University *of Ljubljana*
Faculty *of Computer and Information Science*

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 traditional and state-of-the-art methods. Furthermore, we try to produce meaningful visualizations of the peculiarities of the different types of hate speech, as well as construct an offensive language schema based on the drawn conclusions from the analyses.

Keywords
hate speech, offensive language, exploratory analysis

Advisors: prof. dr. Slavko Žitnik

Introduction

Annotated offensive language datasets are essential to perform any kind of offensive language exploratory analysis. They enable us to categorize texts according to their offensive content automatically. The dataset should be such that it contains as much different hate content as possible, as our main goal is to explain these classes. We found many data sets that were promising, but at the end most of the data we analyzed came from the most prominent social media platform, Twitter. Some examples of offensive content are hate speech, obscene, vulgar, cyberbullying and many more. In this paper, we explore many of the existing methods that enable us to analyze hate speech. We also try to highlight the similarities and differences among these methods. In addition, we have made efforts towards presenting the differences between various hate content with visualizations.

Offensive language categories of interest

In the rest of the work, if not explicitly said otherwise, we will be interested in exploring the main characteristics and interrelationships of the following types of hate speech:

- Abusive
- Hateful
- Discredit
- Cyberbullying
- Harassment
- Profane
- Slur
- Vulgar
- Obscene
- Hostile
- Racist
- Homophobe
- Offensive
- Threat
- Sexist
- Insult

Data

For this project, we combined multiple publicly available datasets. Datasets are from different sources (Twitter, forums, etc.). Before using the data, we had to preprocess it. We removed URLs, user mentions, short words, emojis, and more. We used *ruby* script [1] that was made for cleaning Twitter data and done some processing with regular expressions in Python. In the following subsections, we describe how the datasets were gathered by the authors and some specifics about each of them. The distribution of the categories in the combined dataset is presented in Figure 1.

Multimodal Twitter [2]

The Multimodal Twitter dataset contains 150,000 tweets. The team gathered all the tweets from September 2018 until February 2019, and they contain all of the 51 *Hatebase* terms [1]. Tweets with less than three words were filtered out. Also, tweets containing porn-related terms were removed. Only tweets containing images were kept. Each of the tweets was categorized by 3 workers into one of 6 categories. Specific of

[1] List of these terms can be found at: https://hatebase.org/

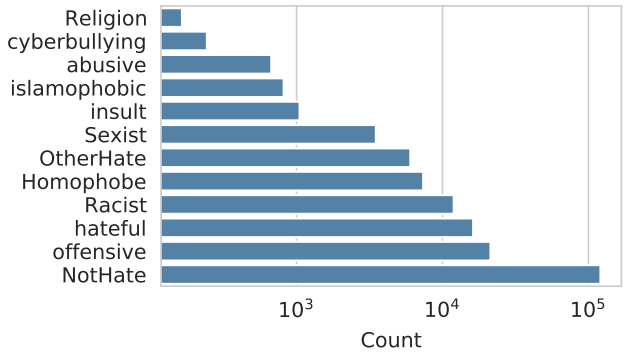

Figure 1. Histogram of offensive language categories present in the combined dataset.

this dataset is, that it also contains images that can be offensive. We didn't work with images, but we appended the text extracted from images to the text from tweets. The categories are: *no attacks on any community, racist, sexist, homophobic, religion-based attacks, or attacks to other communities*.

Conan [3]

Italian, French and English speaker experts (NGO) wrote prototypical Islamophobic short hate texts. Then operators responded to those prepared texts and provide additional hate text and response. The operators were following the same guidelines of the NGO for creating proper hate speech and responses. The motivation of the guidelines was to collect as much and diverse data as possible. Three non-expert annotators translated hate texts to English. Validation process has been conducted by NGO trainers. We extracted 408 texts, containing hate speech and put them into *Islamophobic* category.

Detecting cyberbullying in online communities [4]

Messages were gathered from forums of a popular computer game League of Legends. They were filtered by searching for words, contained in the wordlist from [2]. In this way, 20 topics were selected for each dataset. Annotation was performed by three human experts labelling each message in those topics. We combined all those topics into one category – *cyberbullying* (243 messages).

A Benchmark Dataset for Learning to Intervene in Online Hate Speech [5]

Dataset contains around 5k conversations retrieved from Reddit and 12k from Gab. Those were manually labelled as hate or non-hate speech by Mechanical Turk workers. This dataset also contains the human-written intervention response (which could be used to build generative models to automatically mitigate the spread of such types of conversations. We labeled those conversations with *hateful* category.

[2]https://www.noswearing.com/

Automated Hate Speech Detection and the Problem of Offensive Language [6]

Authors extracted tweets from around 33k Twitter users, which contained terms from the lexicon from https://hatebase.org/. In such way, they extracted around 85 million tweets and then they randomly sampled 25k tweets from it. Each tweet was manually labelled by CrowdFlower workers with one of the labels: *hate speech*, *offensive but not hate speech* or *neither*. We used tweets with *offensive* category.

Multilingual and Multi-Aspect Hate Speech Analysis [7]

Authors gathered tweets and then processed them by deleting unarguably detectable spam tweets, removing unreadable characters and emojis and masking the names of mentioned users. Annotators (smaller public) had to face the lack of context generated by this process. Each tweet was annotated with maximum three classes, but we were only interested in the *abusive* category. If two annotators agreed on two labels respectively, labels were added to the annotations. Dataset contains English, French and Arabic tweets, but we only used the English tweets (5k tweets).

Detecting Insults in Social Commentary [8]

Dataset consists of comments, that are labelled with 0 and 1, meaning an insulting comment. It was used in a competition, where participants predicted whether a comment posted during a public discussion is considered insulting to one of the participants. If it was directed toward non-participants (such as celebrities, public figures), it is not considered as an insult. Comments are taken from multiple blogs/forums. It is not known how the comments were annotated. We put those comments in the *insult* category.

Hate Speech Dataset from a White Supremacy Forum [9]

Around 10k sentences have been extracted from Stormfront (White Supremacy Forum). Web-scraping techniques were used and only English comments were filtered. 3 annotators developed guidelines on how to annotate sentences and designed a web-based tool, which allowed them to better understand post's author's intention. Sentences were classified as conveying hate speech or not. We used those sentences for the category *hateful*.

Methods

We used different approaches in natural language processing to explore texts with offensive language. Here we provide a list of the models we considered for our project:

1. the traditional technique TF-IDF (Term Frequency - Inverse Document Frequency)
2. non-contextual pre-trained word embeddings:
   (a) Word2Vec model,

(b) GloVe (Global Vectors for Words Representations),

(c) fastText model,

3. contextual pre-trained word embeddings:

   (a) BERT (Bidirectional Encoder Representations from Transformers

   (b) ELMo (Embeddings from Language Models)

Firstly, we gained some insights from the datasets by using the TF-IDF method to obtain embeddings. [3] We then used the KNN method for clustering. [4] Once having exhausted the possibilities of TF-IDF, we proceeded with the more recent models mentioned in the list above.

To visualize such high dimensional data we used methods to reduce the dimensionality of embeddings. Besides linear transformations such as PCA (Principal Component Analysis), we also used more advanced methods, such as MDS (Multi-Dimensional Scaling) and t-SNE (t-distributed stochastic neighbor embedding). Additionally, we used hierarchical clustering of embeddings and presented the data in a form of a dendrogram.

In the continuation, we delve deeper into the specifics of the applied models, and elaborate on their pros and cons.

Word2Vec model trained on all gathered data

As already described above, we went through all of the datasets at our disposal and merged the ones which could be integrated with the others. This way, we managed to double our text material. Having created this larger dataset, we trained a *word2vec* model on it. We decided to use embeddings of length 300, and a context window of size 10. Unfortunately, satisfactory results were obtained just for three of the dataset categories. Those categories and their most similar words were the following:

- **religion**: *Jews, Islam, religion, doctrine, cult, Christians, Christianity, Muslims, human, people*
- **islamophobic**: *Muslims, Islam, Britain, rapes, rapist, Muslim, anguish, law, Prophet, Mosque*
- **racist**: *racial, ignorant, white, saying, people, black, stupid, human, stating*.

Pre-trained fastText model on the entire Urban Dictionary [5]

Pre-trained word embeddings represent a powerful resource for doing exploratory language analysis. And, there exist a great deal of them, however most of them have been trained on corpora which contain text written in the standard form of a language. On the other hand, the vast majority of hate speech conversations take place on social networks where slang is also present. Consequently, these pre-trained models might not always be able to catch some patterns peculiar to a particular type of hate speech. Therefore, we have decided to take advantage of a pre-trained fastText model on the entirety of Urban Dictionary (all terms, definitions, examples, and tags were treated as running text) presented in [10]. This dictionary has an abundance of slang words and phrases which we believed could help shed a different light on the interpretation of hate speech.

The authors in [10] trained two separate fastText models. For the first model called **UD-base**, they tokenized the entire corpus without any explicit guidance on how to join phrases during training time. Also, they allowed the model to learn representations for word n-grams with a length of up to five. In **UD-phrase**, the second group of word embeddings, they considered all phrases and their various occurences throughout Urban Dictionary.

We utilized these two models by finding the 30 most similar words to each one of our categories of interest. However, during this search, we excluded all words having the same lemma or stem as a term in our list. By this procedure, we hoped to obtain more insightful and fair visualizations of the different word clusters.

As in the case of the word2vec model we trained, we used dimensionality reduction techniques to try extract meaningful patterns from the embeddings of size 300. The results, as well as the conclusions derived from them, are presented in subsection **Analysis of Urban Dictionary fastText model**.

GLoVe Model

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. It is similar to Word2Vec model, as both models learn geometrical encodings of words from their co-occurrence information. They differ in that Word2Vec is a *predictive* model, whereas GloVe is *count-based* model. Training of GloVe is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space [11].

The starting point for vector learning is the ratio of co-occurrence probabilities. Compared to probabilities, the ratio better distinguish relevant words from irrelevant ones and also is able to better discriminate between the two relevant words.

Authors of GloVe [11] found out that on the same corpus, vocabulary, window size and training time, GloVe consistently outperforms Word2Vec. It achieves better results faster, and also obtains the best results irrespective of speed.

[3] In these 2 notebooks: https://github.com/tm1897/nlp_offensive_lang/blob/main/notebooks/reddit_tf-idf.ipynb and https://github.com/tm1897/nlp_offensive_lang/blob/main/notebooks/stormfront.ipyt.ipynb TF-IDF analysis was performed.

[4] This notebook https://github.com/tm1897/nlp_offensive_lang/blob/main/notebooks/tf-idf_clustering.ipynb improves to the other two TF-IDF notebooks in the sense that k-means is also performed, and the obtained clusters are definitely reasonable.

[5] Urban Dictionary is a crowdsourced online dictionary for slang words and phrases. As of 2014, the dictionary had over seven million definitions. Their official webpage is: https://www.urbandictionary.com/.

Pre-trained Glove model on the Twitter data [6]

First we converted GloVe file containing the word embeddings to *word2vec* format for convenience of use - we have used `glove2word2vec` function. Then we took 30 most similar words to each of the given terms and also excluded all words having the same lemma/stem as a given term. Then we created visualizations of different word clusters and checked some analogies.

Trained Glove model on our dataset

We also trained our own Glove model on the combined dataset with nearly 300k sentences, containing one kind of 11 categories of hate speech. We trained a model with 100 batches, each batch consists of 200 threads. The embedding length was 300, thee context window was of size 10. The main problem in preparing the model was that the data was poorly processed. In addition to English, other languages or characters that are not used in English appear in the same sentence on Twitter and other forums. Due to problems with the unicode table, we had to invest additional work to properly import the stored models and to evaluate them.

ConceptNet Numberbatch Embeddings

ConceptNet Numberbatch represent a set of semantic vectors, and are also part of the ConceptNet open data project. ConceptNet is a knowledge graph that provides lots of ways to compute with word meanings, one of which is word embeddings, while ConceptNet Numberbatch is a snapshot of just the word embeddings. These embeddings benefit from the fact that they have semi-structured, common sense knowledge from ConceptNet, giving them a way to learn about words that isn't *just* observing them in context. Numberbatch is built using an ensemble that combines data from ConceptNet, word2vec and GloVe models, as well as from OpenSubtitles 2016. A more detailed description of the way these embeddings were obtained can be found in [12].

BERT

We used BERT for contextualized word embedding. To extract information from tweets, we concatenated each tweet string with "This is category", where the category is a category of hate speech (racism, sexism, etc.). Then we extracted vector embedding of the category at the end of each tweet, which also contained the context from the tweet. The method for word embedding with BERT is well explained in the Medium blog post [13].

Results

Analysis of Urban Dictionary fastText model

In Figures 2 and 3, we show two of the visualizations created from the Urban Dictionary embeddings. We chose these two pictures, because we believed they depicted well the relatedness among different categories. Additional visualizations can be viewed at [7].

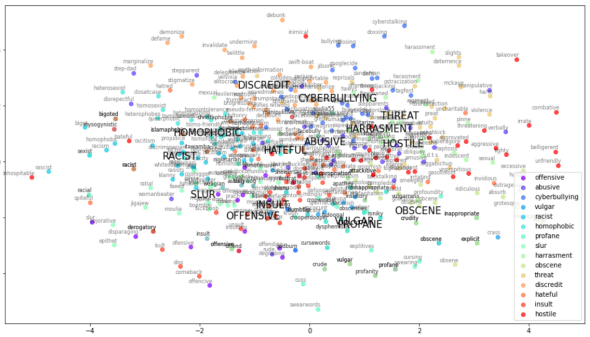

Figure 2. Multidimensional scaling performed on the embedding vectors in **UD-base**. Clusters tend to overlap, but in an explainable way.

In Figure 2, multidimensional scaling is performed on the embeddings from the model **UD-base**. At first glance, it might seem as if all the clusters are merged, however there exist pattterns. For instance, terms *threat, harrasment, hostile* are very close together, but that is not something we would not expect, because even humans would often classify a statement in each one of these hate speech types simultaneously. The same applies to *vulgar* and *profane* which have a significant overlap in our visualization.

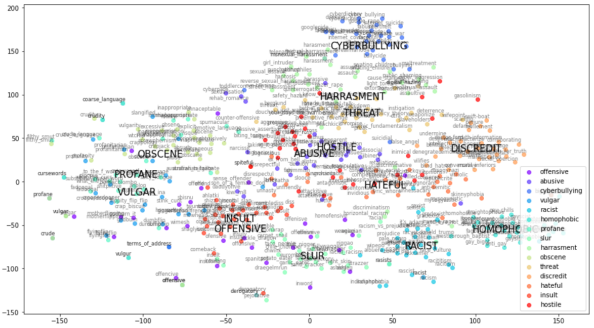

Figure 3. t-SNE was used to visualize selected word embeddings from **UD-phrase**. Created clusters were well described and appropriately grouped.

On the other hand, in the t-SNE visualization in Figure 3, we can easily notice distinct clusters. In addition, the grouping of the clusters is as one would assume it should be. The cluster of every term is surrounded by clusters of terms representing similar kinds of hatred.

To conclude, it should be noted that most terms' list of most similar words according to the models **UD-base** and **UD-phrase** is more adequate than the one produced by word2vec models pre-trained on texts of standard English. We believe the reason for this lies in the fact that Urban Dictionary has a

[6] 2 Billion tweets, 27 Billion tokens, 1.2 Million vocab, 25 dimensional vectors, accessible on https://nlp.stanford.edu/projects/glove/.

[7] https://github.com/tm1897/nlp_offensive_lang/blob/develop/notebooks/pretrained_fasttext_model.ipynb

much larger record of the strong language used in hate speech expressed on social networks.

Analysis of Glove

From Figures 4 and 5 we can observe clusters, constructed with t-SNE and PCA. We can see that clusters make sense and we also observe that if two clusters are close together using one of the methods, they are close the other one. The most interesting visualization seems the one on Figure 4, where we observe very clean distinction between some terms and some overlapping of terms in the middle. Those in the middle are the ones that a tweet/reply may belong to more of the categories, i.e. *hostile* and *hateful*. Because the model was trained on a really big corpus, most similar words really belong to the certain category of a hate speech.

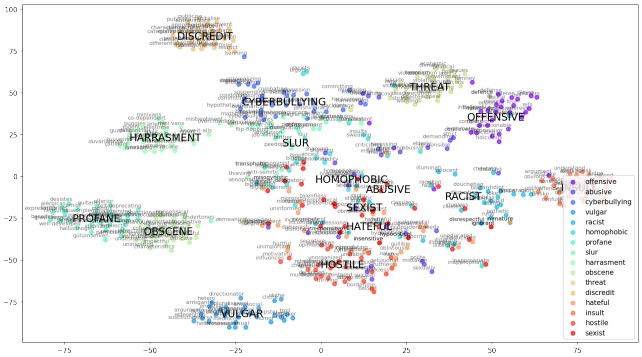

Figure 4. Similar words for pre-trained GloVe model using t-SNE method.

From Figure 5 we can also observe how similar certain categories of hate speech are. Harassment is more similar to profane than for example racist. We can observe some clusters of those categories of a hate speech - cyberbullying, discredit, slur form their own cluster. In many cases, hate speech can be placed in several categories, with cyberbullying more often being considered a discredit/slur than a racist (or some more distant category from Figure 2).

We can now compare the pre-trained model with our model. First, we want to point out that for similar words, our model chose some words that have nothing to do with hate speech at all. The reason is probably too little data for the training.

If we compare how the graphs of clusters of different categories differ, we see that in both models, categories are nicely separated (Figure 4 and 6). The main difference is that in our model, racist, homophobic and sexist categories overlap with each other, as well as insult and offensive categories.

Let's take a look at Figure 7 - it is showing how close the categories are from each other. We can observe a very concentrated cluster, which could mean that the model for several categories took the same words as the most similar words. This may be due to the fact that we did not have all the categories shown in the graph in our dataset.

We also compared both models by looking at analogies

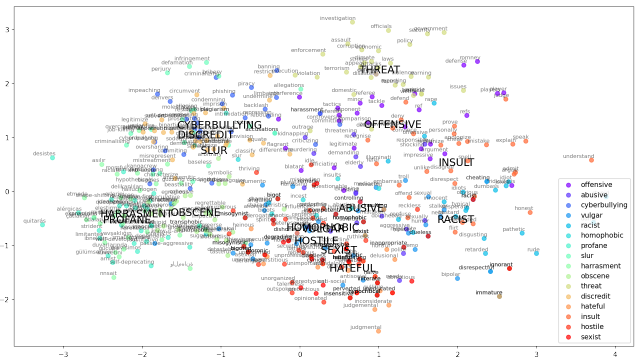

Figure 5. Similar words for pre-trained GloVe model using PCA.

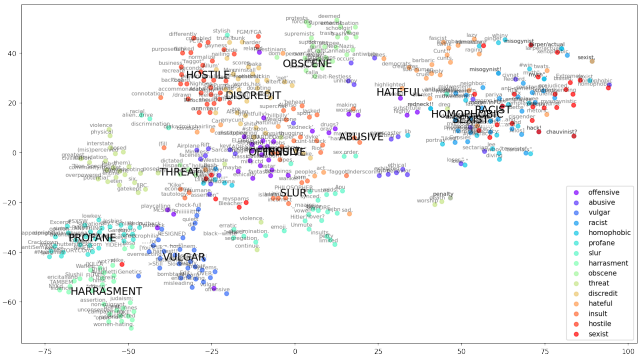

Figure 6. Similar words for trained GloVe model on our dataset with t-SNE.

where the theme of hate speech is used. Analogies we tested were:

- black is to nigger as white is to ?
- white is to trash as black is to ?
- obama is to cool as trump is to ?

Answer to the first analogy from the pre-trained model is "clown" with the similarity score of 87%. Answer from our model is "Anti-FirstAmendment" with the similarity score of 60%. Answer to the second analogy from the first model is "garbage" with the similarity score 92%. The second model returns "arizona" with 93%. For the last one, the fist answer is "netanyahu" (Prime minister of Israel) with 90% and "child" with similarity of 85%. We got a feeling that answers from the pre-trained model were more neutral in terms of a hate speech. The second answer "arizona" is not very reasonable, as we don't see any connection between Arizona, trash and black. The pre-trained model returns more meaningful suggestions, which is expected.

t-SNE employed on ConceptNet Numberbatch embeddings

As with the other pre-trained embedding from before, we employed dimensionality reduction techniques to the ConceptNet Numberbatch embeddings. It turned out that t-SNE was the

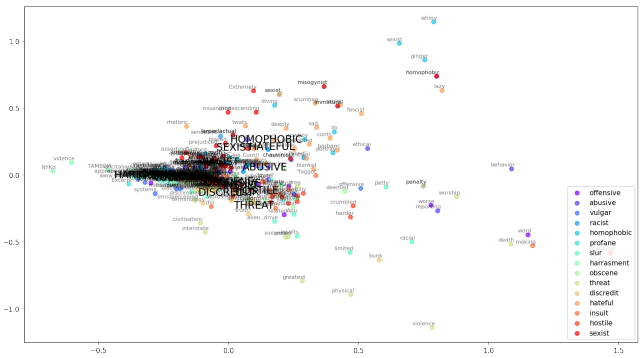

Figure 7. Similar words for trained GloVe model on our dataset with PCA.

most effective, and produced well distinct clusters visble in Figure 8.

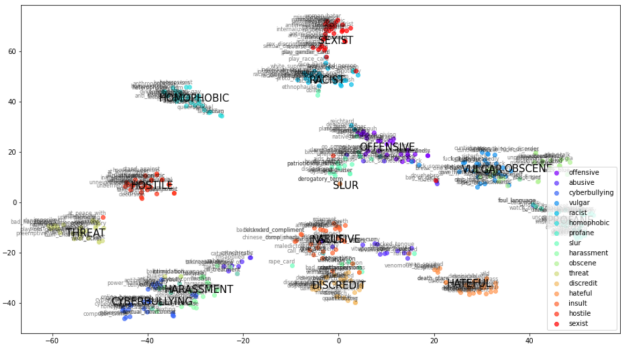

Figure 8. Employing of the technique t-SNE on ConceptNet Numberbatch embeddings led to well defined word clusters.

Another point we observed is that having ConceptNet as the core of the embeddings is rather beneficial. The graph model has a rich vocabulary, so when we search for similar words (same stem or lemma) which need to be avoided in the visualization, we find lots of them, and that was not necessarily the case with the other pre-trained embeddings. And, among the words that remain as the most similar to a particular term, we can definitely notice quite expressive and offensive ones. However, it should also be noted that for some categories, these most related words are not a good match to the category in question.

BERT

For each category, we computed the mean vector. Using cosine distance we plot the dendrogram which is shown in Figure 9. Three clusters are formed in the dendrogram.

First, we have *hateful* category, which has the least similarity to any other category. Probably because this category contains data from multiple sources, and authors can have different perspectives on what *hateful* means. In one dataset, the text documents contain conversations between multiple people. This also adds to the uniqueness of this category.

In the second cluster, we have two groups. The first group contains *offensive, abusive, homophobe,* and *not hate*. Texts marked as *not hate* are not innocent, but also contain a lot of hate speech. It quite often happens, that *no hate* tweets contain words, that are very frequent in other categories. The second group contains *insult, sexist, other hate,* and *racist*. Category *other hate* is often a combination of different hate categories.

The third cluster contains three categories: *islamophobic, cyberbullying,* and *religion*. It makes sense that *islamophobic* and *religion* are close to one another because the first is a subcategory of the second.

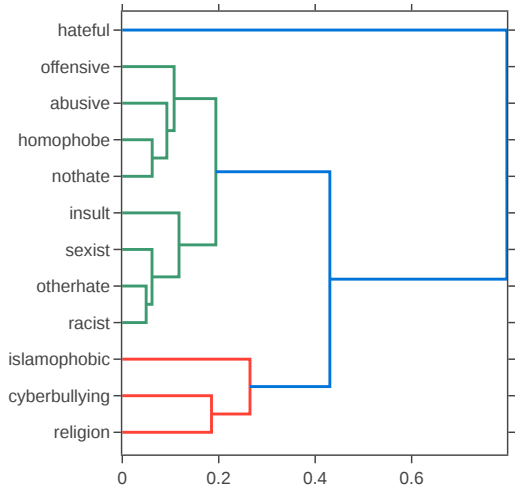

Figure 9. Dendrogram of hate speech categories average vectors by cosine distance.

We also extracted one tweet from each category, which is the most similar (by cosine similarity) to the average vector of its category:

- **Islamophobic** I will not allow my kids to marry a muslim! Go to hell Muslims, I hate you!
- **Cyberbullying** And another individual who can't cope with the fact that they are wrong degrades into personal insult because they can no longer validly argue their side of the story. You can flame and insult me all you would like. It just validates my point.lol You never had a valid argument, and you barely understand English. I hope Riot permabans your account for spouting such hateful trash.
- **Hateful** Just so odd this 'white knighting' from Republicans. @RealCandaceO advocates for felons in prison to vote and @kanyewest says if he was worried about racism he'd have left America long ago. These people, both of them are POISON.
- **Offensive** cant hate onna bitch getting paid for what she got
- **Abusive** @user wont waste time reading retard's hate speech cant even spell jealous correctly

- **Insult** DO YOUR JOB IN HATE LOSER
- **Homophobe** Yoda Declares Bert amp; Ernie NOT Gay; Sends SJW's Into Hateful Rage https://t.co/iEKraMFuk5 https://t.co/dGUUZn4QjjYODA CONFIRMS BERT AND ERNIE ARE NOT GAY: SENDS SJW'S INTO HATEFUL RAGE GAY, THEY ARE NOT BE ACCEPTING OF THIS, WE MUST.
- **Otherhate** @walegates Much as I hate to say it, but this Gussi pronouncing retard has a point? Shoe really does have size... https://t.co/DCrgxL1AO1None
- **Not hate** "Playing the 'Race Card', Race Baiting is Democrat/Socialist Hate Speech." trailer park prophet... https://t.co/Q9xSSNWWAeNone
- **Racist** I'm glad there weren't any nigger hoodies https://t.co/LFHsOGXHELReal hate comments received by our cast on social media.
- **Sexist** @Beys_Knees I see you too liking those BOSS hate comments you ain't slick cunt https://t.co/QATtD5k89GNone
- **Religion** @jncatron Because they are retarded! I don't like those who support those who hate Christians and Muslims! https://t.co/C3ryg2eo6LThe sole purpose of non-Jews is to serve Jews. Goyim were born only to serve us. They have no place in the world- only

Discussion

Availability of hate speech datasets

In this project, we explored the space of online offensive language. We were given a lot of datasets, but unfortunately most of them couldn't be accessed anymore. Datasets from Twitter were often containing only Twitter IDs, and the content, the tweets were deleted by Twitter due to supporting offensive speech.

Issues related to dataset preprocessing

We put quite some effort into cleaning the data. We both created our own preprocessing code and used pre-made scripts. However, there still exists ample room for improvement in this regard. We managed to remove URLs, user mentions, emojis, etc. to an extent that we were starting to obtain meaningful results even when training our own contextual models. Nevertheless, there are still many unknown characters which we were unable to remove. Additionally, social media conversations contain a lot of typos, slang, and other variations of the language that make the preprocessing stage even more involving. And, herein lies the main advantage of pre-trained models which have been developed and curated over a certain period of time.

Shortcomings of traditional methods

Classical methods for embedding documents, such as TF-IDF, are a good way to start exploring the data. Therefore, they represented our initial approach. Unfortunately, they have many shortcomings, because of only relying on simple word and document counts. So, the best insights when utilizing TF-IDF came in combination with k-means clustering, shown in [8].

Final schema for offensive language

Throughout the project, we tried to better understand the nature of hate speech and its properties. In doing so, we realized that there are several forms of hate speech. In it's most basic form, expressing vulgar words that address another person for the purpose of insulting, slandering, discrediting, etc. This form of hate speech is currently the easiest to detect on the Internet with various methods, which we also studied and tested in the project. Of course, there are various challenges, such as the use of slang or imitation of vulgarity in order to make it harder to perceive this speech. The next stage of hate speech is such that it does not contain distinctly vulgar words, but the hate is distinguished from the broader context of the text. One such example could be cyberbullying, which can be passive aggressive and it is harder to recognize automatically. One sensible division of hate speech could be based on how quickly and effectively it can be detected. The most aggressive group would include the categories: vulgar, hostile, insult, threat, offensive and sexist. A group that is somehow a little less aggressive, more passive, and doesn't use as many vulgar words would include homophobe, islamophobic, profane and religion. Here, hate speech is expressed more in the sense of slandering personalities worshiped by one religion or another. There is also a lot of homophobic speech present in very passive form, like 'go home where you belong', which is again, harder to detect. In the last group, we would include cyberbullying based on the reasons we have elaborated above.

To conclude our analysis of the pecularities of the different types of hate speech, we would mention another observation across which we came in almost all of our models. And, that is that there exist categories, classes which overlap significantly. In both of the models we trained, those categories were religion and islamophobic. In the visualizations of the pre-trained models however, this was very often the case for vulgar, profane and obscene.

Notes on other directions of research

Lastly, we would like to elaborate on ideas which we were considering at first, but later turned out to be inadequate for our task, as well as provide ideas for continuation and improvement of the research we've made:

- Topic modelling has gained a lot of attention recently, and there are certainly reasons for that. The idea can

[8] https://github.com/tm1897/nlp_offensive_lang/blob/main/notebooks/tf-idf_clustering.ipynb

serve very well to create interesting interactive visualizations. Nevertheless, on one of the axis the documents in the corpus are shown, in our case that would be sentences or tweets, and showing thousands of them is definitely infeasible. Hence, we abandoned the idea of performing topic modelling on our datasets.

- We used the general BERT model from Google, but there exist pre-trained BERT models that are specialized in a particular topic. If one could find a BERT model trained on offensive language dataset, that is similar to the one that we were working with, we would most likely be able to capture deeper patterns and connections among categories.

References

[1] Script for preprocessing tweets. https://nlp.stanford.edu/projects/glove/preprocess-twitter.rb. Accessed: 20.05.2021.

[2] Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. Exploring hate speech detection in multimodal publications, 2019.

[3] Yi-Ling Chung, Elizaveta Kuzmenko, Serra Sinem Tekiroglu, and Marco Guerini. CONAN - COunter NArratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2829, Florence, Italy, July 2019. Association for Computational Linguistics.

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

[5] Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. A benchmark dataset for learning to intervene in online hate speech, 2019.

[6] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language, 2017.

[7] Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. Multilingual and multi-aspect hate speech analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4675–4684, Hong Kong, China, November 2019. Association for Computational Linguistics.

[8] Recruitment prediction Competition. Detecting insults in social commentary, 2012. data retrieved from Kaggle, https://www.kaggle.com/c/detecting-insults-in-social-commentary/overview.

[9] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[10] S. R. Wilson, W. Magdy, B. McGillivray, K. Garimella, and G Tyson. Urban dictionary embeddings for slang nlp applications. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 05 2020.

[11] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[12] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge.

[13] Andreas Pogiatzis. Nlp: Contextualized word embeddings from bert. https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b.