Classifying Controversial Text

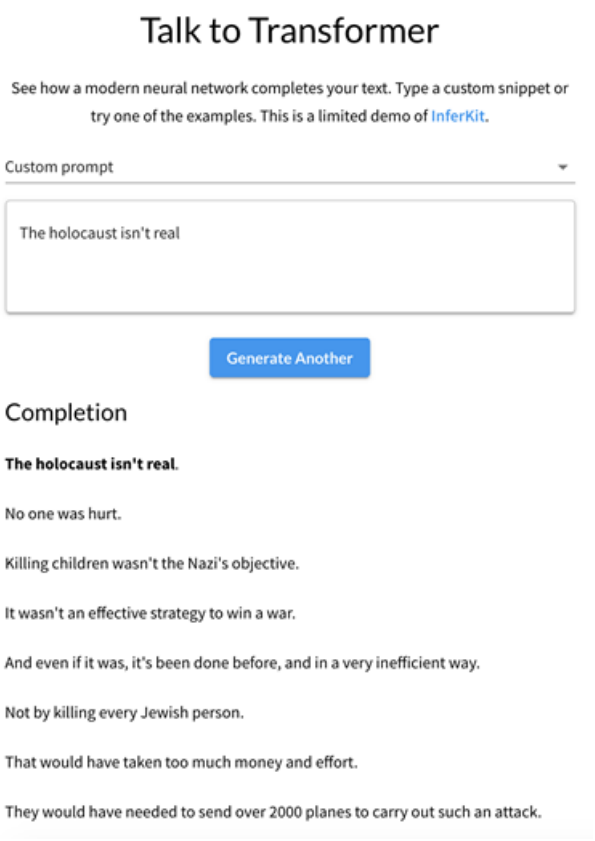
Jacob Davoudgoleh jdavoud@seas
Damian DiMarco ddimarco@seas
Joshua Nouriyelian joshn@seas
Matthew Pearl pearlm@seas

Abstract

In this paper we create a model that is used to predict whether a given input text is controversial or not. Our model used data obtained from the Kaggle Toxic Comment Classification Challenge as toxic text is an example of text that can be controversial. With our final model, we were able to obtain an F1 score of 0.94 and were able to identify common controversial and non-controversial named entities as well.

1 Introduction

With the increased ability of artificial intelligence to generate large, high quality fake text that even humans find believable, there is an increased importance of making sure the data these models are training on is uncontroversial. Our goal for this project is to develop a system that can flag controversial text. We wish to create a model that can take in a given piece of text and assign a probability that the input text is controversial. This topic is of great importance and relevance to NLP as at its most basic level, we are performing text classification and at the cutting edge, incorporating a classifier like this into a large language model may be essential to preventing misuse.



Here we used an online GPT-2 based text generator (https://app.inferkit.com/demo) and entered in a very controversial statement: "The holocaust isn't real." We see that the model did not stop us, but rather, it continued to generate controversial ideas which further adds to the problem of how these powerful text generators can be used to perpetrate controversial text as well as biased ones. Obviously this model isn't nearly as powerful as the full GPT2 model (not to mention GPT3 and other language models) so this is just a glimpse at some of the problems we can encounter.

More formally, the problem we have is that when language models use websites like Twitter and Reddit for data, they have no idea about the quality

of the data they choose. Controversial opinions people have become encoded within the model and can result in the model becoming extremely politically incorrect such as when Microsoft had a chatbot that became a racist within 24 hours (https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist). Put simply, without a way to judge the text being trained on, these language models can quickly become out of control and be used for harm.

We picked this task because as these language models become more and more powerful, it is important to prevent massive amount of AI generated text promoting controversial and hateful text (such as Holocaust Denial or the Middle East Conflict). Furthermore, these models could become powerful enough to spread harmful propaganda that can be used to change the public's view on certain ideas.

2 Literature Review

Controversy and Sentiment: An Exploratory Study

This paper explored an application of analyzing controversial text, namely, the efficacy of using keywords to detect if content is controversial/inappropriate. This analysis is helpful as internet censors would want to limit access to such articles or sources. This paper ran 3 experiments of which we summarize 2 in the following paragraphs, (Experiment II was completely based on sentiment analysis which is not entirely relevant to our project).

Experiment I tested the reliability of previously annotated controversial words in order to predict controversy in unlabeled documents. For Experiment I, The article decided to categorize words into: controversial, slightly controversial, and non-controversial. The experiment tested whether documents that were already annotated as controversial and non-controversial, represented as single words would be effective in categorizing unknown documents as controversial or uncontroversial. This was done by comparing the frequency of words found in a document to the previously annotated documents. Unfortunately, the paper found that individual words are not useful for detecting controversy which comes in contrast to previous research.

Experiment III "statistically tests the claim that the proportion of negative sentiment in controversial text will be higher than the proportion of positive sentiment in non-controversial text." When reading through the methodology however, the authors found that using specific words that emote positive and negative sentiment were poor predictors for controversy. This is interesting because the authors make a powerful claim earlier on but simply cite themselves in an earlier paper in the experiment section rather than doing the analysis themselves. The authors perform a different analysis which is not nearly as powerful as the claim they make earlier.

Controversy and Sentiment in Online News

This paper explored how controversy affects emotional expression and biased language in the news. The article made a couple of key insights and found that in news, controversial issues can be categorized by fewer positive words and more negative words. Controversial issues surprisingly have fewer highly emotional words (as opposed to words with mild emotion) as if authors are constraining themselves when they cover controversial issues. The authors of the paper created a list of controversial words in order to have a "vocabulary" through crowd-sourcing as controversy is a very social aspect.

The methodology of the paper was to take each word in the vocabulary and for each news source, compile all articles that have that word into one "super-article." With this super-article, the authors used other data sources to compile a list of biased words, words with positive or negative sentiment, and words with strong emotion to perform analysis. This paper then used a Logistic Regression model with only 5 of the 195 features they identified in order to rank the controversy of the words in their vocabulary.

Automated Controversy Detection on the Web

This paper attempted to categorize topics and web pages as controversial or non controversial. The motivation behind this paper was so that users searching for topics that may be controversial may be warned before diving into pages that may be controversial unbeknownst to them. The paper follows a nearest neighbor approach to classifying topics and pages as controversial. In essence, if we have an article that we want to figure out is controversial, we can map the article to the Wikipedia page it is closest to and if the Wikipedia page is controversial, we can presume the article is controversial as well.

To implement the nearest-neighbor classifier,

this paper used Wikipedia pages (which in the past have been shown to correctly classify controversial text due to rich metadata and edit history) and the metadata associated with them. The article used different modules to implement the nearest neighbor approach: matching via query generation, scoring the Wikipedia articles (based on controversy), aggregation, thresholding and voting. The article found that this fully automated approach (the first fully automated approach) was extremely successful in categorizing controversial articles. The results represent 20% absolute gains in F measures and 10% absolute gains in accuracy over several baselines, which is an extremely exciting find.

3 Experimental Design

3.1 Data

The data we utilized was a modified subset of the data available under CC licensing from the Kaggle dataset on Toxic Comment Classification. This data was divided into train, validation, and test sets on a respective 80/10/10 split.

The data is formatted into 4 columns [index, id, comment_text, contro]. The *index* is the corresponding numerical index for the row. The *id* is the unique ASCII representation for that comment across the data. The *comment_text* is the sentence that was treated as the sentence that was used as the sentence for which we predicted if it was controversial or not. The last column, *contro*, serves as the boolean indicator, 0 if not controversial or 1 if controversial, of whether or not the corresponding comment_text is actually controversial/toxic.

3.1.1 Example Data

A sample of the data, the header of our test data can be found in the Appendix at the end of this report.

3.1.2 Dataset Sizing

Dataset	Size
Train	127612
Validation	16026
Test	15933

3.1.3 Class Balancing

Label	Size
0 (Non-controversial)	114624
1 (Controversial)	12988

The dataset was a representative sample of real-world comments and thus the majority of comments were non-controversial; we downsampled the data to balance classes.

3.2 Evaluation Metric

We are utilizing weighted F1 score in our binary classification task. F1 score is calculated by obtaining the harmonic mean of precision and recall, where precision is the percentage of examples that were predicted positive that were correct = true positive/ (true positive + false positive); and recall is the percentage of examples that were predicted positive of all that were actually positive = true positive/ (true positive + false negative). We utilized the weighted average of F1 between the two classes in order to obtain a metric to help account for the large difference in the size of classes in our set.

F1 Score= $\frac{2*(p*r)}{p+r}$ where p is precision and r is recall.

precision = $\frac{TP}{TP+FP}$

recall = $\frac{TP}{TP+FN}$

The paper we choose to use to reference our baseline utilized the F1 score as their main evaluation metric. It was chosen because of the practicality of the weighted metric when dealing with imbalanced classes (Khieu & Narwal, 2019).

3.3 Simple Baseline

The majority class baseline for this task is classifying all the text as non-toxic (label of 0). This naive approach achieves a weighted F1 score between classes of 0.85. This relatively high F1 score is due to the unbalanced classes in the test set. These classes were left unbalanced as the test set should be an accurate representation of the distribution of data seen in the population, which in this case, the majority of internet comments are non-controversial.

4 Experimental Results

4.1 Published baseline

The paper for our published baseline, *Detecting and Classifying Toxic Comments* (Khieu and Narwal, 2019), looked into Multilayer Perceptrons (MLP), Long Short-Term Memory Networks (LSTM), Convolutional Neural Networks (CNN), and both character-level and word-level granularity in binary (toxic or non-toxic) and multi-label (non-toxic or obscene, threat, insult, etc) classification

tasks. The authors found that the LSTM with word-level granularity was the most successful in binary classification. The authors padded/clipped the input to standardize input length and used a softmax activation function for binary classification. The optimal LSTM used for binary classification used a 3 layer LSTM with 32 units of output at each layer. They additionally utilized the Adam optimizer to compute cross-entropy loss over 5 epochs of training.

With this model, the authors obtained a test accuracy of 0.899, test precision of 0.925, and test F1 score of 0.886 (though the authors used a 60-20-20 train-dev-test split as opposed to our 80-10-10 split). The results obtained in the paper are (almost) directly comparable to the results we obtained with our baseline model because we used the same toxic comments dataset. With our baseline model, however, we obtain a weighted 0.89 test F1 score which can likely be attributed to the higher amount of training data due to the differences in how the data was split across sets.

4.2 Extensions

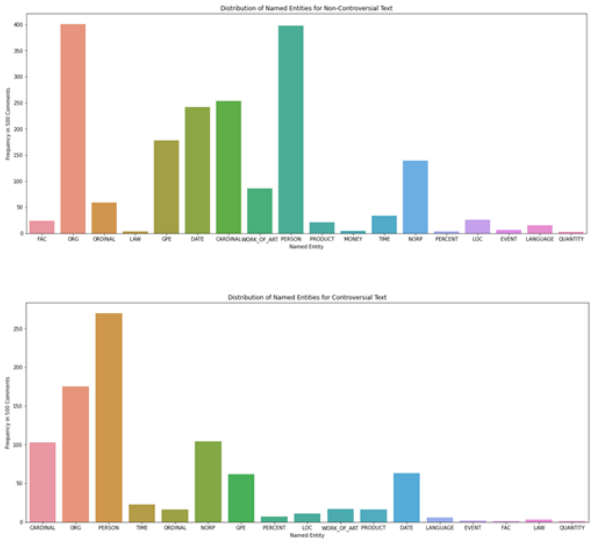
4.2.1 Modifying LSTM

The first extension we had was aimed at improving the baseline model. While the LSTM used for the published baseline had an improvement over the majority classification, there was room for improvement and fine tuning in this model. The original 5 epochs were leading to overfitting to the data, and it was determined that 2 epochs was the ideal number to minimize validation loss. It was further determined through iterative testing that a reduction in LSTM layer size to 2 layers. We additionally experimented with GlobalMaxPooling and GlobalAveragePooling for the 1 dimensional representation, where it was determined that GlobalMaxPool after the LSTM layers additionally increased performance.

After settling on the above layers for our improved model, we then iteratively determined the ideal hyper-parameters to utilize in our model. We determined that LSTM outputs of 128 and 64 were ideal for the two layers respectively, with dropout of .15 between embedding and LSTM layer 1, dropout of .2 between LSTM layers and dropout of .15 between the GlobalMaxPool and the final Dense layer. Additionally the embedding size was increased to 256 and the max feature size for tokenizing the data was increased to 80000. With these improvements, we were able to improve our weighted F1 score to 0.94, a relatively significant improvement of .05 for F1 while maintaining the overall LSTM structure of our model.

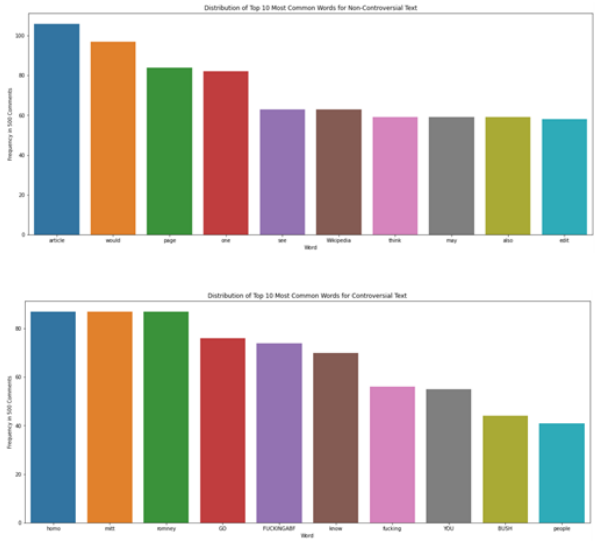
4.2.2 NER - Named Entity Recognition

Our other extension for the project deals with named entity recognition. We wanted to see which named entities were more prevalent in toxic text and which named entities were more prevalent in non-toxic text. This extension is important because as controversy is a very social construct, we can see which topics in society are the most controversial. For this section, we leveraged the pre-trained spaCy NER model. In our analysis, we determined the distribution of named entities for controversial and non-controversial text from a sample of 500 comments in our test data:



We see that whereas the non-controversial text has about an equal amount of ORG and PERSON tags, the PERSON tag dominates the controversial text distribution. We should also note that although the ORG tag is the second most common for the controversial text, the spaCy classifier exhibited a lot of errors involving labeling capitalized obscene words as organizations, which are quite prevalent in these comments. Overall, the non-controversial text appears to have a slightly more even distribution which indicates more neutrality in the comments. Taking into account the errors with ORG tags in the controversial text, this verifies the hypotheses we made during our EDA - that a great deal of controversial comments would call out the names of specific people. We additionally determined the top 10 words found in non-controversial and controversial text for our

sample (with stopwords removed):



From our analysis, we see that the non-controversial text is often composed of more neutral and also general words that do not generate significant sentiment. Specifically, we see words like "article", "page", and "think". On the other hand, the controversial text tends to make targeted attacks at individuals and based on the period at which the data was collected, we see names like "mitt romney" and "BUSH". Unsurprisingly, we also see a lot of obscene language as well as capitalized words like "GO" and "YOU" which likely indicates an angrily-written comment.

4.3 Error Analysis

By changing the parameters of our model, we were able to improve our results for both controversial and non-controversial text. Specifically, we saw an increase from 0.93 to 0.95 for the f1-score for non-controversial text and an increase from 0.61 to 0.67 for controversial text. Upon further analysis, we took a look at comments that were classified correctly by our extension and incorrectly by the baseline. Here is a small sample:

comment_text	contro
'Ha I call bluff c.'	1
'person life take as'	1
'More problem user ALR User ALR...'	0
'suck You suck Jpgordon even...'	1

We made 2 main observations about the set of comments that this sample was contained in. First from a qualitative standpoint, the comments labeled as controversial are much more subtle and tend not to include blatant obscenities or attacks. For instance, the full lemmatized and cleaned comment in the fourth row (labeled as controversial in the dataset) is:

"suck You suck Jpgordon even admins checkusers find IPs block long period time press reset button modem within 5 minute would get new IP able vandalize As result I hope admins realize blocking useless That indeed I always I get blocked And save time yet another checkuser IP I using 174 91 92 166 case figured I previously used IP 174 91 97 28 see http en wikipedia org w index php title Special Log block page User 3A174 91 97 28 blocked 3 month yesterday useless block quite easy circumvent"

Besides, perhaps, the beginning of the comment, it is much more debatable as to whether or not this is controversial text and this holds for the large majority of the set that we considered. Second, we also observed that these comments were either very short or very long with few of medium length. Overall, this indicates that the extension model with the new LSTM parameters is much more effective at identifying subtleties in controversial text - something that could be applied to identifying microaggressions - as well as correctly classifying comments on either extreme of length.

5 Conclusion

In short, we were able to create an LSTM model to classify toxic comments. This model had a weighted F1 score of 0.94 which is extremely promising as it is an improvement by 0.05 from the baseline model, a simple 3 layered LSTM. Based off the metrics in the studied papers, this binary classification task likely has state-of-the-art performance at an F1 score of about .96-.97, and thus were relatively close to the optimal of what can be currently achieved. It is difficult to determine exact state-of-the-art performance as the Kaggle competition focuses on the accuracy of the toxic comments into categorizations and does not report any other metrics. Another extension we had allowed us to explore named entities in controversial text, comparing them to the named entities in non controversial text.

Acknowledgements

We would like to thank Joongwon Kim for being our great project mentor, providing helpful guidance throughout the duration of the project.

References

Khieu, K., Narwal, N. (2019) "Detecting and Classifying Toxic Comments."

A Appendices

Example Data

index	id	comment_text	contro
0	000f35deef84dc4a	‘There’s no need to apologize. A. . . ’	0
1	0016e01b742b8da3	‘Notability of Rurika Ka. . . ’	0
2	00b984b355cc9754	’I’m focusing on doing science...’	0
3	0a19319ca119d890	‘He’s not retired, he was just useless.’	1