Movie Sentiment Analysis

Data Splitting

Simply training on one training set and testing on one validation set is a feasible way to evaluate hyper parameters and classification accuracy. However, it is no question that such an approach is also more subject to variance. In order to have a more stable metric, we perform 5-fold cross validation of 80/20 splits in order to tune are hyper parameters and to evaluate our classification accuracy. By averaging across 5 folds, this not only provides a more accurate proxy for the true test error, but it also reduces the effect of an unfavorable split that could happen due to randomization, where the label distributions are heavily skewed in favor of one label. If we were to only split once, we would have to remove some randomization from our split by arbitrarily ensuring that the both the training and test data have a good proportion of positive and negative texts. Thus, 5-fold cross-validation solves both of these problems: it maintains the element of randomness whilst ensuring a healthy split of labels.

Intuitively, it would make sense that normalizing and cleaning the data should improve the overall effectiveness of the end model. This includes lemmatizing/stemming and removing stop words. However, this was found not to be the case. We will provide reasons later on as to why performing such normalizing techniques is detrimental to accuracy.

How the Data is Cleaned

```
"..<br /><br />Watch the scene where Harilal hears about his father's death.No
dialogues,No screaming,but a speechless shot by Khanna.Its one of the finest
scenes ever shot in the history of Cinema<br /><br />Gandhi,My Father is not
at all exciting cinema but yes,its excellent cinema and a must watch.Brilliant
Attempt.."
```

The first step is to decide how to clean the data. If you look at the example text above, it is clear that there are lots of punctuation and undesirable characters. It is undeniable that punctuation, in most cases, has a negative effect on accuracy and should be removed. We do this by using the TfidfVectorizer, which both removes all such punctuation and lowercases all words by default.

Why Some Stop Words are Kept, and Why Bigrams are Used

The next step of cleaning data is to account for stop words. Indeed, treating stop words as individual entities is detrimental to accuracy. However, when stop words are coalesced as a member of a bigram or trigram, this new group plays a role in determining the tone of the review more than their individual constructs, which ultimately helps in predicting overall sentiment. For instance, take a look at the following review below:

```
The only halfway decent thing is the rubber outfit of the creature(which is
glimpsed in such quick flashes that you don\'t really have time to see how
phony it really is).
```

The word highlighted in blue ("only"), according to some packages, is categorized as a stop word. Without taking into account its context, the word "only", when standing by itself, has very little meaning. However, if we were to take the following two words into account, "halfway" + decent", we can see that the overall tone of this review is negative. In fact, "only halfway decent" is much more negative in tone than is "halfway decent" and hence, the trigram is a better indicator of movie sentiment than is its bigram counterpart. This is one example of how some stop words, when coupled with other words, play an important role in classification reviews. With this in mind, these kinds of stop words should be kept in the vocabulary. For this same reason, bigrams should also be used, as words coupled together are ultimately more meaningful than words viewed as individual entities. For instance, using "decent" as a unigram is misleading. As an individual entity, "decent" possesses a positive connation, and this would be an indicator for a positive review. However, taking into account the word immediately preceding it ("halfway"), and viewing these two words collectively, as "halfway decent", it becomes clear that this bigram elicits an adverse tone, which is ultimately reflective of a negative movie view.

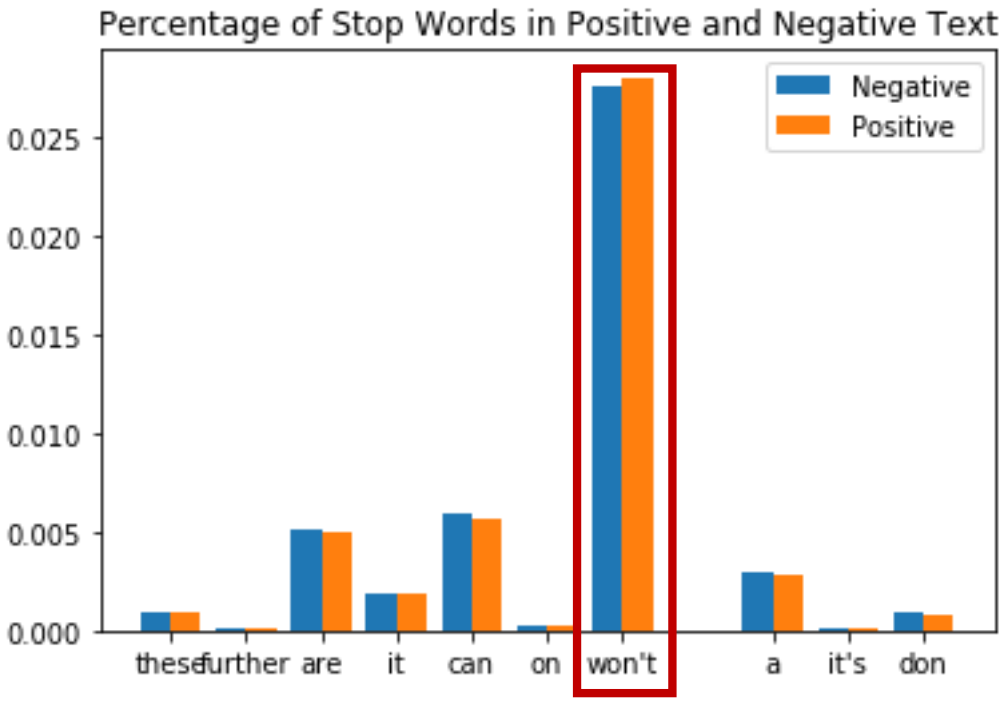
Although some stop words help in determining overall sentiment, it does not mean that all stop words are useful. In fact, many stop words are not useful, but distinguishing between a useful stop word and useless one can be tricky. To approach this issue, we develop a systematic approach to eliminating stop words. Below we describe the method and the rationale behind it:

One way to determine which stop words are more important and which ones are less important is to define a metric that will accomplish this goal. The metric we define is described as:

$$\frac{\text{Count of Stop word(i) in } \textbf{negative} \text{ reviews}}{\text{Total \# tokens negative reviews}} \qquad \frac{\text{Count of Stop\_word(i) in } \textbf{positive} \text{ reviews}}{\text{Total \# tokens positive reviews}}$$

We will use this metric to sort by the difference between the two labels. The logic behind this metric is that if there is very little difference between the same stop words belonging to their respective class label, then that particular stop word would less likely be useful for determining overall sentiment. Conversely, if the difference of this metric is large, then the stop word of interest is more likely to be useful for predicting sentiment. Naturally, one would expect a useful feature to appear with higher frequency in one class and lower frequency in another class if it is indeed an important feature. Hence, with this kind of rationale, removing stop words that appear with relatively equal frequency in both class labels should theoretically lead to an improvement in classification accuracy. For example, let us reference the graph below (***next page***). First, look at the word "won't" surrounded by a red box: the blue bar corresponds to proportion of the word "won't" in the negative text and the orange bar corresponds to the proportion of "won't" in the positive text. If the tops of these two bars are close, it means that "won't" is a word used with similar frequency in texts of either sentiment.

* It is worth noting however that in this project, we choose to use bigrams, namely because it is more computationally feasible, and trigrams yield little improvement to accuracy overall.



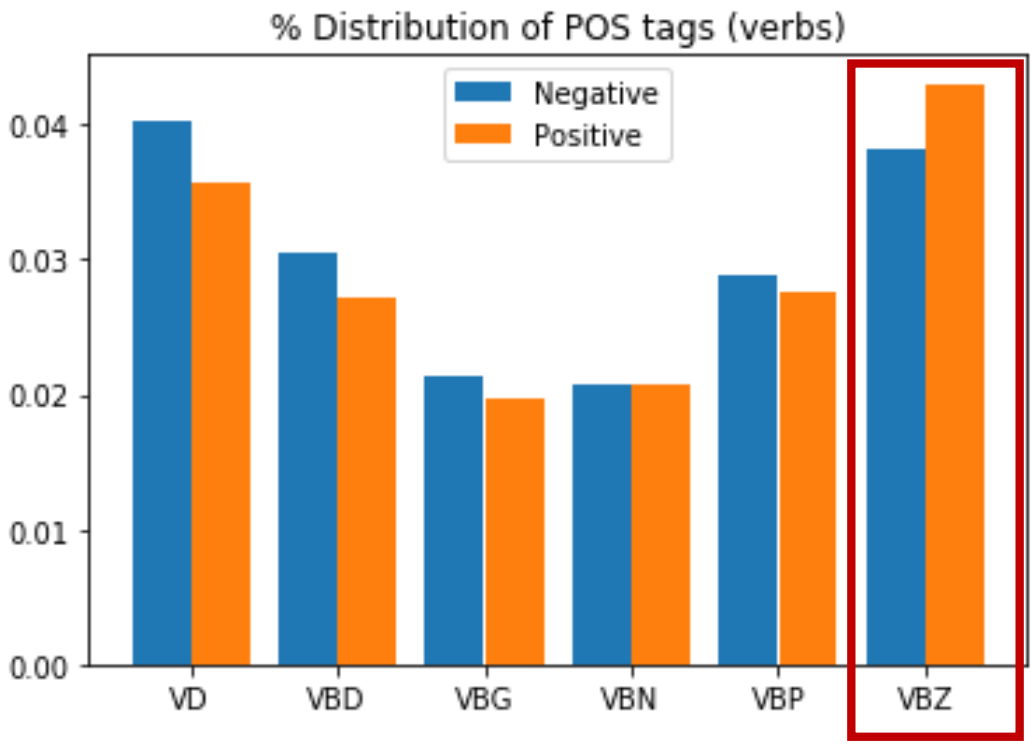
Of course, words can be used with similar frequency but used in a completely different way depending on their class labels. This is a drawback of the metric: it fails to take into account how the words are utilized. However, it provides a heuristic measure and although it is not optimal, it gets the job done.

The bar chart displays some of the top stop words that have the least difference in usage between the positive and negative labels. Removing some of these words via the metric described earlier ultimately yields better performance. More importantly, following the metric saves time. It makes it so that we do not have to go through the entire set of stop words and systematically remove them one by to see the effect each word has on the classification accuracy. The metric allows us to simply cycle through the first n words to observe the overall effect on prediction power. In this way, we are able to almost improve our cross-validation accuracy by 1%.

Why Lemmatizing and Stemming May not be a Good Idea

The next step is performing some stemming/lemmatizing. Below is the percent distribution of POS tags (verbs). To account for the size of the text, we divide the frequency of the particular word by the total number of words in the respective label to obtain the bar chart below:

$$\frac{\text{count of specific word of POS tag in text i}}{\text{total number of words in text i}} \quad \text{where (i = } \textbf{'positive'} \text{ or i = } \textbf{'negative'} \text{)}$$



It is clear that for almost every verb type, there appears to be a higher frequency of verbs across all verb categories in reviews of negative sentiment. However, only in the "VBZ" POS tag is there a higher frequency in the positive text (as shown in the red square). This difference may be small, but it seems to have a significant effect on predictive power. Indeed, lemmatizing for such singular verbs with an "-s" (VBZ verbs) at the end results in worse accuracy. In this case, verb tense does have a positive effect on classification and hence, and because this example is not the only case where the tense/suffix matters, blindly lemmatizing words is not a good approach to our problem. Due to this fact, we generalize and say that it is better not to lemmatize nor stem the words.

Why we Choose Logistic Regression

After the preprocessing and data cleaning steps, we must figure out a data representation for our problem. We choose a tdf-idf vector that includes a very large portion of the vocabulary (100,000 words to be exact). The reason we choose such a large feature space is two-fold: first, we don't really know exactly what features down the line will be useful, even if they do appear with high frequency. By arbitrarily designating a smaller feature space, such as 1000, we are eliminating many of the possible features that will help distinguish between positive and negative reviews. Thus, it is better to have a larger feature space and use regularization as a way of reducing dimensionality. More importantly, by using bigrams, the vocabulary size greatly expands. In order to ensure that our feature space includes important bigrams, it is better to have a larger feature space.

Clearly, this would result in overfitting as there are thousands of features. Because the vocabulary is so expansive, producing a systematic way to remove features one by one to reduce its size is not the best approach. It is here that logistic regression shines. In effect, we can use the L2 loss with a heavy penalty as a regularization parameter to help us select the most prominent features and zero out the less important ones.

Because of this, logistic regression is uniquely suited for this problem, as it employs feature selection (mandating a heavy penalty) as part of its kit. Utilizing other methods, such as LDA or Classification Trees could perform better with a reduced feature space, but this would require a more systematic way for feature space reduction. Finding the most important features from hundreds of thousands of features is quite a difficult task and it is for this same reason that logistic regression is so powerful.

Thus, to tune for the optimal hyper parameter "C", we perform a grid search for 5 fold cross-validation while varying the weights for C and choose the C value that yields the highest accuracy. The C value was found to be approximately 18.80. As expected, this "C" value is quite severe; setting the "C" penalty at this value does indeed reduce the feature space quite a bit by setting numerous weights very close to 0.

Overall, the accuracy we obtain on our cross-validation set hovers around 91.2%. To obtain better accuracy, many things can be done but not limited to are:

- More in-depth research into POS tags and lemmatizing/stemming for only certain words
- Using Doc2vec embedding, as this takes into account the syntactic relationship between words (not just counting the frequency * inverse-document frequency of words) and measures the similarity between documents
- Using Doc2vec embedding coupled with a Recurrent Neural Network