Detecting Offensive Language Using Machine Learning

By:

Group: Harmonious Python (HarPy)

Ashton Brasol, Ben Graf, Jason Louwagie

University of Texas at San Antonio

May 7th, 2021

**Abstract**

The objective of this project was to develop a model that was best able to detect offensive language in a tweet. This model was to reference the words in a tweet to an offensive word lexicon and accurately annotate the tweet. There were three classifications the model was to use to annotate the tweets: Not Offensive (NOT), Targeted Insult (TIN), and Untargeted Insult (UNT). Models that were tested where RandomForest (Average set to Micro), RandomForest (Average set to Macro), LinearSVC (Average set to Micro), LinearSVC (Average set to Macro), and Multi-Layer Perceptron (MLP). Of the tested models, RandomForest (Average set to micro) proved to be the most accurate, which was selected to be the final model.

**Approach**

For each model tested, we split the train.tsv dataset into 8743 training observations (80%) and 2119 validation observations (20%).

In our experimentation, we used three algorithms: Support Vector Classifier (SVC), RandomForest (RF), and Multi-Layer Perceptron (MLP) and three forms of vectorizing: CountVectorizer (Count), TfidfVectorizer (TF-IDF), and none at all (None). SelectKBest was used to narrow down the best lexicon and morphological features in our models (SKBest). When testing our models, we alternated between using a variety of features, going as far as shortening a list of our features (Best12), and not at all.

The hyperparameters that were used when testing these models were C values [.01,.1,10] for Support Vector Classifier, n\_estimators [100,200,300] for RandomForest, and hidden\_layer\_sizes [(5,2)] for the Multi-Layer Perceptron.

Many different features were used when testing our models. For SVC, we used unigrams (1,1) and bigrams (1,2). We also used lexicon and morphological features in our tests to see how this may affect the accuracy of our models. These features consisted of elements such as “number of letters”, “Number of !”, “proportion of capital letters” and “count of bad words”, to name a few. This approach was inspired by (Cècillon et al., 2019) where morphological features were used with language features to detect abusive language using an SVM classifier to distinguish abusive language with higher accuracy. We hoped to take a similar approach in increasing our final model’s accuracy between our three classifications.

**Model Selection**

Early on, we tested with our average set as f1\_micro and f1\_macro to be the scores used for tuning and found the results virtually identical throughout the models. After finding the best parameters for each setup, we then predicted on the validation set to generate micro F1 and macro F1 scores for comparison between the models. A summary of our results for a variety of configurations is below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Vectorizer** | **Lexicon/Morphology Features?** | **Validation Micro F1** | **Validation Macro F1** |
| SVC | Count | No | 0.7268 | 0.4487 |
| RF | Count | No | 0.7348 | 0.4699 |
| SVC | Count | Yes | 0.7324 | 0.4385 |
| RF | Count | Yes | 0.7263 | 0.4347 |
| SVC | TF-IDF | No | 0.7282 | 0.4815 |
| ***RF*** | ***TF-IDF*** | ***No*** | ***0.7381*** | ***0.4876*** |
| SVC | TF-IDF | Yes | 0.6947 | 0.3818 |
| RF | TF-IDF | Yes | 0.7272 | 0.4335 |
| MLP (15,5) | TF-IDF | Yes | 0.6583 | 0.4745 |
| SVC, SKBest | TF-IDF | No | 0.7253 | 0.4443 |
| SVC, SKBest | TF-IDF | Yes | 0.7244 | 0.4357 |
| SVC | None | Yes | 0.6758 | 0.3219 |
| SVC | Count | Best 12 | 0.6829 | 0.3255 |
| RF | Count | Best 12 | 0.7272 | 0.4292 |

Ultimately, the best-performing model by both metrics was the RandomForest classifier using the TF-IDF vectorizer and no other features. The ideal parameters for the vectorizer were to include only unigrams, require at least 5 appearances of a word for inclusion as a feature, and using English stop words. The RandomForest included 200 estimators and scored 0.7381 in micro F1 and 0.4876 in macro F1. Its accuracy was also 0.7381.

**Error Analysis**



Our final model correctly predicted the classification of a tweet 1,564 times out of the 2,119 total tweets. Overall, this model had an accuracy rate of 73.80%. However, if we begin to break down each component, we can see that the accuracy rate fluctuates between classifications. We can observe that we correctly identified 1,308 out of the 1,409 total possible non-targeted tweets for an accuracy of 92.83%. For targeted insults, we identified 249 out of 617 for an accuracy of 40.35%. Lastly, if we looked at the untargeted insults, where we can see that we only identified 7 of the 93 for an accuracy of 7.53%. To understand the instances where our model failed to accurately predict the type of tweet it was presented with, we must understand that there are six possible failure types. If the tweet was originally annotated as Not Offensive (NOT), the model may have predicted that it was a Targeted Insult (TIN) or an Untargeted insult (UNT) instead. A TIN may have been identified as either a NOT or an UNT. Lastly, an UNT may have been identified as a NOT or a TIN. In many cases, context was the leading cause of misidentification, impacting multiple error types and tweets. There were also instances where there may have been some annotation errors in the underlying dataset. Here are some examples of tweets that were misclassified:

|  |  |
| --- | --- |
| **Error Type** | **Tweet** |
| Annotated as **Not Offensive**  Predicted as **Targeted Insult** | @USER @USER ߤߏ‍♂️ if he is gay oh well ߘߘ‚ he be havin mfs HOT! Lol I like how he plays |
| Annotated as **Not Offensive**  Predicted as **Untargeted Insult** | @USER Bitter Barack. That says it all |
| Annotated as **Targeted Insult**  Predicted as **Not Offensive** | @USER @USER So Alyssa you saying he’s guilty? What ever happened to innocent before guilty? And if he is innocent would you at least tweet an apology? Shut up and answer the tweet. |
| Annotated as **Targeted Insult**  Predicted as **Untargeted Insult** | @USER @USER @USER You are full of shit Sweetie"" |
| Annotated as **Untargeted Insult**  Precited as **Not Offensive** | ANTI-ANTIFA IS BALLS |
| Annotated as **Untargeted Insult**  Predicted as **Targeted Insult** | @USER And when they met he looked at her and said I'm in charge in you're my bitch. Even though he knew that she is a Domme and was there in that capacity. He tried to control the situation and take over and she had to shut him down and get the hell out of there so she didn't get hurt. |

The first tweet contains an abbreviation of a profane word (‘mfs’) and a word that can be interpreted as an insult if used in a negative context (‘gay’). This particular tweet could be considered offensive by simply looking at language usage, but the context suggests otherwise. The second tweet is similar but appears to be an error on the side of the annotator. The word ‘bitter’ may not stand out as profane or offensive. However, in this instance, ‘Bitter Barack’ is clearly a targeted insult. The model appears to have been able to identify that the tweet is an insult despite it not containing obviously offensive parlance, but it was not able to pick up the target. The third tweet continues to showcase the weakness of context. Since there are no blatantly offensive words, the model failed to pick up the insult in this tweet when ‘Shut up’ is being used in a negative context. Similarly, the model successfully identified the profanity in the fourth example tweet but failed to identify that it was targeted due to the prose being used. The final two tweets share the same underlying weakness. In both, the model is misclassifying the context in direct relation to the words being analyzed. ‘Balls’ can be a profane word, depending on context. Similarly, the last tweet may contain harsh language and directional language, but the context does not indicate that the speaker is targeting a specific person. When we are strictly observing the words in the tweets, the model has done a fantastic job at annotating the data. However, the model has not been trained to take context or idiom into consideration.

**Conclusion**

Throughout this project, we attempted to develop the best model to annotate whether a tweet is Not Offensive, a Targeted Insult, or Untargeted Insult. We found our best model by testing multiple models, applying different parameters, and creating morphological features in an effort to increase the models’ F-Score. We experienced limitations in the form of the Multi-Layer Perceptron algorithm as it did not cooperate well with our code. There were challenges getting it working in a pipeline, and that limited our ability to test it extensively. Additionally, we were surprised to find that adding the lexicon and morphology features generally lowered F1 scores, which is antithetical to the intent of creating them in the first place. Of the models test, RandomForest (average as micro) is the best model with an F-Score of .7381.

References

Cécillon N, Labatut V, Dufour R and Linarès G (2019) Abusive Language Detection in Online Conversations by Combining Content- and Graph-Based Features. *Front. Big Data* 2:8. doi: 10.3389/fdata.2019.00008