Josephine Boenawan & Theresia Susanto
Prof. Zhi Li
11/5/2017

Part 3: Project Write Up & Reflection

a. Overview

For this project, we used three books from the Guttenberg Project: A Tale of Two Cities, Great Expectations, and Oliver Twist. First, we found the word frequencies for each book, and then we did a sentiment analysis to see the positive, negative, and neutral using NLTK package. We also did a text similarity analysis to see how different each text is with one another. We hoped that we can compare the three chosen books and make recommendations of which books to read through some characteristics based on the judgement of text sentiment analysis. Furthermore, it would be interesting to find how much the author's diction - word choice - change throughout the three books. A question that would be interesting to answer is whether Dicken's word choice is a reflection of his mental state.

b. Implementation

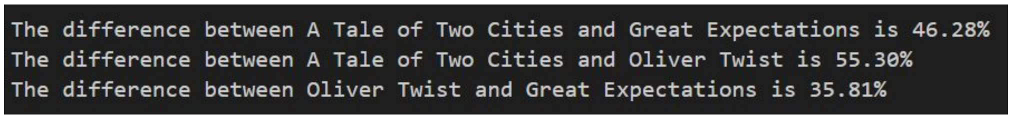
Our first step to analyze the text is to clean it of stopwords such as which, herself, has, in, it, etc. This way, it gives a cleaner data structure to give a better sentiment analysis. If we were to include the stopwords, it would result in higher neutrality, which is not what we want. Getting rid of neutral words will achieve a clearer distinctions between positive and negative words. We also removed the header and tail of the text files to rid of unnecessary words that are either not written by the author or does not relate to the storyline. The innerProduct and docDist are two functions that calculates the similarity score by using the distance of the dot product of two vectors.

One design decision that we had to make is creating a dictionary instead of a list of tuples for the word frequency. This is because dictionary is a more efficient way of storing both keys and values, although one downside is that it needs to be converted to a list of tuple in order to sort it. We had to join the list that we created into a string of text because the sentiment analysis takes in a string instead of a list. This is another design decision that could be improved so we do not have to switch between two data structures as much.

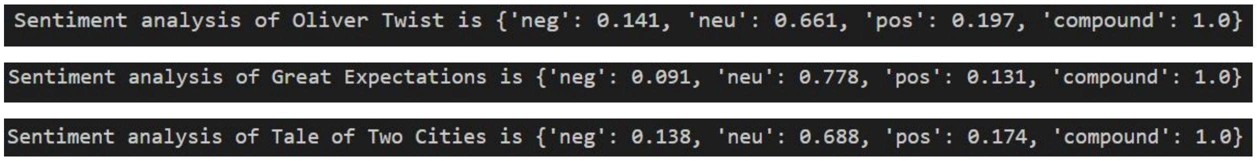
Additionally, we had to remove more stopwords because after removing the initial set of stopwords, we found that "said" was the most frequently used word in all Dickens three books. We feel that it should be removed to improve the sentiment analysis, so we did a lot of specific codes tailored to analyze text from Dicken's books.

c. Results

These are the outputs from the text analysis ran through three books:



```
The difference between A Tale of Two Cities and Great Expectations is 46.28%
The difference between A Tale of Two Cities and Oliver Twist is 55.30%
The difference between Oliver Twist and Great Expectations is 35.81%
```



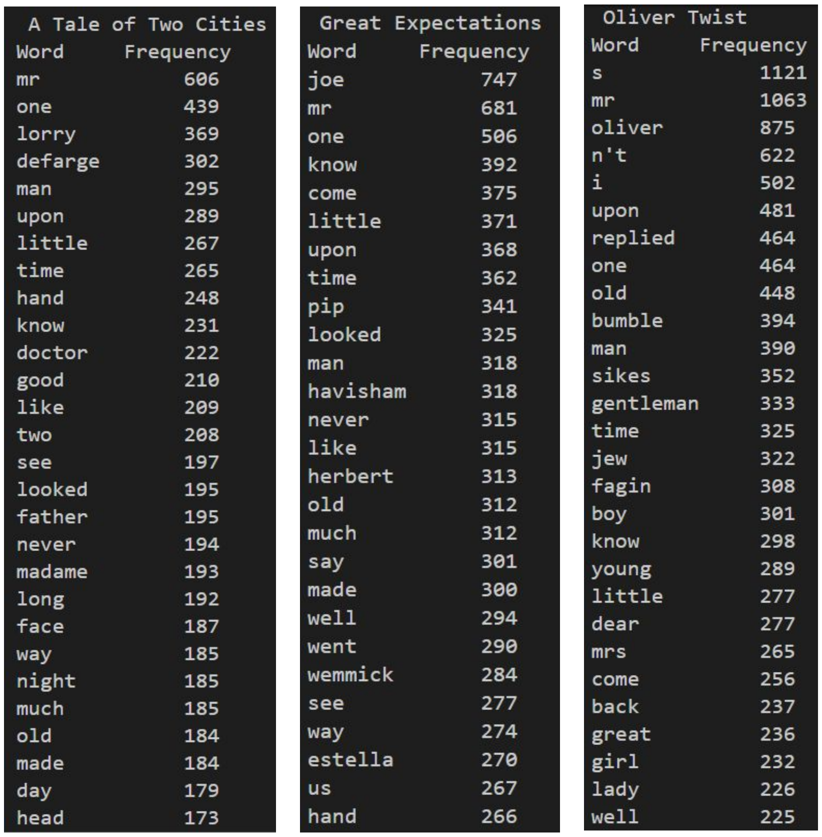
```
Sentiment analysis of Oliver Twist is {'neg': 0.141, 'neu': 0.661, 'pos': 0.197, 'compound': 1.0}
```

```
Sentiment analysis of Great Expectations is {'neg': 0.091, 'neu': 0.778, 'pos': 0.131, 'compound': 1.0}
```

```
Sentiment analysis of Tale of Two Cities is {'neg': 0.138, 'neu': 0.688, 'pos': 0.174, 'compound': 1.0}
```

Figure 1.0

Our initial goal is to determine if Dicken's books uses negative words to insinuate a darker atmosphere. Our hypothesis is that Dickens would use a lot more negative words to create a gloomy theme. However, this is proven to be false because all of Charles Dicken's books uses a lot more positive words despite different storylines (See Figure 1.0).



```
 A Tale of Two Cities
Word      Frequency
mr              606
one             439
lorry           369
defarge         302
man             295
upon            289
little          267
time            265
hand            248
know            231
doctor          222
good            210
like            209
two             208
see             197
looked          195
father          195
never           194
madame          193
long            192
face            187
way             185
night           185
much            185
old             184
made            184
day             179
head            173
```

```
 Great Expectations
Word      Frequency
joe             747
mr              681
one             506
know            392
come            375
little          371
upon            368
time            362
pip             341
looked          325
man             318
havisham        318
never           315
like            315
herbert         313
old             312
much            312
say             301
made            300
well            294
went            290
wemmick         284
see             277
way             274
estella         270
us              267
hand            266
```

```
 Oliver Twist
Word      Frequency
s              1121
mr             1063
oliver          875
n't             622
i               502
upon            481
replied         464
one             464
old             448
bumble          394
man             390
sikes           352
gentleman       333
time            325
jew             322
fagin           308
boy             301
know            298
young           289
little          277
dear            277
mrs             265
come            256
back            237
great           236
girl            232
lady            226
well            225
```

Figure 1.1

Initially, we wanted to see if there was a significant pattern of word frequency relating to the theme of the book. For example, the word "little" is used frequently in Great Expectations and A Tale of Two Cities. However, "little", "time", and "man" that are seen in both stories carry neither positive or negative sentiment. Unfortunately, the frequency of the words are not as significant as we had thought. However, we found that "night" is one of the most frequently used in a Tale of Two Cities as well as "prisoner" with 138 mentions in the text (See Figure 1.1). This proves that the most frequent words weakly correlates to the theme of the book.

The most similar of the three books is Oliver Twist and Great Expectations. Interestingly, Oliver Twist and Great Expectations is written far apart from each other. This may be because the storyline of both stories include orphans as the protagonist and follows a similar storyline. Therefore, it uses similar words to describe the character and setting. From this analysis, we know that A Tale of Two Cities is the most different from the other. From further research, we found that the time period of A Tale of Two Cities was set in 1775-1790s while both oliver Twist and Great Expectations were set in the 1800s.

In the future, we think it would be interesting to analyze a sample of famous classical texts using sentiment analysis to see if there is a pattern or correlation between using positive words as more favorable in literature than using negative words, even if the storyline has a tragic ending.

d. Reflection

Was your project appropriately scoped?

From a process point of view, it would be very helpful if there was more time and quick intermissions just like doing the final project, so we have more guidance to improve the codes such as reducing running time. Therefore, I think it would be best for us to have gone over the main idea with the Professor to see if it was feasible and interesting enough. Next, we should do sprints where we have a minimum viable product such as one function so if we run into problems, we can easily fix it and move on. However, with the time and skillsets we have, this project is appropriately scoped. We wanted to figure out the difference in diction and writing style using python, and that is what we tried to do in the amount of time we had to do outside research and skills we acquired during class time. This code needs a lot of improvement, especially with running time because it takes a long time to run all the three texts.

Did you have a good plan for unit testing?

We started out testing the codes for the text cleaning, word frequency, and sentiment analysis on just one text first just to make sure that it works. If it works, then, we test the second text before continuing to write the code for the similarity test.

How will you use what you learned going forward? What do you wish you knew before you started that would have helped you succeed?

Going forward, we could use what we learn and improve it by adding more complexity to it and testing it on different units. We learned that the similarity testing can be used for programs such as plagiarism tools and we found it really interesting, so definitely we can increase the depth of our knowledge and apply it to a different field.

Were there any issues that arose while working together, and how did you address them? What would you do differently next time?

We formed the partnership a little later in the project stream, so we had already started to design the codes individually. Fortunately, we had the same goal in mind to continue the project so what we did to integrate the two codes together is choose and pick the codes that work, and continue from there. However, we were both suck at the same problem, which was the text similarity analysis. What we did was break down the vector formula and tried to incorporate the formula using python's syntax. It was most difficult for us to understand the capabilities of nltk and so we did a lot of research through stackoverflow and other sites before continuing to construct our design.