University of Ljubljana
Faculty of Computer and Information Science

Offensive language exploratory analysis

Rok Petrovčič Vižintin, Marko Krajinović and Tim Štromajer

Abstract

...

Keywords
Keyword1, Keyword2, Keyword3 ...

Advisors: Slavko Žitnik

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 in the latest virus era. But unlike in the real world, where different minded people usually live apart, here on the internet everyone is much closer. Different groups of people or individuals are being targeted and harassed with the use of social media, blogs, online gaming and more.

In this project we are working with the already annotated posts from different social networks, some of which contain offensive language like hate-speech, racism, sexism, stereotype and other. We analyze each offensive language class to find its specific features and compare it with other classes to find similarities. For this task, we use different methods. 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 even which classes use similar vocabulary. With K-means clustering and hierarchical clustering we get a picture which 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 for classification. For each conducted method we provide 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 PCA. The lower dimensional data is further reduced using a non-linear t-SNE embedding to produce 2 and 3 dimensional scatterplot visualizations. Another form of dense representation visualization is created with radviz. The best attributes are selected from the reduced space through a machine learning approach using vizrank.

Related work

0.1 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% non-political. 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.

0.2 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 identified the following frames:

1. Economy
2. Welfare
3. Accommodations
4. Humanitarian aid
5. Refugee camps
6. Border
7. National refugee policy
8. EU refugee policy
9. Elections
10. Crime and terrorism
11. Refugee movement
12. War
13. Values and culture
14. Human interest
15. Unaccompanied children
16. Brexit

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.

Data

For the purposes of our analysis, we collect multiple data-sets 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. Our data contains text snippets pertaining to the following offensive language classes:

1. none
2. offensive
3. abusive
4. cyberbullying
5. profane
6. hateful
7. racist
8. homophobic
9. sexist
10. intellectual harassment
11. political harassment
12. appearance-related harassment
13. religious harassment

As the data was collected from different sources, we unified the format to a .csv file containing two columns. One contains the text instance an the other the appropriate offensive speech label. Such a format ensures a streamlined process for further analysis. The information about all data-sets is presented in the following table 1.

Ref.	Source	Classes	Size
[3]	Twitter	hateful, offensive	24,802
[4]	Twitter	racism, sexism, homophobia, religious harassment	149823
[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

Table 1. Information about the data-sets that were used as the source for this research.

Certain data-sets did not contain speech instances in a usable text format. Those had to be pre-processed, 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] data-set 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 only used a subset of those that were still available.

Methods

Tf-idf

Tf-idf is a text encoding method which utilizes sparse vectors. All input documents are used to create a corpus of terms that appear in the data. Each document is then encoded with a vector, where indices map to a term in the corpus and the value represents information about a given term in the document. In the case of tf-idf the term importance is positively impacted by the number of times it appears in the document, but negatively by the total number of documents it appears in. This relation is calculated with the following formula.

$tf-idf(t,d) = tf(t,d) * (log(\frac{n}{df(t)})+1)$

$tf(t,d)$ denotes term frequency: how many times the term t appears in the given document d. $df(t)$ represents document freqency: in how many documents the term t appears.

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 that appeared in less than five times, or the terms that were present in more than 50% of documents. We mitigated the second draw-backs, by also encoding n-grams. These were encoded alongside the single terms. This approach adds some context regarding word order in the text.

We combined all the data from one speech class into one document and preprocessed them using tokenizing and stemming functions. We also removed any punctuation and links. We used an array of all the speech class documents as the input to *scikit-learn*'s TfidfVectorizer. We used stopwords from *nltk*'s English stopword corpus, 200,000 vectorizer features, n-gram range from 1 to 3, a document frequency maximum of 50% and a minimum of 5 occurences.

The Tf-idf matrix was then used as the input for k-means clustering.

K-means clustering

K-means clustering is a method for partitioning n vectors into k clusters. Each vector is assigned to the cluster with the nearest mean. Clusters are constructed in such a way that each cluster minimizes within-cluster variances.

We used *scikit-learn*'s KMeans class. The implementation used was *k-means++*[9]. We grouped the speech class documents mentioned in the previous subsection into 6 clusters, using a max iteration limit of 10,000 and a cluster seed reinitialization count of 1,000.

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.

BERT

In the past, words have been represented either as uniquely indexed values (one-hot encoding), or more helpfully as neural word embeddings where vocabulary words are matched against the fixed-length feature embeddings that result from models like Word2Vec or Fasttext.

BERT (Bidirectional Encoder Representations from Transformers) offers an advantage over models like Word2Vec, because while each word has a fixed representation under Word2Vec regardless of the context within which the word appears, BERT produces word representations that are dynamically informed by the words around them. Aside from capturing obvious differences like polysemy, the context-informed word embeddings capture other forms of information that result in more accurate feature representations, which in turn results in better model performance.

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. The pretrained model has 4 dimensions: 13 layers (initial embedding + 12 BERT layers), batch number equal to the number of sentences in a text, 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:

1. Text taken directly from dataset.
2. Text taken form dataset with added sentence ”This is *offensive class*” (for example: This is hateful.).

After that, we took the mean of all text vectors of each particular class for two different embeddings. This way every offensive language class was described by one vector, which can be very easily compared by cosine distance.

Results

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.

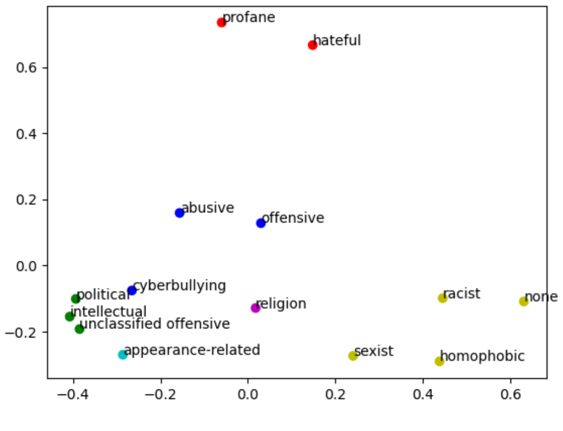

Figure 1. Scatter plot of speech class clusters

Cluster	Classes	Keywords
●	profane, hateful	icc, rapist, world cup, borisjohnson, fag, boris, impeachtrump, cricket, this world, traitortrump
●	intellectual, political, unclassified offensive	fuckhead, tard, spook, u dumb, hrc, fuck rt, libtard, douche bag, fuckoff, notmypresident
●	offensive, abusive, cyberbullying	bitch rt, riot, bad bitch, op, ing, fuck idiot, bitch n't, n't fuck, bad ass, bitch like
●	none, racist, homophobic, sexist	dyke, white trash, milf, nigga said, surrender, buildthewall, nigga got, hillbilly, gt gt, nigga nigga
●	appearance-related	fatass, skank, camel, fuckface, gym, pizza, grandpa, christmas, lose weight, gross
●	religion	terrorism, religion, muzzie, surrender, jew, spring, legacy, buildthewall, victim card, graham

Table 2. Offensive speech class clusters as shown in Figure 1 and their top 10 keywords

The classes which are closer use a lot of similar language. Similar style of discourse is also usually present in such classes. Intellectual and political are similar because political arguments tend to devolve into attacks on the other party's intelligence. Cyberbullying usually contains abusive language. Racism, sexism and homophobia are all similar in nature in the way that they attack different personal attributes. One would expect the religion class to be closer to the previous three, but it uses more distinctive keywords. Also interesting is the fact that profane and hateful use a lot of keywords that could be attributed to the political class, while not being very similar according to tf-idf. A reason for this could be that the political class is diluted with many other terms, and does not have a big overlap with the data used for the profane and hateful classes.

BERT

With BERT, described in previous section, we created two dendrograms. Both dendrograms from figure 2 and 3 represent cosine distances between different mean BERT vectors, which were created by averaging text vectors from the same offensive language class. The only difference between them is, that to each text, used in second dendrogram, was added additional sentence at the end: ”This is *offensive class*” (for example: This is hateful.).

From the first dendrogram we can clearly distinguish between three groups. In the first group we have offensive language classes: sexist, homophobic, racist and religion. All this classes have in common, that they target people's characteristics or features that they obtained at birth and usually do not change it through lifetime.

Second group in the first dendrogram contains offensive language classes: profane, hateful, political, offensive, appearance-related, abusive and intellectual. Authors of these texts usually do not agree with certain situations or views of other people. The only class that stands out from this group is appearance-related offensive language. It targets people's features which is typical for the previous group, but at the same time it also targets certain situation of people's appearance, which is typical for this group.

In the third group there is only one offensive language class: cyberbullying. The reason, why it is so distant from the other two groups, is, that the texts for this group were taken from computer gaming environments. The texts contain lots of words only specific for gaming.

Second dendrogram from figure 3 shows very similar results to the first one. The first group again contains offensive language classes: sexist, homophobic, racist, religion. But second and third group are not clearly separated as before. Hateful, profane, abusive, political, intellectual and offensive clearly form one group, but appearance-related is now more distant. This makes much more sense, as we have discussed before, that appearance-related offensive language can be typical for both first and second group. Cyberbullying in the third group is again the most distant from all the other classes, although now much closer to appearance-related offensive language.

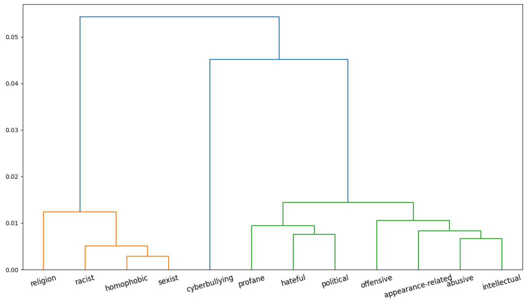

Figure 2. Dendrogram of cosine distances between mean bert vectors (direct text from dataset).

We also calculated cosine distances between each text and it's mean BERT class vector. For each class we took text that had the smallest cosine distance. This way we obtained the most representative text for each offensive language class. we were particularly interested in offensive words in these texts. Table 3 shows the extracted offensive words from each text, that we obtained this way.

From only looking at the used offensive words we can

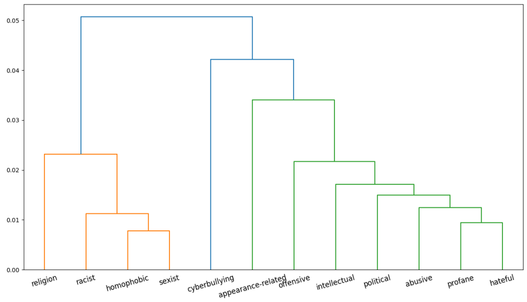

Figure 3. Dendrogram of cosine distances between mean bert vectors (text with added sentence).

again create partial groups, similar to the ones created by the two dendorgrams. Offensive, abusive, profane, intellectual, political and appearance-related use some similar words and could form one group. Racist and sexist classes also use similar words. Cyberbullying again can not be placed in any group.

class	words
offensive	stupid, bitch, niggas
abusive	stupid, bitch, fucking
cyberbullying	worthless, jerkoff, jackoff
racist	nigga
homophobic	gay, faggot
profane	shit, fuck
hateful	bad, troll, traitor
sexist	nigga, weak
appearance-related	fat, skank
intellectual	stfu, fat, fuck, fucktard
political	dumb, fuck
religion	trash

Table 3. Offensive words extracted from most representative texts for each offensive language class, calculated with cosine distance.

References

[1] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

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