FRI Natural language processing course 2021
Project report

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

Offensive language exploratory analysis

Rok Petrovčič Vižintin, Marko Krajinović and Tim Štromajer

Abstract

Online platforms are often used to express personal opinions and views. This can lead to disagreement and use of offensive language. In our explanatory analysis we explored the relations between different forms of such language. We collected texts annotated with 18 different types of offensive language classes and looked at their differences and similarities. Tf-idf was used to calculate similarities between different offensive language classes and then represent them with k-means clustering. We analysed the differences between the vocabularies of all the offensive language classes with non-contextual dense embeddings and found the most representative documents for each class. Contextual word embedding BERT was used to extract word and sentence embedding vectors for each offensive language class and then calculate the distance between them. Finally, we collected all the results, obtained with different algorithms and defined the final schema for offensive language.

Keywords
Offensive language, Tf-idf, Word2Vec, BERT

Advisors: Slavko Žitnik

1. Introduction

The internet has become a very common tool for people to express their opinions and communicate with each other. This role has become even more important during the COVID-19 pandemic. But unlike in the real world, where people with different views can be miles apart, everyone is much closer on the internet. Different groups of people or individuals are being targeted and harassed on social media, blogs, online games and more.

In this project we are working with the already annotated posts from different online spaces, some of which contain offensive language like hate-speech, racism, sexism, threats et cetera. We analyze each offensive language class to find its specific features and compare it with other classes to find similarities and differences. We use various methods to achieve this. Term frequency-inverse document frequency (TF-IDF) reflects how important a word is to a document in a collection of documents. With this method, we obtain the most important words for each offensive language class and compare their vocabularies. Using k-means and hierarchical clustering we get a picture which of offensive language classes are most similar to each other. Besides traditional methods, we also use contextual word embedding Word2Vec and non-contextual word embedding BERT. The Word2Vec algorithm is used to learn word associations from a large corpus of text to detect synonymous words and BERT is used to compute vector embeddings of different offensive language classes and compare them. For each of the method we also provide a visualization to better clarify our findings.

For visualizations we utilize different techniques based on the text representations. Sparse vector representations are visualized with rooted and un-rooted dendrograms. Dense representations are linearly reduced to an intermediate lower dimension using MDS and PCA.

2. Related work

2.1 Predicting the Type and Target of Offensive Posts in Social Media [1]

The authors of [1] set out to create an annotated dataset of offensive tweets called *OLID (Offensive Language Identification Dataset)* and test different natural language processing algorithm's performance on that dataset.

The dataset uses a three level annotation schema:

1. Level A: offensive or not offensive
2. Level B: targeted or untargeted insult
3. Level C: who the target of the insult is - a group, an individual or other

The data set was compiled using the Twitter API and searching for specific keywords, 50% of those political and 50% apolitical. It contains 14,100 English tweets, around 30% of those offensive. The set was annotated by hand by crowdsourced annotators.

The authors tested the dataset with a Support Vector Machine (SVM), bidirectional Long Short-Term-Memory model (BiLSTM) and a Convolutional Neural Network (CNN). The models were evaluated on the basis of detecting all three annotation levels, using a macro-averaged F1 score. They found that the CNN yields the best results.

2.2 Media Framing Dynamics of the 'European Refugee Crisis': A Comparative Topic Modelling Approach

The authors of [2] used Latent Dirichlet Allocation topic modelling to track the overall course of the European refugee crisis debate and media framing in 5 different countries and languages. They used a data set of 130,042 articles from 24 news outlets in Spain, Hungary, Germany, Sweden and the United Kingdom. The keywords they used were *refugee* and *asylum* in each country's respective language. They evaluated the LDA models by assessing their semantic and predictive validity.

They assessed the dynamics of the media coverage calculating the average number of articles connected to the frames per week, to see how discourse shifted. They found that the different peaks and valleys on the graphs were connected to real world events and that frame dynamics change drastically from the beginning to the end of the crysis.

3. Data

For the purposes of our analysis, we collect multiple datasets from various existing annotated sources. Our goal is to find relations between different types of offensive language. As such, it is important that the data includes numerous granular labels. The labels we collected cover various types of offensive language and targeted harassment of specific groups. In this chapter we cover all the datasets we used in our analysis.

3.1 Overview

Each dataset we used contained a subset of the labels for our research. General information about all the datasets is presented in table 1. The following subsections describe the datasets in more detail, providing insight into how they were collected and what their original purpose was.

3.2 The dataset from Automated Hate Speech Detection and the Problem of Offensive Language

In [3] the authors extracted 25k tweets and labeled them as one of three categories: hate speech, offensive but not hate speech, or neither offensive nor hate speech. Each tweet was labeled by three or more people, using the majority decision. 5% of tweets were labeled as hate speech, 76% as offensive and 16.6% as non-offensive.

3.3 The MMHS150K dataset

In [4] the authors created a manually annotated multi-modal hate speech dataset formed by 150,000 tweets, each one of them containing text and an image. They classified texts in one of 6 categories: No attacks to any community, racist, sexist, homophobic, religion based attacks or attacks to other communities. Each tweet was labeled by three people and classified by majority vote. They classified 112,845 tweets as non-hateful and 36,978 tweets as hateful. The latter are divided into 11,925 racist, 3,495 sexist, 3,870 homophobic, 163 religion-based hate and 5,811 other hate tweets.

Ref.	Content Source	Classes	Size
[3]	Twitter	hateful, offensive	24,802
[4]	Twitter	racism, sexism, homophobia, religious harassment	149,823
[5]	Facebook, Twitter	hateful, profane, offensive	7,005
[6]	Twitter	abusive, hateful	80,000
[7]	League of Legends forum	cyberbulliing	17,354
[8]	Twitter	racism, sexism, appearance-related, intellectual, religious harassment	24,189
[9]	Wikipedia	toxic, severe toxic, obscene, threat, insult, identity hate	223,549

Table 1. Information about the datasets that were used as the source for this research.

3.4 The HASOC dataset

In [5] the authors created three datasets from Twitter and Facebook for Hindi, German and English languages. Firstly they classified each text with binary classification: Hate / Offensive or Neither. Then they classified each offensive text in three categories: hate speech, offensive or profane and also as targeted or untargeted. The final dataset contains 7005 texts, out of which 36% are offensive.

3.5 The dataset from Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior

In [6] the authors obtained 32 million tweets, extracted metadata and filtered them, to obtain the final number of 80k tweets. Tweets were classified into four categories: abusive, hateful, normal, spam. Each tweet was classified by five different people.

3.6 The dataset from Detecting cyberbullying in online communities

In [7] the authors collected two datasets by downloading the general forum of the popular online games World of Warcraft and League of Legends. The annotation was performed by three people as a tuple of the form: (offender, victim, message). The resulting dataset 1 contains 16975 messages with

137 harassment cases and dataset 2 contains 17354 messages with 207 harassment cases.

3.7 The dataset from A Quality Type-aware Annotated Corpus and Lexicon for Harassment Research

In [8] the authors collected 50k tweets, based on the words in their lexicon, containing offensive words, covering five different types of harassment: sexual, racial, appearance-related, intellectual, political, and generic content. Each tweet was then assigned, by three different people, one of three labels: yes, no, and other. Finally, they excluded non-consensus texts.

3.8 Toxic Comment Classification Challenge dataset

This dataset was prepared for a toxic comment classification challenge by Jigsaw/Conversation AI hosted on Kaggle [9]. The text was collected from Wikipedia comments. Each comment was labeled by human annotators with 6 different classes. These included toxic, severe toxic, obscene, threat, insult and identity hate. The combined training and testing dataset contains a total of over 220000 labeled comments. Each comment contains a binary label for every class and can belong to multiple toxic language classes.

3.9 Offensive language classes

The data we analyzed contains text snippets pertaining to 18 different offensive language classes. The following list shows each of the classes and an example of a representative text snippet. The snippets were selected through ranking based on the method described in 4.

Offensive: ”First I charge my phone, then I text your bitch”
Hateful: ”@ICC has problem with gloves, bats and everything else other than games being washed out!”
Racist: ”The Nigger Marriage yeah fb is Sending me today”
Homophobic: ”Look who’s like two weeks late to this faggot trend”
Sexist: ”How you and every other feminazi look”
Religious harassment: ”I don’t trust the jews but one would have to be pig-brain retarded to trust you.”
Profane: ”He’s literally f***king the Flag... flagday TrumpIsATraitor”
Abusive: ”Bad part about taking trips with ya homie is all he gotta say is lets slide out ya jump but ya bitch been waiting on a vacay about a year na”
Cyberbullying: ”Fck you riot, you started to remove my previous posts, and Idiot get fired already, nobody likes you.”
Appearance-related harassment: ”lol i see you hang with that fat skank any! lol”
Intellectual harassment: ”think u need to read a little history dumb fuck ever heard of genghis or the persians”
Political harassment: ”because twatwaffle libtards would rather support her possible abortion than allow her 2 defend herself from rape”
Toxic: ”bite me bitch wikipedia sucks ass”
Severe toxic: ”You’re a motherfucking idiot You’re stupid as all hell and know *nothing*! Butt out of the business of your superiors!”
Obscene: ”There is definitely cleanup to be had after fucking bitches so hard and so long...”
Threat: ”I am going to kill you i am going to get a gun and blow your head off you stupid retard”
Insult: ”GAYWAD you are a huge homo”
Identity hate: ”You are a french faggot and you like watching men mud wrestle!!!!!!”

4. Methods

In this chapter we describe how we preprocessed all text from the datasets into a uniform shape. We also describe methods which were used in our explanatory analysis, and how we used them.

4.1 Data preprocessing

The datasets we collected originated from various sources. Due to that, they were all stored in a different formats. We transformed each one into a unified format which ensured a streamlined process for further analysis. The resulting data was structured in two columns. One contains the text instance an the other the appropriate offensive speech label.

Certain datasets did not contain speech instances in a usable text format. Those had to be preprocessed, before they were saved to the standardized format. The text entries from the League of Legends forum in the data-set [7] were stored as scraped HTML. This data was parsed for text and was further cleaned of forum quotation indicators and user id’s.

The [6] dataset only contained tweet id’s instead of text instances. We used those to read the text content of the tweets directly from twitter. This data was collected in 2018. Because of their age and offensive content, many of the tweets have since been removed from the platform. Due to that we used a subset of those that were still available.

4.2 Text preprocessing

In order to remove the noise from the text content and obtain more accurate results, we preprocessed each text instance. This subsection describes the methods we implemented. These filters were not applied to the data directly, but were performed during analysis. Due to this, each separate analysis method could utilize only the types of text preprocessing that benefits it’s properties.

The first method utilized regular expressions to remove URLs from the text. The web addresses do not benefit the analysis, because the addresses of each individual post are unique and as such do not give any insight about the content being posted.

A large portion of the source data originates from Twitter. Twitter posts contain certain tags that are specific to the platform. We implemented functions that filter out such content. One method removes the tags that mention a specific user, as this does not relate to the content of the text, but who it is addressed to. We also removed the ”RT:” tag, that signifies that the post has been re-tweeted. This tag is not unique to any specific type of language. As such, it only creates a correlation between posts that were collected from Twitter.

Lastly, we implemented a method that removes consecutive duplicate phrases. Certain posts contained spam, where text that was duplicated multiple times and posted consecutively. This was especially prevalent in the [9] dataset. In practice the multiple instances of the same post do not change the meaning, but they do affect results of some methods. In order to account for that our method detected duplicated consecutive text and only kept a single instance.

4.3 Term relevance in documents

We used Tf-idf to compute this metric. The resulting matrix was later used for k-means clustering. This method has two main drawbacks. Most documents only contain a small subset of terms, so the vectors are very long and sparse. The second is that the encodings don’t preserve the word order in encoding. As such, no context can be inferred about the usage of the words, only their relevance. We addressed the first problem by ignoring all terms that appeared in less than 10% of documents, or the terms that were present in more than 80% of documents. We mitigated the second draw-back by also encoding bi and tri-grams. These were encoded alongside the single terms. This approach adds some context regarding word order in the text.

We combined all data from one speech class into one document and preprocessed it using the methods described in 4.3. We used a list of all the speech class documents as the input to *scikit-learn*’s TfidfVectorizer. We used stopwords from *nltk*’s English stopword corpus, an n-gram range from 1 to 3, a document frequency maximum of 80% and a minimum of 10%.

4.4 Vocabulary analysis with non-contextual dense embedding

Each offensive class contains certain keywords that are specific for that type of language. We analysed the differences between the vocabularies of all the classes. This was performed using a non-contextual dense word embedding representations of the keywords.

The text was preprocessed, removing URLs, Twitter user and retweet tags. Additionally, all stop words were removed. The resulting text was then tokenized and stemmed using the porter stemmer. The text from each language class was combined and the most important terms were selected using TF-IDF. We only considered uni-grams, because the terms were further embedded using dense term embedding models. Only terms with the document frequency between 0.9 and 0.1 were considered.

The resulting terms were mapped back from stems to their full forms and embedded using different non-contextual dense models. Top 50 terms from each class were selected to represent the vocabulary. The embeddings were summed and compared using cosine similarity. In dense embeddings, such as word2vec, terms with similar meaning are mapped to a similar vector. This way, different words with common meanings make similar contributions to the final embedding.

The dense embeddings were created using pretrained models. We explored the results obtained by using the following three models:

1. Word2vec Google News 300 [10]: This model was created by Google. It was trained on approximately 100 billion words from a part of the Google News dataset. The vectors are 300 dimensional and contain mappings for 3 million words and phrases. They were trained on the skip-gram model. The embedded terms contain linear semantic relations in the mapped space.
2. GloVe twitter 50[11]: The model was trained on 2 billion tweets and contains embeddings for nearly 1.2 million keywords. The vectors are of 50 dimensional.
3. FastText 300 [12]: This model was trained on Common Crawl and Wikipedia data. The models were trained using the c-bow method. They contain embeddings for numerous different languages beside English. The vectors are 300 dimensional.

The vocabulary similarities between classes are visualized with a color matrix. This type of visualization shows which offensive language classes have similar vocabularies. The embeddings are also visualized as points in space. We reduced their dimensions using PCA and MDS to visually show how the classes cluster and the distances between them.

4.5 Representative documents for each class

Dense representations were also calculated for the full documents using the same embedding approach as the vocabulary analysis. In this instance, every term that appeared in the text post was mapped and combined into a singular embedding. This represents the terms of each document using the semantic awareness of dense vector mapping. This approach was used to find the most representative documents for a class. Examples of such representative documents are displayed in the data section 3. Cosine similarity is calculated between each pair of documents within the class. The documents with the highest similarity are chosen as the representative set of documents.

4.6 K-means clustering

We used *scikit-learn*’s KMeans class. The implementation used was *k-means++*[13]. We grouped the speech class documents mentioned in the previous subsection into 6 clusters, using a max iteration limit of 500 and a cluster seed reinitialization count of 500.

The goal of using k-means clustering was to see which offensive speech classes are most similar, and to extract the top 10 keywords of each cluster.

4.7 BERT

BERT (Bidirectional Encoder Representations from Transformers) produces word representations that are dynamically informed by the words around them.

The main purpose of BERT in our research was to extract features, namely sentence embedding vectors, from text data. We used the pytorch interface for BERT and a pretrained model released by Google named bert-base-uncased. The pretrained model has 4 dimensions: 13 layers (initial embedding + 12 BERT layers), a batch number equal to the number of sentences in a text, a token number equal to the number of tokens in a sentence and 768 features. We embedded each text from our datasets in two different ways: Text taken directly from dataset and Text taken form dataset with the added sentence ”This is <*offensive class*>” (for example: This is hateful.).

For each text which was directly taken from the datasets, we calculated BERT vectors for every word and then took the mean of all words to obtain a one dimensional vector for each text. We then calculated the mean of all text vectors for each particular class. This way every offensive language class was described by one vector, which were then easily compared by cosine distance. For texts with the additional added sentence at the end, we again calculated BERT vectors for every word, but then took the mean of only word vectors of the offensive class words in the added sentence. This way every offensive language class was again described by one vector and cosine distances were calculated between them.

5. Results

5.1 K-Means clustering

By using k-means clustering, we grouped the documents into 6 clusters and extracted the top 10 keywords from each cluster. To display the scatter plot in Figure 1, we used principal component analysis to reduce the tf-idf vectors to 2 dimensions.

Cluster	Classes	Keywords
●	intellectual, political	twatwaffle, dumb fuck, fucktard, fuckbag, asshat, shithead, amp, dickwad, assbag, fuckhead
●	obscene, threat, insult, toxic, severe toxic, identity hate	wikipedia, page, edit, article, vandalism, user, revert, motherfucker, fag, talk page
●	homophobic, sexist, appearance-related	dyke, fatass, camel toe, milf, skank, gt, gt gt, camel, toe, full gallery
●	profane, hateful	fucktrump, trumpisatraitor, shameonicc, borisjohnsonshouldnotbepm, doctorsfightback, icc, dhonikeepstheglove, amp, dickhead, douchebag
●	offensive, abusive, cyberbullying, racist	hoes, this nigga, amp, riot, white trash, op, bad bitch, y'all, runes, lil
●	religion	banislam, bansharia, islam, terrorism, stopislam, religion, muzzie, surrender, terrorism islam, islamist

Table 2. Clusters of offensive speech class and their top 10 keywords, colors correspond to those shown in Figure 1.

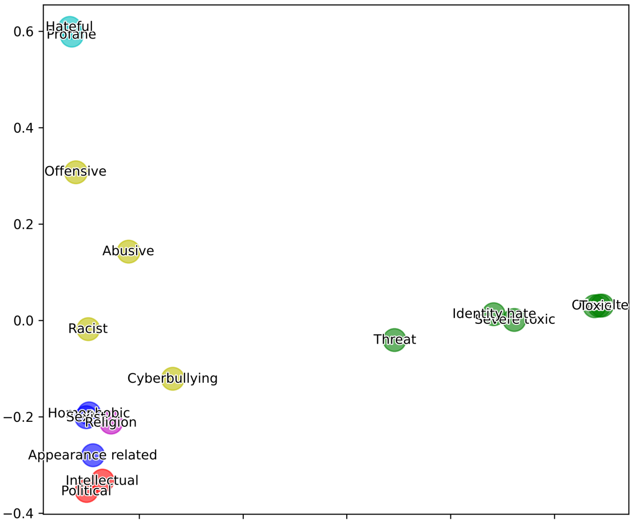

Figure 1. Scatter plot of speech class clusters generated with k=6 k-means clustering on tf-idf vectors, reduced with principal component analysis to 2 dimensions.

5.2 Vocabulary analysis with non-contextual dense embedding

Using the methods described in section 4, we created visualizations using different numbers of top terms, embedding models and dimension reduction techniques. The results are shown in figures 2 and 3. Other models produced very similar results. These visualizations were chosen as they were the most visually clear and provide contrast between two different embedding models. In all instances the mapping forms similar groups of classes.

5.3 BERT

We created two dendrograms using BERT, which is described in the previous section. Both dendrograms from figure 4 and 5 represent cosine distances between different mean BERT vectors, which were created by averaging text vectors from the same offensive language class. For vectors in the first dendrogram, every word vector was used for calculation and for the

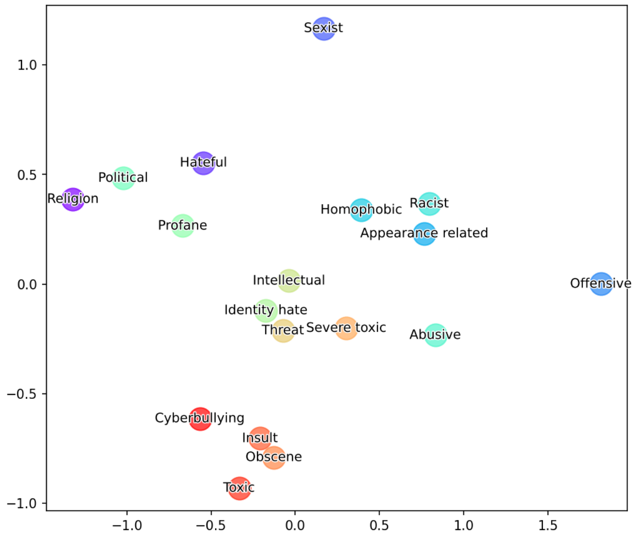

Figure 2. PCA reduction of top 50 terms for each offensive language class. Vocabulary represented as combined term embeddings from GloVe Twitter 50 vectors.

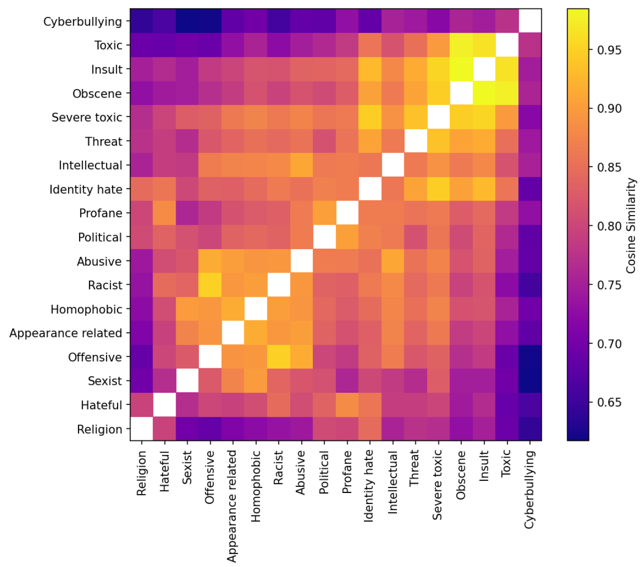

Figure 3. Vocabulary similarity matrix of top 50 keywords. Combined term embeddings from word2vec News 300 vectors.

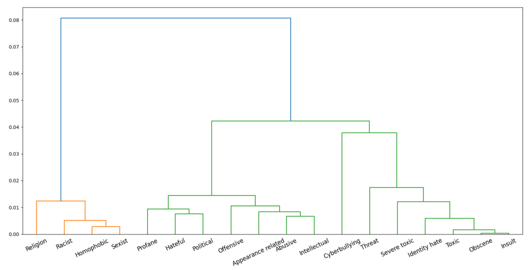

Figure 4. Dendrogram of cosine distances between mean bert vectors. Mean vectors were calculated using every text word vector.

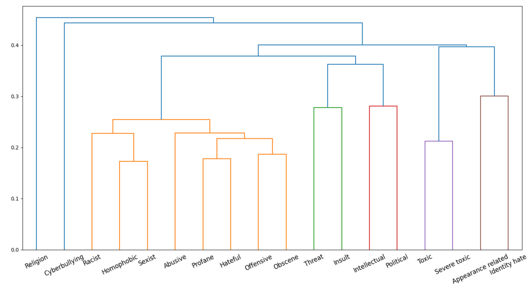

Figure 5. Dendrogram of cosine distances between mean bert vectors. Mean vectors were calculated using only offensive language class word vectors of added sentence.

second dendrogram only vectors of offensive word of added sentence were used for calculation.

With the first dendrogram we obtained 4 groups and with second 8 groups.

5.4 Result analysis

We found that all the methods that we used found similarities between the classes **homophobic** and **sexist**. Tf-idf based k-means and vocabulary analysis with non-contextual dense embedding (NCDE) also found **appearance-related** as similar. BERT grouped the first two classes with **racist** as well. NCDE also found **racist** and **homophobic** to be similar. All three classes use similar words like ”faggot”, ”nigga” and ”dyke”.

The classes **threat**, **identity hate**, **severe toxic**, **toxic**, **obscene** and **insult** are found to be similar by all three methods. This occurs because these classes come from the Kaggle [9] dataset, which contains specific discourse from Wikipedia. BERT finds them similar due to their common structure, and tf-idf finds them similar due to very specific terms used: ”wikipedia”, ”article”, ”revert”, ”vandalize”, et cetera. The group is further split into two smaller groups by NCDE. One of those two groups is closer to other identity based groups, because it uses terms such as ”nigger”, ”faggot”, ”muslim” ”nazi”. The other is mostly isolated, but is the closest to the **cyberbullying** class. BERT also finds this class to be closest to the classes with data collected from Wikipedia, as they are more similar in structure compared to social media posts.

NCDE and BERT found **hateful**, **political** and **profane** very similar. Tf-idf also found **hateful** and **profane** similar, but not **political**. This is due to the context of the conversations being similar, but the most frequent terms being different, due to frequent use of various dataset-specific hashtags (#fucktrump, #trumpisatraitor, #shameonicc, ...) in the text marked as **hateful** or **profane**. NCDE also detects vocabulary similarity due to words with similar contextual meaning (for example: fuckbag - fuckhead, dickbag - dickwad, republican - liberal), while tf-idf distinguishes them as separate terms.

offensive was with all three methods found similar with **abusive**. Both classes use very similar non-specific offensive words like ”ugly”, ”shit”, ”hoes”, ”sex”. Similarity is not that strong, because these words are also spread amongst most other classes. Bert also groups **hateful** together with these classes. They are contextually similar, because they are all general offensive classes that originate from social media posts.

The **religion** and **cyberbullying** classes were found to be the most different to other classes by almost all methods. While **religion** shares some terms with **identity hate** it mostly uses it’s own more specific terms like ”jews”, ”muslim”, ”islam” and ”christian”. Terms like ”anti-semite” were also detected as relevant by NCDE amongst the two classes. **Cyberbullying** has some overlap with other classes in the terms used as insults but it also uses jargon very specific to League of Legends: ”riot”, ”op”, ”runes”, ”silver” et cetera.

Intellectual classified differently by all three methods. NCDE found it to be relatively similar to most other classes. When using the Google News word2vec model, it was closest to **abusive**, while GloVe-50-twitter placed it closer to **appearance related**. Tf-idf and BERT found it relatively similar to **political** and **appearance related**. The reason for disagreement is due to smaller size of data for this class.

6. Discussion and conclusion

We found that a lot of results were overly-specific due to the nature of the datasets. For example, the cyberbullying dataset which was closely related to League of Legends, the Kaggle [9] dataset which was closely related to Wikipedia, and also any datasets containing tweets, due to the prevalence of hashtags.

Due to the subjective nature of language we also can’t be sure that the people who annotated unrelated datasets would agree on the annotations from a different dataset. Various classes could also be condensed into other classes as their children; for example both racism and sexism could fall under appearance-related. It would make sense to granularize all parent classes, or to annotate children with their parent classes as well. The data-sets [3], [5] and [6] label the documents as either hateful, offensive, abusive and profane. However they did not further annotate them as more specific classes, such as racism or political harassment.

A large impact on the results can also be attributed to the fact that some datasets used one or more annotations for each text instance, while some only used one. To achieve the best results, all dataset entries should be annotated with all the categories that they fall into. For future work, it would be useful to explore options in creating a more unified and structured dataset. This type of content would additionally benefit all future work in this domain.

A shortcoming of our approach with BERT is that it found differently annotated texts from the same datasets very similar, and could be improved by filtering out all words and signs specific for different datasets (Twitter, Wikipedia, gaming forum). Our NCDE method for comparing vocabularies for each class combined the most important keywords, which produced an overview of the terms. Mapping each individual term with this method did not produce clear clusters. Improving these results could show to which degree the terms overlap between different groups. This could potentially be improved by training the dense models on the provided dataset, ensuring mapping of terms more accurately represents the documents of analysis. Tf-idf’s bag of words approach cannot infer the semantics of different terms, and the only way to get around that is to use a different method.

6.1 Missing speech classes

There are four speech classes that we did not include in our research due to missing datasets. In this subsection we briefly describe their meaning in relation to other offensive speech classes.

Slur refers to language focusing on some identifiable feature of a certain group, usually ethnic or racial [14]. Based on this definition, it could be a subclass of identity hate.

Vulgar, according to Oxford Languages [15], means lacking sophistication or good taste and can also refer to making explicit and offensive reference to sex or bodily functions; coarse and rude. This class could be adjacent to sexism, homophobia, offensive and harassment.

Discrediting language [16] is intended to harm the reputation or authority of an individual. This class could be adjacent the intellectual harassment and political class. These classes often target an individual’s competence, intending to undermine their influence.

Hostile, by definition from the Online Etymology Dictionary [17] is language describing characteristics belonging to an enemy or a person of opposition. This could be a parent class of religion and political.

6.2 Offensive language schema

In this section we present an offensive language schema based on the results of this exploratory analysis and additional research. It explains the relations and distinctions between the classes. Based on the results that we obtained with all three methods Tf-idf, non-contextual dense embedding and Contextual word embedding BERT we created the final schema, which is shown on figure 6. We split 23 offensive language classes into 6 main groups, which are connected with different parent classes.

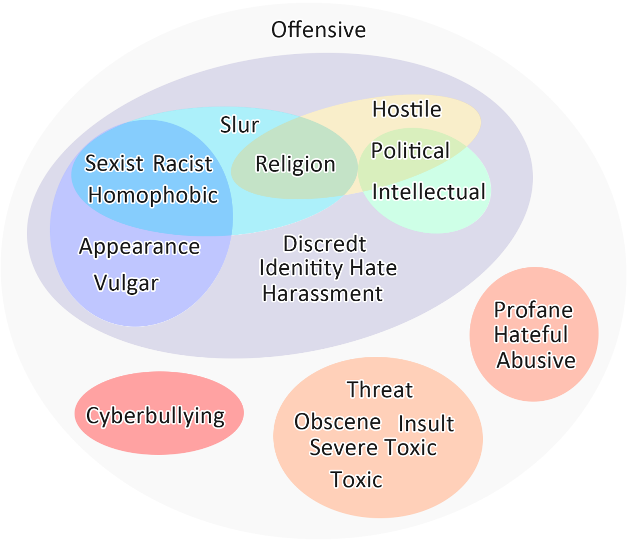

Figure 6. Diagram of offensive language relations. Ellipses show the membership of classes to groups and sub-groups.

The first group, which contains the classes **racist**, **homophobic** and **sexist**, target people's characteristics or features that they obtained at birth and usually do not change throughout their lifetime. Their parent class is **appearance related** and **vulgar**.

The second group contains **religion**. This class uses some very specific terms and is therefore alone in the group. The parent class of **religion** and **sexist**, **racist**, **homophobic** is **slur**, for the reasons described in the previous subsection.

The third group contains the classes **political** and **intellectual**. Both classes were collected from the same dataset, which is the main reason for their similarity. The parent class of **political** and **religion** is **hostile**, for the reasons described in the previous subsection. Parent classes of all previous groups are **discredit**, **identity hate** and **harassment**. They all target an individual's appearance or identity.

The fourth group contains classes **hateful**, **profane** and **abusive**. All three classes have very similar meaning and represent very similar texts among the different datasets.

The fifth group contains classes **threat**, **obscene**, **insult**, **toxic** and **severe toxic**. All classes were collected from the same Wikipedia dataset and therefore have similar context.

The sixth group contains **cyberbullying**. Due to its specific use of gaming vocabulary, it is the only class in it's group.

Finally, all groups are connected under the parent class **offensive**.

References

[1] Marcos Zampieri et al. Predicting the type and target of offensive posts in social media.

[2] Tobias Heidenreich, Fabienne Lind, Jakob-Moritz Eberl, and Hajo G Boomgaarden. Media Framing Dynamics of the 'European Refugee Crisis': A Comparative Topic Modelling Approach. *Journal of Refugee Studies*, 32(Special$_{I}ssue_{1}$) $: i172 - -i182, 122019$.

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

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

[5] Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19, page 14–17, New York, NY, USA, 2019. Association for Computing Machinery.

[6] Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior, 2018.

[7] Uwe Bretschneider and Ralf Peters. Detecting cyberbullying in online communities, 2016.

[8] Mohammadreza Rezvan, Saeedeh Shekarpour, Lakshika Balasuriya, Krishnaprasad Thirunarayan, Valerie L. Shalin, and Amit Sheth. A quality type-aware annotated corpus and lexicon for harassment research. *Proceedings of the 10th ACM Conference on Web Science*, May 2018.

[9] Toxic comment classification challenge.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.

[11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[12] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[13] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical Report 2006-13, Stanford InfoLab, June 2006.

[14] Brian Mullen. Sticks and stones can break my bones, but ethnophaulisms can alter the portrayal of immigrants to children. *Personality and Social Psychology Bulletin*, 30(2):250–260, 2004.

[15] Oxford languages via google.

[16] Merriam-webster.com dictionary.

[17] Online etymology dictionary.