Transphobic Speech Classifier

Custom project by

Marin Karamihalev
Siyana Ivanova
Jakub Blaszkowski
Wojciech Drezek
Jae Sun Lee

NLP 2019

Content Warning

Transphobic and misogynistic slurs; discussion of transphobia and misogyny; discussion of hate speech.

Overview

In this project, our goal was to observe the prevalence of transphobic content online, while also applying natural language processing methods for classifying a specific "subgenre" of hate speech in practice. Not a lot of research has gone into how trans issues are perceived on the Internet, which made this topic simultaneously more interesting and more challenging.[1]

The objectives were to create a machine learning model which distinguishes and classifies transphobia in English-language texts, test it on a real-life set of data (i.e. texts written by people in conversation online as opposed to generated by us for our purposes), and then analyse and interpret the results from processing the data, while also creating a reusable piece of software which can work on entirely new texts.

The resulting project is an application which classifies a text input into one of three categories: containing transphobia, related to trans topics but not containing transphobia, and unrelated to trans topics. To train our classifier, we used a dataset obtained from Reddit, a website containing various forums (called "subreddits") on a wide range of subjects.

Background

We chose this topic due to our interest in how online hate speech works and in trans rights. According to a study from the University of Arizona published in 2018, suicide among transgender adolescents is much higher than among their cisgender peers. In particular, among young trans men the rate of suicide attempts is as high as fifty percent. The researchers claim that a likely reason for this is the marginalisation and harassment that these individuals experience in their day-to-day lives.[1]

In this day and age, a large portion of communication happens online, on platforms like Reddit, which is especially notorious for hate speech (along with other social media like Twitter). This is why we attempted to examine the occurrence of transphobia (but also of positive or neutral sentiments on trans topics) on Reddit and study the circumstances in which it happens. When choosing to do this project, we hoped it may shed a little bit of light on this issue and provide some information about transphobia, giving our work some justification outside of creating another speech sentiment classifier.

For the purposes of the project we delineate transphobia in accordance with the dictionary definition,[2] but also in more specific terms as the presence of certain words, phrases, and sentiments.[3,4] Examples include but are not limited to slurs like "tranny,"

"trap," or "shemale;" phrases popular with trans-exclusionary radical feminists such as "TIM/ TIF" (trans-identified male/ female"); purposeful misgendering and insistence upon calling trans men "women" or trans women "men." This definition informed the way our team manually labeled the dataset we worked with.

The subreddits we picked to collect the data from were chosen for being representative of the three categories we were working with: transphobic (labelled -1), trans-related but non transphobic (labelled 1), and not trans-related (labelled 0). The full dataset consisted of 3000 posts - 500 per subreddit - and the selected forums are as follows.*

- *reddit.com/r/GenderCritical*
    - a community of trans- and sex worker-exclusionary radical feminists, chosen as routinely representative of transphobic content
- *reddit.com/r/MGTOW*
    - chosen in order to obtain some data containing non-transphobic hate speech, in particular misogyny
- *reddit.com/r/traaaaaaannnnnnnnns*
    - the content on this subreddit is by trans people for trans people, but it often contains slurs (reclaimed or used ironically), making it a good pick for nuances in speech
- *reddit.com/r/genderqueer*
    - a discussion forum for gender identity issues, chosen as firmly representative of the kind of post labelled with a "1"
- *reddit.com/r/Showerthoughts*
    - a subreddit dedicated to contextless, short form thoughts on a variety of topics
- *reddit.com/r/offmychest*
    - a subreddit dedicated to making anonymous confession-type posts on a variety of topics

NLP methods

In this section, we discuss the natural language processing techniques that our program employs. There were three NLP/ machine learning Python libraries used to apply these techniques: pandas, nltk, and sklearn.** The methods can be divided into three stages: cleaning the text, training the classifier, and producing the final results. Prior to these stages, the data was gathered from Reddit using a web scraper, and then came the process of manually labeling the posts with a value of -1, 0, or 1.

---

* The descriptions of the subreddits have been derived from the information in their respective sidebars, which can be seen at the web addresses provided.

** Found at
https://pandas.pydata.org/
https://www.nltk.org/
https://scikit-learn.org/stable/

Cleaning the text began by making every letter lowercase, followed by word tokenization - splitting each sentence into the words it is composed of. Next, all stop words were removed. Stop words are high-frequency words like “the,” “also,” or pronouns; they do not distinguish the text from most other texts and therefore are often removed.[5] Non-alpha words (numbers, symbols) were also cut from the text. Finally, the words were lemmatized - reduced to their root - using the WordNet lemmatizer[6] and part-of-speech tagged.

To train the classifier, we first split the data into a training set and a testing set. We chose a size of 0.2 (i.e. 20% of the data) for the testing set. The manually placed labels were then translated by the label encoder into the standard 0, 1, 2. Next, a TF-IDF vectorizer was used on the train and test data to convert the text to a matrix of token counts - marking how frequent a term is - and transform the matrix into a TF-IDF representation. Term frequency, in this case, refers to the number of times a word appears in a body of text, while the inverse document frequency is calculated by taking the logarithm of the number of documents in the corpus divided by the number of documents in which the term appears. The IDF serves to downscale the importance of words that are simply common.[7] After the calculation was made, the train X IDF data and the train Y data (the labels) were fitted to an SVM linear model. Then, the prediction scores were calculated by giving the SVM the test X TF-IDF data. Finally, the prediction was scored using a function from the sklearn library.

The standard classify function, given the SVM model, TF-IDF vectorizer, and the text, cleans the text, transforms it using the TF-IDF vectorizer, predicts the Y score, and returns the prediction.

Testing the application

Different tests were performed with variations of the SVM and naive Bayes classifier using the functions provided by the sklearn Python library. We tested the effects of swapping different kernels: linear, poly, RBF, and sigmoid. Then, for each one of them we tested changing the hyperparameters of the model. The linear kernel was used as is due to the lack of relevant options to alter. For the poly kernel, several different degrees were tested with 3 producing the best result; for RBF and sigmoid, the gamma value was changed, but the results were much worse than those from poly and linear kernels, so they were disregarded in the end.

The naive Bayes classifier has fewer parameters, so tests were only done with different values for smoothing. This did not improve the results or introduce any major changes. Using only the SVM linear classifier, we tested the effects of changing the steps of the training process. At the cleaning stage, we tried holding everything else stable while changing one aspect, for example leaving non-alpha words in the dataset and then observing the result. The best combination of steps we discovered for cleaning the data was the one that remained in the final program and is described in the Methods section above. At the training stage, we tested different numbers of max features for the TFIDF

vectorizer. The best results occurred around 4000, but the changes did not appear to have a huge effect.

Lastly, the classifier was tested with different testing set sizes, for example 0.05, 0.1, 0.2, 0.3, before settling on 0.2 as the optimal value. After testing, the accuracy score varied between 80% and 88% with the most influential factor being the chosen model. The linear kernel consistently produced the highest accuracy results, between 86% and 88%.

Results

The results obtained after training the classifier were largely satisfactory in terms of accuracy. The best score we were able to achieve by adhering to what we learned from testing the program was slightly over 88%.

```
In [136]: # Load data -> Train SVM model
          df = load_data(cleaned=True)
          model, tfidf = train(df)

          SVM Accuracy Score ->  88.33333333333333
```

[Figure 1]

In the early stages of testing the classifier with new inputs, we used short sentences which it evaluated mostly correctly, with a few notable exceptions. To illustrate this, we include some examples below.

```
In [162]: bad = 'trans people are not people'
          label = classify(model, tfidf, bad)
          print(bad, ":", label)

          neutral = 'cats are cute'
          label = classify(model, tfidf, neutral)
          print(neutral,":", label)

          pos = 'trans people are awesome'
          label = classify(model, tfidf, pos)
          print(pos,":", label)

          trans people are not people : [-1]
          cats are cute : [0]
          trans people are awesome : [1]
```

[Figure 2]

This simple, unambiguous language results in correct classification. However, there were some exceptions.

```
In [167]: ex1 = 'trans people suck'
          label = classify(model, tfidf, ex1)
          print(wrong,":", label)

          ex2 = 'trans women are women'
          label = classify(model, tfidf, ex2)
          print(wrong1,":", label)

          ex3 = 'trans men are men'
          label = classify(model, tfidf, ex3)
          print(funny,":", label)

          trans people suck : [1]
          trans women are women : [-1]
          trans men are men : [1]
```

[Figure 3]

The negative sentiment in the first example above is incorrectly labeled as positive. Additionally, the sentence "Trans women are women," which should be rated as "1," gives a result of -1 in this case, while the equivalent "Trans men are men" is correctly classified. There is a likely reason for this particular mistake, about which we admittedly merely hypothesise; it has to do with the source of the vast majority of examples of transphobia on which the classifier was trained, r/GenderCritical. The posts on this subreddit often speak of women in the context of being transphobic, directing hate speech mainly at trans women and often making the point that they should be barred from women's spaces. Therefore it seems plausible that the classifier takes mention of women and trans people together as transphobia. Excluding this interesting quirk, the initial results were adequate.

Further, more extensive testing with new inputs was performed next: 60 separate sentences were given to the classifier, of which 30 would correctly result in a mark of 0, 15 - a mark of 1, and 15 - a mark of -1. A classification report was made for this test, shown in Figure 4. The accuracy score was very close to the one obtained earlier: about 87%. Precision, or the rate of true positives given by the classifier[8], was worst for sentences which should have been scored with -1 - roughly half of the examples classified as transphobic were false positives. On the other hand, precision was 100% for both unrelated and positive/ neutral trans-related examples. Conversely, recall was best for transphobic sentences, with no false negatives appearing in this category at all. For 0 and 1 scores, recall was also very high - 86% and 83% respectively. This resulted in

good overall F1 scores (calculated as twice the product of precision and recall, divided by their sum[8]): above 90% for 0 and -1 and 64% for -1, brought down somewhat by the precision result.

nlp-project / test data / **results.txt**

```
Class counts
 0    30
 1    15
-1    15

Accuracy score: 0.8666666666666667

Classification report
              precision    recall  f1-score   support

          -1       0.47      1.00      0.64         7
           0       1.00      0.86      0.92        35
           1       1.00      0.83      0.91        18

   micro avg       0.87      0.87      0.87        60
   macro avg       0.82      0.90      0.82        60
weighted avg       0.94      0.87      0.89        60
```

[Figure 4]

Discussion

```
In [7]: full_data['class'].value_counts()

Out[7]:  0    2237
         1     574
        -1     189
        Name: class, dtype: int64
```

[Figure 5]

```
In [7]: full_data['class'].value_counts(normalize=True)

Out[7]:  0    0.745667
         1    0.191333
        -1    0.063000
        Name: class, dtype: float64
```

[Figure 6]

In the dataset of 3000 posts we collected, around 6% were marked at the manual labeling stage as transphobic, and 19% as trans-related but not negative, leaving nearly 75% of completely unrelated content. These results are partially due to our choice of subreddits: we purposefully selected two which we expected to be largely non-relevant to trans topicss, two that have a lot of posts discussing trans issues in a neutral or positive manner, one often containing transphobia, and one which attracts other types of hate speech (mostly misogyny). While this choice proved to be good for training the classifier, it does skew the impression one may get from the data about the prevalence of transphobia on Reddit. What we did learn is that trans-related issues as well as transphobia tend to be congregated in designated spaces, and the topic is somewhat fringe in "mainstream" Reddit where there is barely a mention of it. Respectful discussion mostly happens on subreddits frequented by trans people and allies, while fully intentional transphobia tends to occur on forums dedicated to ideologies which are at odds with the concept of gender being separate from biological sex. The percentages of each label are further skewed by the fact that there are two separate trans spaces on our list, while only one of the primarily negative subreddits is specifically transphobic. If we assumed to have an additional one with the same frequency of transphobic speech, we would end up with 12% as compared to our 19% of positive or neutral content - keeping in mind that the stated purpose of r/GenderCritical is to discuss radical feminist issues as they apply to cis women.

Conclusion

While the results obtained by our classifier were largely satisfactory for our purposes, and there was something to be learned about online transphobia from making this project, there are many ways in which it could be improved upon. These include but are not limited to gathering a new, much larger dataset to train a classifier on, this time accounting for bias that may result from Reddit slang or the specific nature of some subreddits and aiming for a more representative sample of the website's content; using datasets gathered from other social media, for example Twitter or Instagram; or exploring further ways to distinguish between transphobia and other kinds of online hate speech. Overall, the intersection between natural language processing and social issues is a deeply intriguing one and warrants further exploration.

Sources cited

1. Rapaport, Lisa. “Trans Teens Much More Likely to Attempt Suicide.” *Reuters*, Thomson Reuters, 12 Sept. 2018, www.reuters.com/article/us-health-transgender-teen-suicide/trans-teens-much-more-likely-to-attempt-suicide-idUSKCN1LS39K.
2. “Transphobia.” *Merriam-Webster*, Merriam-Webster, www.merriam-webster.com/dictionary/transphobia.
3. Dennis, Riley J. “Anti-Trans Slurs You Shouldn't Use | Riley J. Dennis.” *YouTube*, YouTube, 7 Dec. 2017, www.youtube.com/watch?v=_LO0jic6f9w.
4. Wynn, Natalie. “Are Traps Gay? | ContraPoints.” *YouTube*, YouTube, 16 Jan. 2019, www.youtube.com/watch?v=PbBzhqJK3bg.
5. Bedi, Gunjit, and Gunjit Bedi. “Simple Guide to Text Classification(NLP) Using SVM and Naive Bayes with Python.” *Medium*, Medium, 9 Nov. 2018, medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34?fbclid=IwAR1NGg9m-PCouy_sUb2hFTh-Md9__lAmvN9LHhgowCHIAmSSETWZfHz4dwc.
6. “Source Code for Nltk.stem.wordnet.” *Nltk.stem.wordnet - NLTK 3.4.1 Documentation*, www.nltk.org/_modules/nltk/stem/wordnet.html.
7. *Idf :: A Single-Page Tutorial - Information Retrieval and Text Mining*. www.tfidf.com/.
8. “A.I. Wiki.” *Skymind*, skymind.ai/wiki/accuracy-precision-recall-f1.