Ross Bollinger

Professor Nathalie Japkowicz

5 December 2018

CSC-480

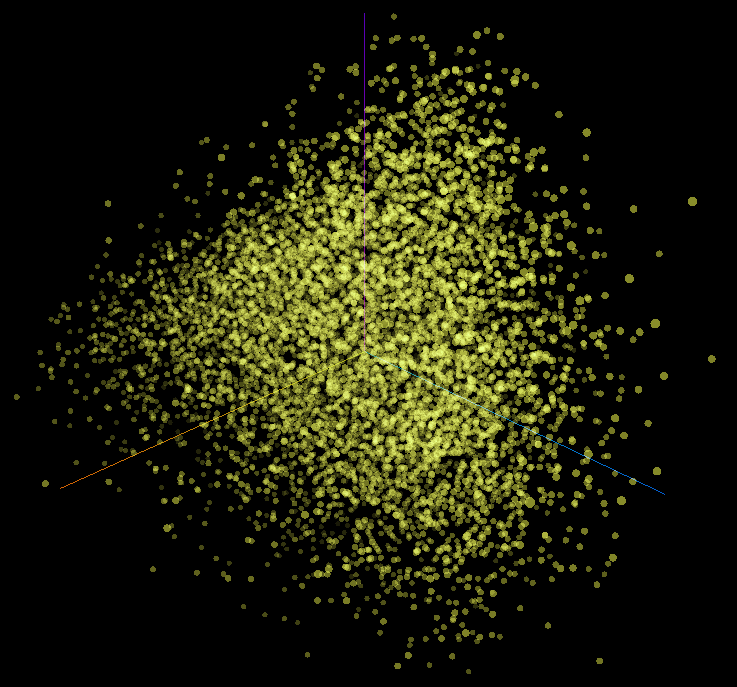
Using Word2Vec to Classify Text from English Renaissance Plays

**Introduction**

Text classification is a large facet of Natural Language Processing and a fairly self-explanatory concept. It is the idea of assigning classes to text based on the content of the text. There have been many studies surrounding this concept, with a large variety of different texts being used to test this concept and its effectiveness. One main focus in the field of text classification is word representation. This idea is centered around representing words for machine learning algorithms in order to make them easier for said algorithms to interpret and learn from. This idea stemmed from the high the dimensionality associated with text data, and it allows for less complexity in processing said data. A more recent word representation method, Word2Vec, is a word embedding system. This means that through its unique word representation, it is capable of understanding context and syntactic similarity between words.

The way that Word2Vec builds a model is quite interesting. When given a text corpus of sentences, Word2Vec builds a model in which it assigns a vector to each sentence. In doing so, it creates a sort of graph of words in which each word is a point on said graph. To understand the relationships between words, it tracks how close together some words in the graph are than others. This creates a framework of context and semantic similarity that it can apply to any text.

Below is a visual representation[1] of word2vec’s model:



There have been many previous papers that explore the efficiency of Word2Vec with other texts. Some use text from Fox News broadcasts, to famous novelists, to even social media posts, with overall promising results. However, in researching this topic, I couldn’t seem to find any experiments using theatrical plays for experimentation. This is where the goal of the paper began.

The goal of this paper is to test the effectiveness of the Word2Vec model on datasets constructed from plays written by English Renaissance Era playwrights. During the English Renaissance, a massive explosion of artistic interest occurred in England that made a great impact on the performing arts of the time. Many plays were being written during this time, and there was an accepted style and language used in writing said plays. This makes them easily discernable to us as Renaissance English plays. Works from playwrights like William Shakespeare (one of the very popular playwrights of the time and the most influential Renaissance playwright today) have become such a cultural staple that it would be difficult not to classify works of that era. This is the perfect kind of data to test Word2Vec with. The style of the time seems to be very similar with every play from that time if they are only viewed from a language perspective. However, they are often very complex in terms of plot, characters, dialogue, and structure. Word2Vec would be a great fit for this type of data because of this because it is capable of picking up on such complexities.

Specifically, I plan to test how good Word2Vec is at discerning different playwrights based on text from several of their plays. To test this, I will run a Word2Vec model through two classification algorithms commonly used for textual classification and compare the performance evaluation results to those of a Bag of Words Model. The Bag of Words model is a simple word representation model that assigns frequency counts to words and uses those counts as features for classification. This method only looks at text on a surface level because it only looks at how many times a word occurs in the text. Knowing this, I hypothesize that Word2Vec will outperform Bag of Words in all cases.

**Data Preparation**

For my data, I picked a total of 7 plays from 3 English Renaissance playwrights. The playwrights chosen are Ben Jonson, Christopher Marlowe, and John Webster. These are three playwrights who wrote in competition with Shakespeare, so I decided not to include any of Shakespeare’s works. I took the plays from Project Gutenberg[2] and manually formatted it so that only the dialogue was present in the text (no staging directions or character names denoting who is speaking). I then used NLTK’s[3] sentence tokenizer to separate the data into sentences and format them to a csv format. I then compiled them all into one massive training set with 3 classes: Jonson, Marlowe, and Webster.

**Table 1.** Statistics of Play Sentence datasets.

|  |  |
| --- | --- |
| Statistics | Values |
| # of plays | 7 |
| Total # of sentences | 8,260 |
| # of Jonson plays | 2 |
| # of Jonson sentences | 3,257 |
| # of Marlowe plays | 3 |
| # of Marlowe sentences | 2,397 |
| # of Webster plays | 2 |
| # of Webster sentences | 2,608 |
| Total # of words | 110,263 |
| Avg. # of words in a sentence | 13.35 |
| Min. # of words in a sentence | 1 |
| Max. # of words in a sentence | 174 |

**Table 1.** shows the statistics of the training data set that I used. In processing the data, I removed all punctuation except for “.”,”?”, and “!” so as to keep the sentences as sentences. I then tokenized the data into separate sentences with separate words (basically an array of arrays of words) and fed it into the model.

**Experimentation**

Using the Gensim[4] Python library and code from a relatively similar experiment[5], I created a Word2Vec model of size 200 from the training set. I then ran this model through two classifiers using the SciKitLearn[6] Python library and the previously mentioned code. The two classifiers I decided to go with were Logistical Regression and a Linear SVM, since these are both commonly used algorithms for text classification. I recorded the classification accuracy, precision, recall, and F1-score values as the performance evaluation metrics for the classifiers.

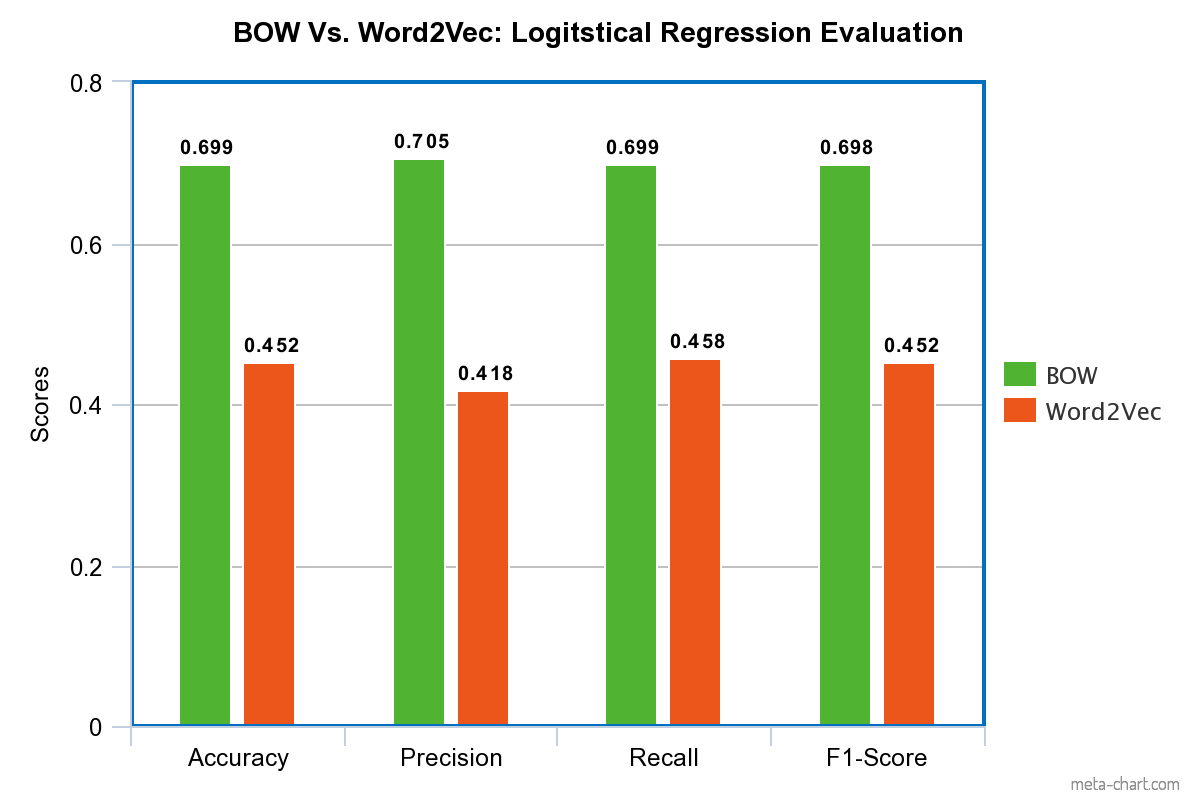
I constructed a Bag of Words model from the training data using the Orange Visual Data Mining Tool[7]. I then ran the model through the same two classifiers used with the Word2Vec model and recorded the same values for performance evaluation. I then compared the results from the two models.

In both cases, I used 10-fold cross validation in order to test the representation models and obtain their performance evaluation results.

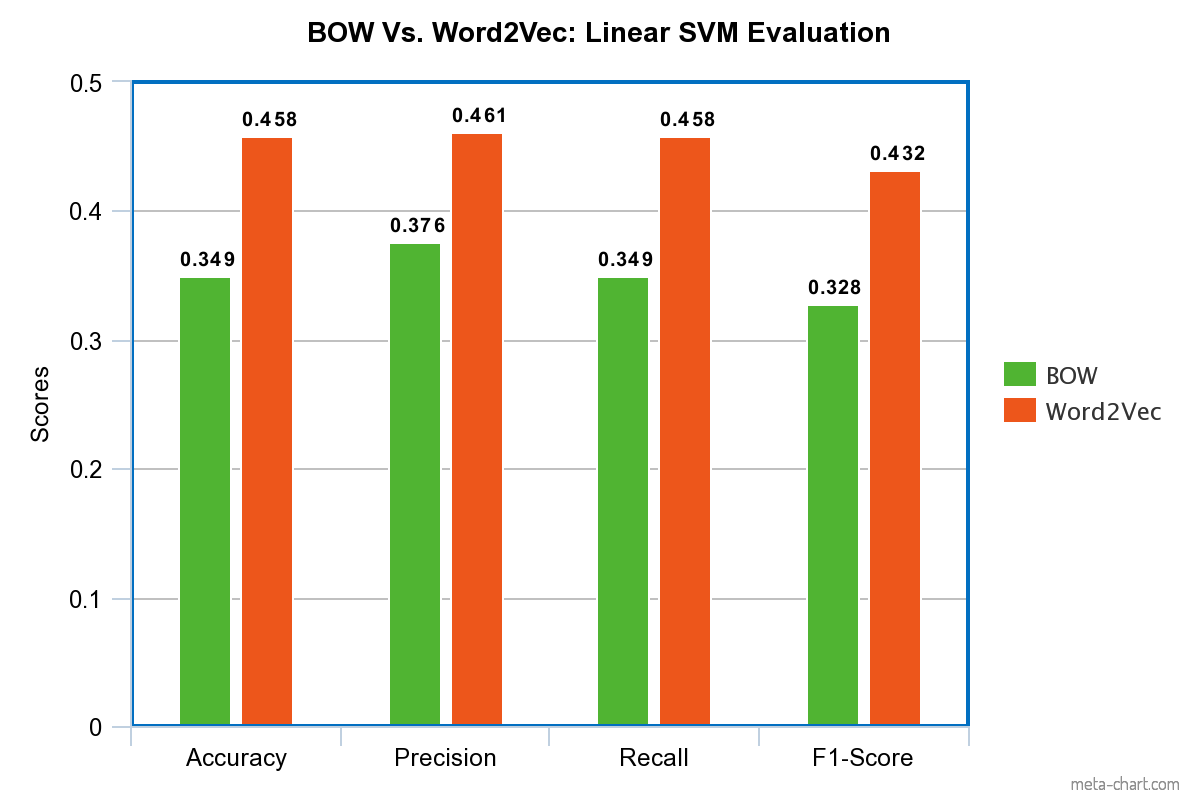
**Results**

**Fig 1.** and **Fig 2.** graph the results of the experimentation.

**Fig 1.**



**Fig 2.**



**Results Discussion**

The results show that the Bag of Words model outperformed Word2Vec using the Logistical Regression algorithm. However, Word2Vec seems to have outperformed Bag of Words using the Linear SVM algorithm. Thus, my hypothesis was only halfway proven correct. There is a variety of reasons for these results.

1. Logistic Regression may not be the best at handling the complexity that the Word2Vec model creates. There were probably too many more features to keep track of with this model, creating a less efficient Logistic Regression model for the classifier to work from.
2. There could be too many outliers in the word2vec model, and Logistic Regression models are known to have worse results when there are more outliers in a data set. SVMs are much better at dealing with outliers, so this could
3. SVMs tend to work more graphically, similarly to how Word2Vec works, so maybe this model was more compatible with this algorithm.
4. Human error in coding and running the programs used to obtain these results in the first place. The fact that two differently coded programs were used for each representation model could also factor into this.

Whatever the reason, it is clear that the results are inconclusive. This is honestly very strange considering Logistic Regression and Linear SVM classification algorithms are supposed to perform at a relatively equal level, but not in this experiment.

**Conclusion**

In this study, I have shown the differences between two word representation models for text from English Renaissance plays and how those models can be used in different ways. In the end, my hypothesis was not fully proven correct, as Word2Vec only outperformed Bag of Words in the Linear SVM case. This, however, does create interesting paths to go forward with:

1. Further experimentation involving these two models. Maybe experiment with varying sizes of data or with more playwrights. There are multiple options for variation here.
2. Further experimentation with different classification algorithms with this model. Perhaps other classification algorithms that are commonly used for text classification, such as Naïve Bayes, could be useful for determining the efficiency of Word2Vec.
3. Since there might have been human error, further experimentation on the settings used in each program or proofreading of code used. This experiment could definitely be done using the same program instead of two separate ones, but due to time constraints, that was not exactly possible in this case.

**References**

1. <https://projector.tensorflow.org/>
2. <https://www.gutenberg.org/>
3. <https://www.nltk.org/>
4. <https://pypi.org/project/gensim/>
5. <https://www.kaggle.com/vukglisovic/classification-combining-lda-and-word2vec>
6. <https://scikit-learn.org/stable/>
7. <https://orange.biolab.si/>