AP3 Write-Up

April 28, 2024

Model Choice and Selection

Our choice to use logistic regression and try to improve upon it stems from the fact that this model is one of the more interpretable models. For instance, by knowing the coefficients associated with each feature (word or n-gram), we can gain a better understanding of which words contribute most to the sentiment classification. We also know that logistic regression can serve as a baseline model for more complex approaches. In other words, it provides a simple and understandable benchmark against which to compare the performance of improved versions of the model. While logistic regression has its advantages, we simultaneously acknowledge its limitations. For instance, this model may struggle with capturing complex nonlinear relationships in the data, which could limit its performance in tasks where sentiment analysis requires understanding subtle nuances in language. Yet, logistic regression remains a viable and still useful approach in scenarios where interpretability and efficiency are prioritized which is why we have chosen it as our baseline model.

Feature Selection Process and Model Improvement

When first running the logistic regression model on the baseline classifier, the highest accuracy we achieved on our development dataset (dev.txt) was 0.51, or 51%. To improve on this, our group decided to incorporate the following features into the model:

- Bag of Words (BoW)
    - Although this feature was already provided in the skeleton code, it is still a useful feature to incorporate in our logistic regression model because of its simplicity and suitability for ambiguous data. Binary BoW representation is very simple as it consists of creating a vocabulary of terms found in the text and using a binary vector to indicate each document's presence or absence of each word in the vocabulary. Moreover, the feature space in many document classification tasks can be extremely high-dimensional and sparse, but binary BoW is particularly useful for text classification problems because it efficiently handles sparsity and enables efficient storage and processing.
- Afinn
    - The Afinn feature creates text classification features based on the Afinn vocabulary, which is used for sentiment analysis tasks, such as categorizing a document as optimistic, pessimistic, or neutral. The sentiment of each word is represented by its score, which makes it clear whether the term has a neutral, positive, or negative connotation; this makes it particularly effective for determining the general tone of the text. Additionally, this will enable the model to capture each word's contribution to the overall sentiment by assigning a score. This granularity makes it easier to comprehend how certain terms affect the classification.
- VADER Sentiment Analyzer
    - The VADER Sentiment Analyzer generates sentiment scores for every word in a given text, similar to Afinn, but it allows for more sensitivity towards punctuation and capitalization. As a result, this enhances VADER's ability to interpret nuances in text

that convey varying degrees of mood or emotion, thus allowing for a more nuanced analysis. The feature is a result of tokenizing the text into words, and then checking to see if that specific word is already in the feature dictionary, which helps avoid recalculating scores. By assigning a sentiment score to each word, we can then aggregate these values to determine the overall sentiment of the document.

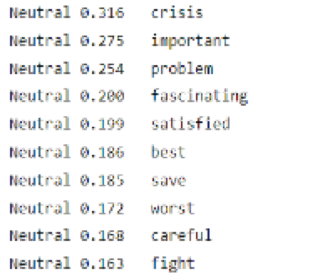
However, despite implementing the above 3 features and incorporating them into the model, the accuracy of the development set was noticeably lower – from our initial baseline accuracy of 51% to 46%. This could be attributed to multicollinearity, which is when two or more of the features/independent variables are directly correlated, specifically between the sentiment scores outputted from Afinn and VADER. As a result, our group decided to test out all combinations of features, in order to determine which ones would yield the highest development (validation) accuracy. We concluded that the best features to use are BoW and Afinn, and in the end, we saw our test set accuracy to be 53%.

In spite of our efforts, there was still a lack of improvement in our accuracy. This motivated our group to switch from word-level features (i.e. putting each word into a feature dictionary) to document-level features (taking each document/political speech and assigning it a value for each feature). The notebook that is named word_level.ipynb contains the code and logistic regression classifier our group built using word-level features, and the notebook that is named document_level.ipynb contains the model using document-level features.

We decided to continue using Afinn and VADER, but instead of applying it as a feature dictionary of words, we directly applied it to each document as a whole; by doing so, we are able to capture the text's emotional connotation, namely its polarity and intensity. Furthermore, we decided to utilize the lexical diversity of the text as a feature, which is calculated by dividing the total number of words in the text by the number of unique words. Texts with higher diversity may be more formal or expressive, whereas writings with reduced diversity may be repetitious or simplistic, and diverse language might be more prevalent in well-crafted arguments or expressive viewpoints, which could correlate with strongly held positive or negative viewpoints. Lastly, TextBlob was used as it is able to output a polarity score based on the direction of the text's sentiment. This feature is especially helpful for sentiment analysis because it offers a simple quantitative way to categorize text according to sentiment.

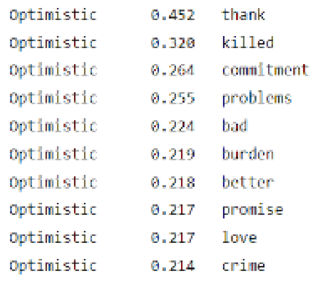
With these features and a little bit of data preprocessing, we were able to build a logistic regression classifier, using the document-level features, and achieve a test accuracy of 61%. While there is plentiful room for improvement, there is a marginal uptick in the accuracy of the document_level feature model compared to the word_level feature model. Moreover, the confidence interval of the document_level feature model is higher, thus indicating greater reliability in its predictions across different samples of the test data.

Lastly, one thing that is interesting is the weights/coefficients of the logistic regression classifier for the word_level feature model.



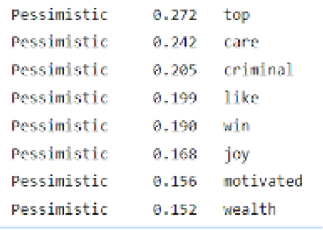
```
Neutral 0.316   crisis
Neutral 0.275   important
Neutral 0.254   problem
Neutral 0.200   fascinating
Neutral 0.199   satisfied
Neutral 0.186   best
Neutral 0.185   save
Neutral 0.172   worst
Neutral 0.168   careful
Neutral 0.163   fight
```

Neutral Sentiment: Positive weights for terms like "crisis," "problem," "fascinating," and "careful" show that, in the context of this model, they are strongly associated with neutral sentiment. Certain terms, like "crisis" and "problem," are generally associated with negativity; nevertheless, in a neutral environment, they could be discussed in a factual or non-emotional context, which could account for their designation as neutral.



```
Optimistic      0.452   thank
Optimistic      0.320   killed
Optimistic      0.264   commitment
Optimistic      0.255   problems
Optimistic      0.224   bad
Optimistic      0.219   burden
Optimistic      0.218   better
Optimistic      0.217   promise
Optimistic      0.217   love
Optimistic      0.214   crime
```

Optimistic Sentiment: It is interesting to see words, such as "killed," "problems," "bad," and "crime" have positive weights associated with "optimism." The classification of these normally pessimistic terms as optimistic could point to them being discussed in the context of a resolution or justice being delivered. Furthermore, they could also be spoken in a manner to incite feelings of hope and "better things to come."



```
Pessimistic     0.272   top
Pessimistic     0.242   care
Pessimistic     0.205   criminal
Pessimistic     0.199   like
Pessimistic     0.190   win
Pessimistic     0.168   joy
Pessimistic     0.156   motivated
Pessimistic     0.152   wealth
```

Pessimistic Sentiment: It is surprising to see how words, such as "happiness," are associated with pessimistic sentiment, along with "criminal" and "struggle." Its classification as "pessimistic" could be indicative of contexts in which these terms are used in conversations concerning the presence or absence of certain positive things.

The presence of typically positive words in pessimistic categories or negative words in optimistic categories suggest words may change their sentiment depending on the context in which they are used. Further training is required, such as by examining the particular instances from the training set, where these terms occur, as it could be useful in comprehending their application and setting. With more time, our group would have liked to add more contextual features and word embeddings to better capture context.

Performance of Model

i) Confusion Matrix

Below, we visualize our models' performance using a confusion matrix. By comparing the distribution of true labels to predicted labels, this type of graphic helps us answer the following questions by classifying labels across different sentiment categories:

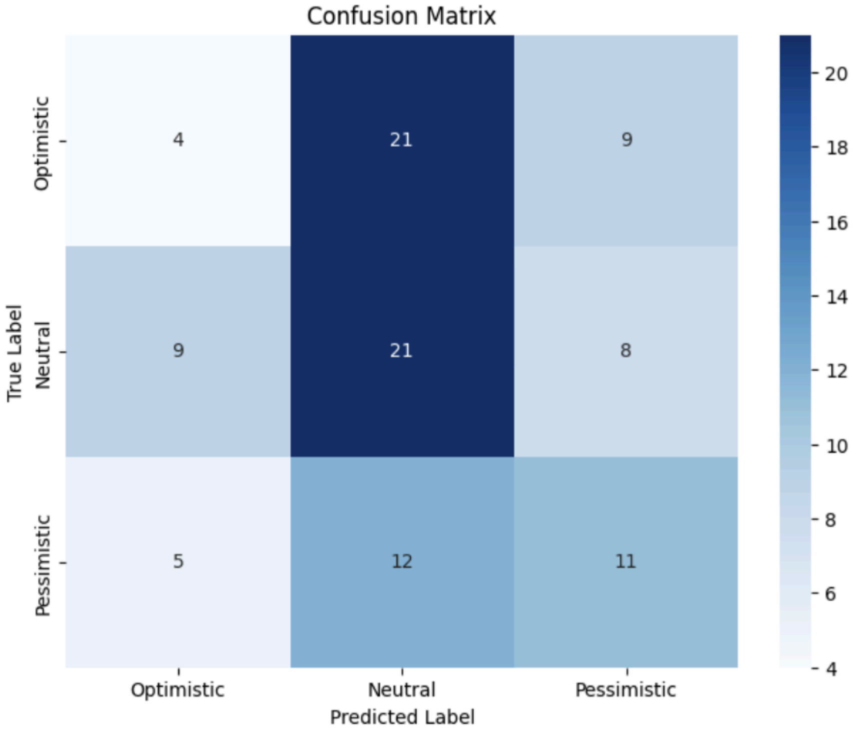
- *What labels are often mistaken for each other?*
  - Using the AFINN dictionary and Bag-of-Words feature, the confusion matrix reveals that many labels are often mistaken for another. Compared to others, the optimism label is most commonly mistaken by the classifier for neutral. However, it also still tends to

mistake pessimistic texts for neutral or optimistic, neutral for optimistic or pessimistic, and optimistic for neutral or pessimistic, although all at lower frequencies.

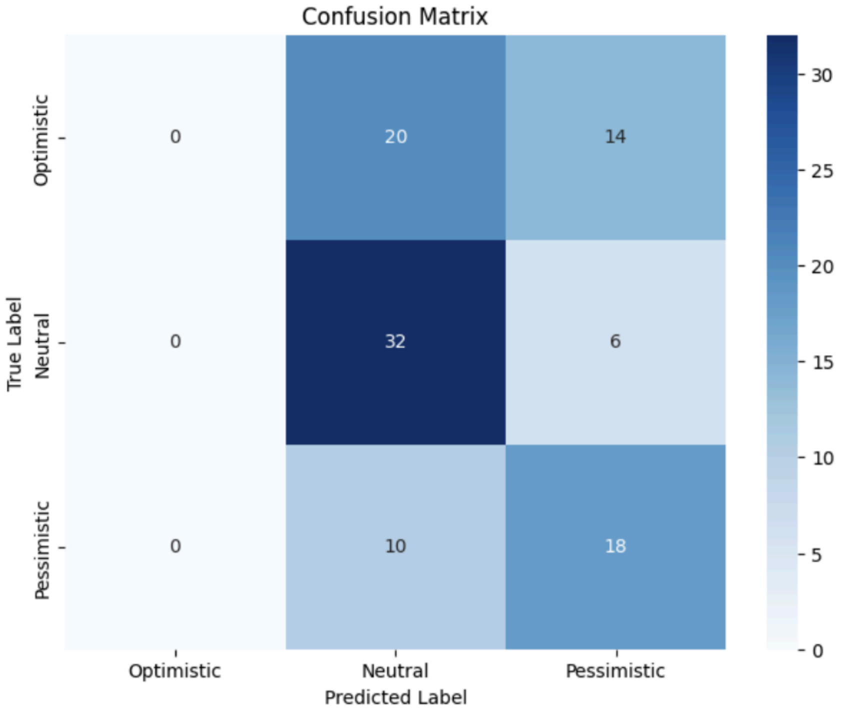
    - In our baseline logistic regression model, we first note that the classifier has difficulty detecting optimism. More specifically, it is unable to predict any labels as such. The confusion matrix shows that there were 32 instances of optimistic sentiment in the true labels, but all of them were incorrectly classified by the logistic regression classifier as either neutral or pessimistic. Yet, there are several reasons for which this might occur. For example, it is possible that the features used to train the model were not well-suited for distinguishing optimistic sentiment from neutral or pessimistic sentiment. Further, it is also possible that the classifier is overfitting to the training data, or that there is a class imbalance in the data that is affecting the classifier's performance. This is supported by the fact that the classifier shows high training accuracy but low test accuracy. We aimed to solve this problem by exploring different features and dictionaries which, in fact, proved to work as we saw with the AFINN classifier.

- *Is one label extremely prevalent? How could this impact the model you developed?*
    - Among predicted labels in the AFINN and BoW classifier, the neutral sentiment is most prevalent, and among the true labels that same dominant category is also neutral. This indicates that the classifier is able to correctly predict and label the majority class most of the time in order to maintain its majority status.
    - Similarly, the baseline logistic regression model had a dominant true label of neutral and a majority predicted label of neutral as well. Despite all its misclassifications, it still manages to catch most of the neutral labels and categorize them as such.
    - However, for all of our models, having a high prevalence of neutral labels will have an advantage and a drawback. On one hand, the model will be more adept and skilled at correctly identifying a neutral sentiment; however, it is possible that this could impact our classifiers’ abilities to identify other sentiments such as optimism or pessimism.

BoW + AFINN Classification: Confusion Matrix

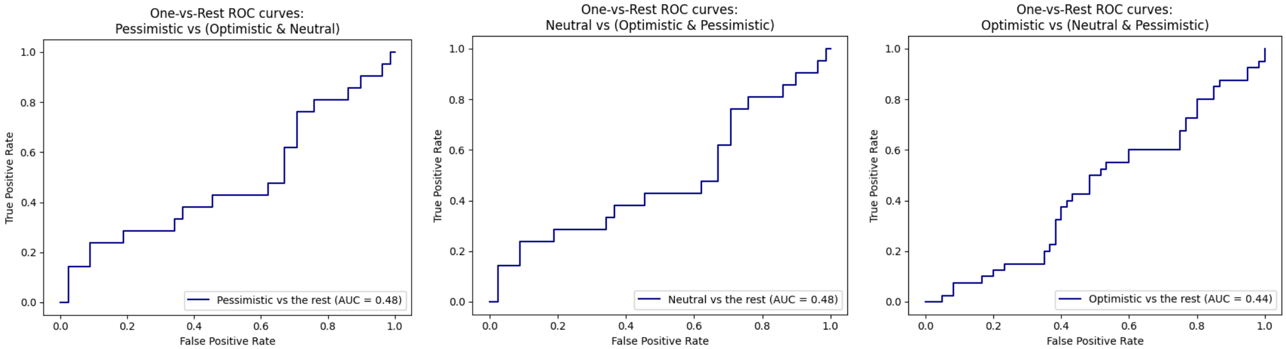


(Baseline) Logistic Regression Classification: Confusion Matrix



ii) ROC Curve

We then used multi-class ROC curves to visualize our models' performance in another light. By visualizing the ROC curve, we can visualize the tradeoff between distinguishing between the True Positive Rate and the False Positive Rate. The AUC is a statistic which generalizes how good a model is at distinguishing between the classes; for this analysis, the BoW + AFINN classifier shows a sub .5 AUC, but shows a broader trend of the ROC curve and this dataset. Models trained on the presented dataset were better at classifying "Neutral" political passages.



- *What kind of systematic mistakes does your model make?*
  - Overall, the model performance of the BoW + AFINN model is very poor. AUC's of good models are closer to 1, AUC's closer to .50 mean that the model is no better than randomly choosing between two labels.
  - Obviously the task of classification is much harder utilizing three classification categories, but we anticipate there being two reasons as to why the model underperforms. This model cannot discriminate between classes. Using Manning 2011 to diagnose the issue, we believe that the reason that our project is difficult to accomplish is because great examples of optimistic, pessimistic, and neutral speeches are difficult to distinguish.
  - The pieces of text that were given to the annotators to annotate were long, and annotators had to observe the overall connotation of the piece of text, therefore weighting the connotation of the individual words in a paragraph less overall.
  - Another interesting aspect of our model is that the model is better at predicting the "Neutral" class among positive and negative classes. We believe that this phenomenon is partly due to the number of neutral data points in the dataset itself. There were 124 texts classified as neutral, and 214 texts classified as optimistic. We know that smaller classes can lead to greater overfitting of the model, leading to a model that is more likely to predict a neutral class.

Reflection of Human Annotation

In AP2, we performed numerous human annotations on the exploration and evaluation data we collected. In our exploration phase, the entire group annotated the same few sets of documents in order to come up with guidelines. This led us to have discussions around the ambiguity of question-marked statements, negative topics spoken about in a positive light, and derogatory remarks about another candidate to bolster oneself. In the end, our guidelines captured our general consensus around each of the nuanced cases. The group then chose to have Dana and Carrie producing two separate independent annotations on all documents and Nikhil acting as an adjudicator for all of them. While all three of us would agree that the process of 500 annotations is arduous and perhaps tedious at times, it was overall an informative and intriguing experience. To learn more about the different ways two people can interpret the same text and what specific syntax or sentence structure we use to determine a sentiment also helped offer insightful information into the features we chose in AP3.

Weaknesses and Shortcomings

Given the vast amount of algorithms and features which still exist, it is possible that our group could have explored more options and discovered new ways to improve our model. One example of this might be looking into external libraries that staff had not suggested. Or, similarly, we could have used more features to incorporate. Another route we could have taken is to have chosen an entirely different baseline model. From what we have learned in this class, we know that NLP models such as BERT can capture more contextual information from the entire input text, leading to better understanding of semantics and context. Perhaps this might have offered a better starting test accuracy and also different approaches to improving the model.