop word, punctuation and white space removal. No hard coded logic or stop word lists are used in this process. This keeps URLs or other tokens including punctuation as one token. Furthermore

no stemming was applied to the tokens, instead lemmatization was used as one can in this case later on use pre-trained word embeddings from i.e. Word2Vec. Furthermore tokenization works better using the lemmas instead of word stems (e.g. We'll becomes ["we", "will"] and not ["we", "'ll"].

4.1.3 Data analysis

As already mentioned, the acquired dataset is an imbalanced one, which can lead to a decrease in performance and accuracy with machine learning classification. As a comparison the paper [Oriola2020] also recognizes the class imbalance and tries to reduce it by applying a synthetic minority oversampling technique called SMOTE [3]. In general there are a few possibilities to tackle the challenge of unbalanced classes:

- changing the performance metric (e.g. F1-score instead of accuracy)
- undersampling, i.e. deleting instances from the over-represented class
- oversampling, i.e. adding copies of instances from the under-represented class
- generating synthetic samples (e.g by using SMOTE)

In our experiments we chose to try the different approaches and applied a simple undersampling, as well as an oversampling using SMOTE. Additionally, we used the imbalanced dataset to train a classifier to compare how this affects the performance.

In further analysis of the data, we had a look at the length of hate speech posts versus non-hate speech posts. This can be seen in Figure 1.

Here one can see, that the hate speech posts contain more words (tokens before cleaning) than non-hate speech posts. In average a hate speech post contains 18.18 words, whereas a non-hate speech post only contains 15.85 words. Unlike expected, the hate speech posts are longer than the non-hate speech posts.

A more interesting look at the data are the most commonly used words per class. As can be seen in the word clouds in Figure 2 there are some obvious differences, such that the hate speech posts use words like “bitch”, “faggot” or “nigga”. But interestingly enough, the non-hate speech posts also often consist of the words “trash” or “white”.

For a better insight with what data we are dealing, a few examples are shown in the following.

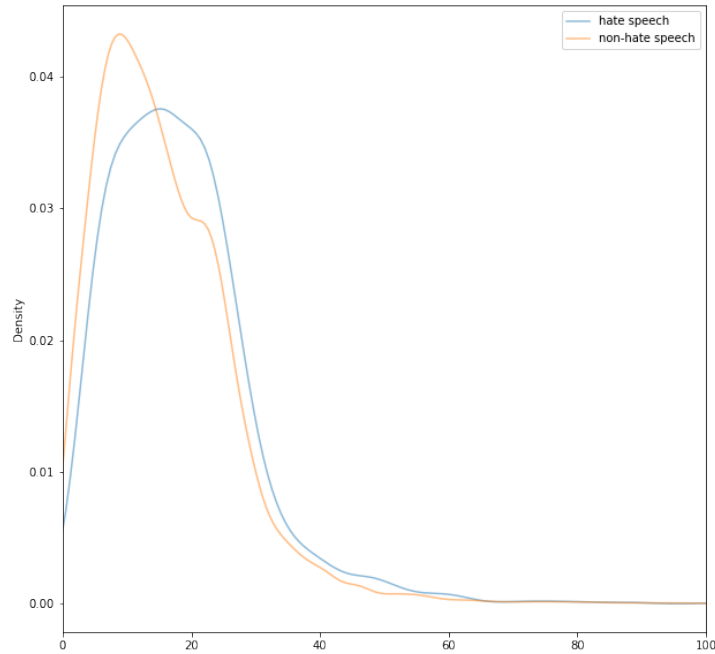


Figure 1: Density distribution of the length of a post (tweet or forum post)

Examples for non-hate speech (neutral sentences):

- "billy that guy would nt leave me alone so i gave him the trudeau salute"
- "this is after a famous incident of former prime minister pierre trudeau who gave the finger to a group of protesters who were yelling antifrench sayings at him"
- "askdems arent you embarrassed that charlie rangel remains in your caucus"

Examples for hate speech:

- "california is full of white trash"
- "and yes they will steal anything from whites because they think whites owe them something so it s ok to steal"



Figure 2: Word clouds

- "why white people used to say that sex was a sin used to be a mystery to me until i saw the children of browns and mixed race children popping up all around me"

One can clearly see the hate expressed in the hate speech examples and see their discriminating nature.

5 Analysis

6 Conclusion

References

- [1] Hind Saleh Alatawi, Areej Maatog Alhothali, and Kawthar Mustafa Moria. *Detecting White Supremacist Hate Speech using Domain Specific Word Embedding with Deep Learning and BERT*. 2020. arXiv: 2010.00357 [cs.CL].
- [2] Pinkesh Badjatiya et al. “Deep Learning for Hate Speech Detection in Tweets”. In: *CoRR* abs/1706.00188 (2017). arXiv: 1706.00188. URL: <http://arxiv.org/abs/1706.00188>.
- [3] Kevin W. Bowyer et al. “SMOTE: Synthetic Minority Over-sampling Technique”. In: *CoRR* abs/1106.1813 (2011). arXiv: 1106.1813. URL: <http://arxiv.org/abs/1106.1813>.
- [4] Arawinda Dinakaramani et al. “Designing an Indonesian part of speech tagset and manually tagged Indonesian corpus”. In: Oct. 2014, pp. 66–69. DOI: 10.1109/IALP.2014.6973519.
- [5] Wyatt Dorris et al. “Towards Automatic Detection and Explanation of Hate Speech and Offensive Language”. In: Mar. 2020, pp. 23–29. DOI: 10.1145/3375708.3380312.
- [6] Antigoni-Maria Founta et al. “A Unified Deep Learning Architecture for Abuse Detection”. In: *CoRR* abs/1802.00385 (2018). arXiv: 1802.00385. URL: <http://arxiv.org/abs/1802.00385>.
- [7] Prashant Kapil, Asif Ekbal, and Dipankar Das. *Investigating Deep Learning Approaches for Hate Speech Detection in Social Media*. 2020. arXiv: 2005.14690 [cs.CL].
- [8] Ona de Gibert et al. *Hate Speech Dataset from a White Supremacy Forum*. 2020. URL: <https://www.aclweb.org/anthology/W18-5102.pdf> (visited on 11/23/2020).
- [9] P. Roy et al. “A Framework for Hate Speech Detection Using Deep Convolutional Neural Network”. In: *IEEE Access* 8 (2020), pp. 204951–204962.
- [10] Arum Saksesi, Muhammad Nasrun, and Casi Setianingsih. “Analysis Text of Hate Speech Detection Using Recurrent Neural Network”. In: Dec. 2018, pp. 242–248. DOI: 10.1109/ICCEREC.2018.8712104.
- [11] Syahrul Syam, Budhi Irawan, and Casi Setianingsih. “Hate Speech Detection on Twitter Using Long Short-Term Memory (LSTM) Method”. In: Nov. 2019, pp. 305–310. DOI: 10.1109/ICITISEE48480.2019.9003992.

- [12] Thomas Davidson et al. *Automated Hate Speech Detection and the Problem of Offensive Language*. 2020. URL: <https://arxiv.org/pdf/1703.04009.pdf> (visited on 11/23/2020).