Offensive Language Detection Machine Learning Challenge

IS6713 - Rowdy Code Runners
Lily He
Richard Tarbell
Jenna Wallace

1 First Steps

Our first step was to load the training and test data files and append the tweets and labels to empty lists. We ran a quick check to make sure the data loaded correctly and then converted the lists to NumPy arrays.

Our next step was to explore the training data. We used a pipeline and GridSearchCV to vectorize and test different parameters. We included parameters for ngrams, the minimum times a word would be present, and whether to include stop words. We also included SKB to test the best number of features to include, and included different C values for the classifier. We created a test split (80/20) and fit the model.

- Used a pipeline and GridSearchCV to vectorize and test different parameters.

```
{{’vec__ngram_range’:[(1,1),(1,2)],
’vec__min_df’:(1,2,4,5),
’vec__stop_words’:[’english’,’None’],
’skb__k’:[10,500, 1000, 5000,’all’],
’clf__C’:[0.01, 0.1,1, 10, 100, 1000]}
```

- Fit the model on xtrain
- Predict on xtest
- Best macro came from best params:

```
{’clf__C’: 1,
’skb__k’: ’all’,
’vec__min_df’: 1,
’vec__ngram_range’: (1, 2),
’vec__stop_words’: ’english’}

Original f1 macro score: 0.4840923367208
Original f1 micro: 0.7258
Original accuracy: 0.7258140632373761
Precision: 0.5673
Recall: 0.4720
```

With this as a starting point, we then considered which features could be added.

2 Feature engineering

Target Word feature: We defined a function to count “you” pronouns (“you, your, you’re”) that might indicate targeted offense. This could be specific to the targeted insult class (TIN).

Positive and negative words features: We defined functions to count the number of positive words and negative words in a tweet.

Each feature was created as a list and then converted to a separate Numpy array.

3 Models and Parameters Tested and Selected

We vectorized the tweets using the initial best parameters (min_df’: 1, ’ngram_range’: (1, 2), ’stop_words’: ’english)’. After that, we used hstack to append the engineered features and ran a train_test_split, with a test size of .2.

We ran a LinearSVC classifier, with test parameters of ’C’:[0.01, 0.1, 1,10] and a crossfold validation of 5. We also tested with a cross fold validation of 3, 5, 10, but CV of 5 yielded the best result. After using predict on the classifier to check the prediction power, the initial model of you words and positive and negative word features yielded these results:

```
F1 macro score: 0.4927324732094609
F1 micro: 0.7263
Accuracy: 0.7262859839546956
Precision: 0.5767
Recall: 0.4794
```

4 Error Analysis

We conducted an error analysis by printing a number of tweets and the ground truth and prediction labels to see if there was consistency in the false positives and false negatives. Also ran this to get a sample of each of the ground truth labels in order to see where the consistent issues might be. This showed that the model was labeling tweets containing a clearly offensive word as not offensive, so needed to add a feature that would catch that.

5 Additional Features Tried

More Target Word Features: We defined three new target words functions to count pronouns that might indicate targeted offense: male targets include “he, him, his”, female targets included “she, her, hers”, and group/nonbinary targets included “they, them, their”.

Offensive Word feature: We defined a classifier using an offensive words lexicon. To do this, we created a class “OffensiveClassifier()” which defined two functions. The first to count the number of offensive words and return a count. The second function identifies if an offensive word is present, returning 1 if present or 0 if not. This was in order to test if weighting the number of offensive words might impact classifying a tweet with a single offensive word as the NOT class.

6 Next Steps

We manually ran different models to check for best engineered features:

- all target group counts features
- you target count feature
- you target count + offensive word count features
- you target count + offensive present
- you target count + offensive word count + pos/neg word count
- all target pronoun features + offensive word count + pos/neg word count
- all target groups + offensive count + offensive present + and pos/neg lexicon. It is a lower score (than above) so including offensive present feature does not help
- all target words + offensive count + offensive present + negative word count - this score was lower than using both pos/neg words, so positive words must help distinguish NOT class.

We tried different classifiers - LinearSVC, SVC and Random Forest Classifier. In all cases, LinearSVC produced the best results.

7 Final Model

In addition to vectorized features, the final model includes offensive word count, all target pronoun features, and both positive and negative word counts. After running SelectKBest with the parameters of ’:[10,500,1000, 5000,’all’]. The “all” was considered the best parameter.

```
train_test_split  test_size=.2
svc = LinearSVC()
parameters = {’C’:[0.01, 0.1, 1,10]}
clf = GridSearchCV(svc, parameters, cv
    =5, scoring = "f1_macro")
```

The evaluation metrics of this model are:

```
F1 macro score: 0.5062340278713133
F1 micro: 0.7277
Accuracy: 0.7277017461066541
Precision: 0.5975
Recall: 0.4894
```

8 Error Analysis on Test set of Training Data

After fitting the training dataset and running the predictions, we wrote code to loop over the train/test dataset to print out a number of tweets and compare the ground truth class to the predicted class. We also used this code to create an additional cell where we could specify a ground truth class so that we could analyze any issues with a particular class.

We added code to count the number of false positives (when prediction was TIN or UNT when ground truth is NOT) and false negatives (when prediction was NOT when ground truth is TIN or UNT.

```
Total Tweets: 2119
False Positive: 160
False Negative: 363
```

False positives labeled as TIN or UNT when ground truth is NOT False negatives of NOT when ground truth is TIN or UNT

Examples of false positives:

```
Tweet: @USER Someone should’ve Taken"
    this piece of shit to a volcano.
Ground-Truth Class: TIN
Prediction: NOT
```

```
Tweet: @USER Besides Jax’s mom and maybe
     Ope he is hands down my favorite
     h e s  like the only good person on
    the show
Ground-Truth Class: TIN
Prediction: NOT
```

```
Tweet: @USER @USER Da fuck is going on
    people?   There’s the men’s room and
     the women’s room  Pick one and
    stick w it
```

```
Ground-Truth Class: TIN
Prediction: NOT
```

Examples of false negatives:

```
Tweet: @USER Buy more icecream!!!
Ground-Truth Class: NOT
Prediction: TIN
```

```
Tweet: #MAGA @USER Sing like no one is
    listening  Love like you've never
    been hurt  Vote GOP when no one is
    watching  And don't listen to
    Liberals' dirt URL
Ground-Truth Class: NOT
Prediction: TIN
```

```
Tweet: @USER Been a Willie fan since
    before most of you were born....LOVE
     that he is holding a rally with
    Beto.... Exactly WHICH fans are
    furious?  Could you give some
    specifics? URL
Ground-Truth Class: UNT
Prediction: NOT
```

9 Running Test file and Appending Labels

Once we settled on our final model, we ran the test data through all the same procedures as the training in order to get a similar sized matrix with all the same parameters (vectorized features, Lexicon-Classifier, Target Words Features, Positive/Negative words.). As with the training data, the hstack command from the SciPy library was used to get the data from the test file into a sparse matrix which was used to predict the labels. In order to get the labels ready to output to a new file, the Twitter ID, original tweet, and the new label were stacked together using np.dstack and then reshaped into an appropriate sized matrix. The resulting matrix was output to a tab-delimited file 'output.tsv'

10 Error Analysis on Test Data

In looking at the first ten tweets, the labels are 90% correct.

```
Tweet: @USER Nancy Lee Grahn You Are
    Awesome! I have been a fan since
    Santa Barbara!! Alex Davis also
    Rocks!!!!! Thank you !!!
Prediction: NOT
Label correct?: CORRECT
```

```
Tweet: @USER She is a Skrull. Enemy of
    The Kree. The Kree are who gave
    Carol her powers and whose uniform
    she is wearing in the first few
    moments of the trailer.
Prediction: TIN
Label correct?: CORRECT
```

```
Tweet: @USER @USER @USER @USER @USER
    @USER @USER Except you kind of are
    when it comes to gun control
Prediction: NOT
Label correct?: CORRECT
```

```
Tweet: @USER @USER @USER You are so
    beautiful
Prediction: NOT
Label correct?: CORRECT
```

```
Tweet: @USER This is what happens when
    liberals get in control
Prediction: NOT
Label correct?: CORRECT (Intention
    unknown)
```

```
Tweet: @USER @USER Daniels said her job
    does not reflect her character,
    really. She is a cheap, sleezy porn
    lap dancer. The is no high road" to
    take in that type of job"
Prediction: TIN
Label correct?: CORRECT
```

```
Tweet: @USER No longer on guard, Marie
    smiles warmly. Merci. I love when I
    am compared to my father. I look up
    to him in many ways." She looks down
     for a moment and then back up. "He
    is also my sire. Unless you already
    figured that out." She laughs softly
    .   Marie beams at the question. +"
Prediction: NOT
Label correct?: CORRECT
```

```
Tweet: @USER Gun control is  omportant.
    It should not be left to NRA to
    regulate it. It does not care about
    the lifes that are waisted along the
     path of its enrichment.
Prediction: NOT
Label correct?: CORRECT
```

```
Tweet: @USER Antifa girl of the month
    centrefold!
Prediction: NOT
Label correct?: CORRECT
```

```
Tweet: @USER @USER @USER @USER Tweet is
    directed at him.  I can care less
    about whatever it is he is
    blathering about.
Prediction: NOT
Label correct?: INCORRECT (Should be TIN
    )
```

Examples of false positives:

```
Tweet: @USER @USER @USER Ha ha yes fuck
    yes
Prediction: UNT
Note: Uses offensive language but is not
     an insult
```

```
Tweet: @USER  Y o u re  welcome! Yo! @USER
     and I were holding up a casual
    tournament watching you. Good shit!
Prediction: UNT
Note: Uses offensive language but is not
     an insult
```

Examples of false negatives

```
Tweet: @USER @USER  Y o u r e  in
    embarrassment to the Republican
    Party! Glad you took the cowardly
    way out and quit!
Prediction: NOT
Note: Should be TIN (Targeted insult)
```

```
Tweet: @USER The far left antifa are the
     real fash .cowards that hide behind
     masks and attack anyone who has a
    diff opinion they even confront not
    only the old and the vulnerable but
    children as well
Prediction: NOT
Note: Should be UNT (untargeted insult)
```

11 Conclusion

A consideration is the accuracy of the ground truth labels. There are several examples of where the ground truth label does not fit the tweet, so this would impact the accuracy of the model. Some challenges of this project potentially had to deal with our overall knowledge of text classification. While we may not agree with all of the annotators labels for some tweets, we know there are multiple things that factor into the appropriate label which neither the annotator nor us saw. For instance, we don't know the context in which some things were said (for example, what happened in the news on that date), what was said in the tweet they're replying to, or even how the tweeter implied the tweet to be read.

There are also challenges with the complexity of the English language, such as where the meaning of one word could mean something entirely different in someone else's daily life. Without knowing these things or how to apply them in our process, we were limited on the complexity and potential accuracy of our labels.

However, even though we may have had limitations due to the fact we are not yet experts in the field, we are confident our model is simple and effective in labeling the given data accurately.