Tweet Classification of Hate Speech

Joseph Allen

November 20, 2017

1 Introduction

I will be performing an investigation into the classification of hate-speech, offensive(but not hate speech) and neutral Tweets using the provided Tweet data. I have previously performed sentiment analysis of Tweets, but I believe the difficulty in this project will come from the necessary distinction between offensive and hateful tweets.

I have opted to choose the NLP task as I would like to challenge myself to learn something new from this interview process.

2 Exploration

I will first investigate the dataset provided in a jupyter notebook. My initial thoughts here are that we need to do some basic cleaning of the data. Mainly cleaning the following from the Tweets:

- Replace mentions with a generic mention.
- Replace links with a generic link.
- Clear any redundant punctuation.

3 Potential Feature Engineering

Now we have some cleaner Tweet texts next we should think about features which may affect the volume of hate speech.

3.1 Spoken Language

As much as I wish it to be untrue I believe language spoken may reflect a different culture and potentially a change in hate speech. It should be relatively simple to detect language from text strings. I will come back to this later if it is needed.

3.2 Naive Sentiment Analysis

simply performing sentiment analysis using the *NLTK* library may give us some insight into the differences between Hate speech and negative speech. This will be trivial to implement so I will attempt this first.

3.3 Naive hate word dictionary

we could naively create, or search for a list of hateful words and slang terms, these may be used in conjunction with naive negative sentiment to distinguish hate speech. I found a shortlist on Facebook used for social media purposes, however this has seemed too general for use as it contains rude words which are not considered hateful.

A further piece of work could be to try to make my own lists from the following websites :

- Disability Slurs - https://en.wikipedia.org/wiki/List_of_disability-related_terms_with_negative_connotations

- Homophobic Slurs - https://en.wikipedia.org/wiki/Category:Homophobic_slurs
- Misogynistic Slurs - https://en.wikipedia.org/wiki/Category:Misogynistic_slurs
- Racial Slurs - http://www.rsdb.org/full

4 Data Cleaning Pipelines

As per the work explored in the notebook, I am going to write an sklearn pipeline which will automatically clean the data and retrain the model. This is beneficial as it allows rapid changes to data as it arrives and is easy to manipulate. I have not used these before, but again I am using this as an opportunity to learn. Now that this is written the same process can be applied to the labeled and unlabeled datasets. Now we can start with the model.

5 Model

A great starting point is to create a baseline, for this I would expect a baseline of about 33 percent as there are three options and an unintelligent model could guess about a third correctly. Luckily the data has already been split for us.

I have decided to start with a simple decision tree, from my initial application I have an accuracy of about 67%. I am expecting that the neutral tweets are correctly classified, whereas the difficulty in distinguishing between offensive and hateful is where the difficulty lies. From basic grid search we have pushed our accuracy to 69.88%. This is a minor improvement which makes me think I need to engineer more features, however due to time constraints I will stop here. After a brief investigation into overfitting and playing iwth parameters I have managed to reach about 72.7% accuracy without overfitting.

5.1 Overfitting

A naive way of testing for over fitting is trying to model on it its own training data, if there is a large improvement on the test score this normally means the model has been overfitted. the test score is about 76%, which would mean that grid search has lead us to overfitting, a common problem.

6 Facebook fastText

Facebook has released a new project called fastText which lets you trivially train a model for text classification, a perfect fit for this type of project. I have left this for last to see if it beats my model. This is a great choice as it boasts to support 294 different languages. this requires some basic changes to our data but shouldn't be difficult. We have 18537 unique words in the test data (top 10000 rows), this trains a model which we can check against our validation data (bottom 2657 rows). This initial dive has given a precision of 0.4%, not very useful, we can scale this up using more epochs or by adding more data. I will leave this for now due to time constraints.

7 Evaluation

Here I will answer the more specific questions at the end of the task.

7.1 Linguistic features

I would describe by brief delve into Natural Language Processing to be naive, I have mainly focused on using data science techniques, only additionally adding sentiment analysis from a library I have used in the past. If I had more time for this project most of my work would go into feature engineering, specifically more features to distinguish between hateful and offensive text. I think creating a custom slur dictionary would have been another naive feature I can add here.

The negative,neutral and compound sentiment scores from the nltk library performed surprisingly well at classifying neutral and negative responses which is half the project.

7.2 Performance

my initial attempts at a model have got to about 72% accuracy, this is by no means groundbreaking and again the difficulties come from classification of offensive speech without much investigation other than comparison to a slur dictionary. I think this could have been improved with a better slur dictionary and some more useful NLP techniques.

The Decision Tree model works averagely, we could reduce noise by tuning parameters further, or using a random forest model. I don't think much more can be gained through these methods though, instead a focus on feature engineering is my recommendation here.

7.3 Extra considerations

I addressed the overfitting discussion in an above paragraph, this level of accuracy with no overfitting for a brief investigation makes me quite happy, especially considering some may not even check for overfitting. More data as well as more investigation into which words make up hate speech would do wonders for this project and I should have investigated it earlier and given it much more time.

This model can add bias on certain words simply by having a slur dictionary that is considered much more offensive, noisy comments tend to have very low or even zero sentiment so again this is something to further investigate.

7.4 Explanation to End user

We check each posted tweet and count how many offensive words that tweet contains, on top of this we calculate the sentiment of the tweet and try our best to distinguish between offensive content and negative opinions. The idea here is to only remove content which is hurtful to society, instead of simple or even heated disagreements.

7.5 What is next?

The pieces of work to be done next are :

- Create a better dictionary of slurs
- get a larger amount of data as this improves model accuracy
- apply some NLP techniques to improve our features