

ASSIGNMENT 3 PROJECT WRITE UP

Project Overview

The data source that I chose to use for this project is the Guttenberg Project. I chose to make use of Aesop's Fables, a collection of fables by Aesop, an ancient Greek storyteller whose stories remains one of the most known and relevant to this day. Fable is a literary genre that features animals, are short and succinct, and always carries a moral. Knowing this, I wanted to analyze what most frequently uses. I also thought it would be very interesting to use sentiment analysis with the nltk package and cosine similarity see how similar one fable is to the next, since they all seem to be of the same length and follow the same formula with regard to plot and characterizations.

Implementation

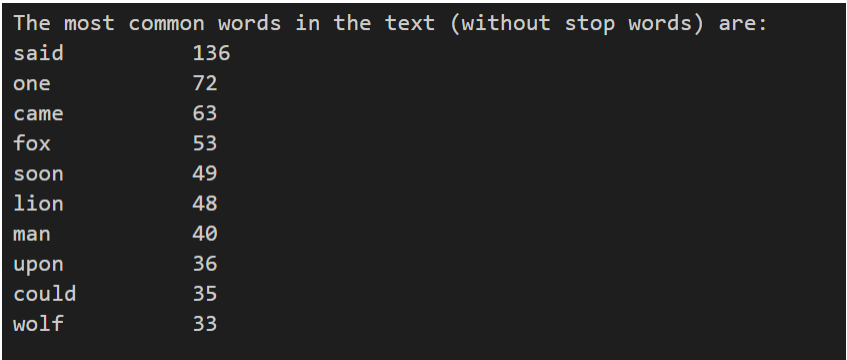
In general, I tried to make all the functions as general and flexible as possible. By that, I mean I wrote my functions to try to take in as many arguments as possible. A large part of it was because I didn't want to have to create new text files for every single fable in the collection that I want to analyze. As seen in my `extract_fable(text,start,end)` function, I modified the previous function of `skip_guttenberg_start_end(text)` which only took on one function. I kept the `skip_guttenberg_start_end` function just to distinguish between cutting off Guttenberg's header from separating individual fables, which is what `extract_fable` was written to do. Having the 'start' and 'end' as arguments allowed for me to pick out whichever section of the text I so choose.

For the most part, I chose to use dictionaries instead of lists, histograms specifically, to allow for keys and values to represent words and their frequencies, since frequency is a big part of my analysis. Dictionaries are also mutable, unlike tuples, which is why it is quite convenient to in the project. An example of its mutability being of great advantage to me is in trying to take out stop words, where items had to be deleted from the dictionary. I used the dictionary's copy function to make sure that I don't modify the original dictionary, which contains all the texts in the book, in case I would need to refer back to it.

Choosing to make use of the nltk package also proves to be doubly effective in this case. First, it provides me a ready list of stop words in English, which saves me time and effort in that I didn't have to create a list of my own. The purpose of my removing the stop words in the text was due to the abundance of it; it dominated the list of most common words in the text, so the result didn't become valuable. Second, it is also used in sentiment analysis. Its sentiment intensity analyzer module allowed natural language processing to detect positive/negative/neutral words to create a compounded score that reveals the text's sentiment.

Results

The first analysis I did is of the most common words in the text. The screencap below shows 'said' to top the list. You could also see three animals – fox, lion, wolf – in the list, with both 'fox' and 'lion' topping 'man'. This speaks accurately to how fables features personified animals. Animals appear more frequently than men and 'said' allows them to take on anthropomorphic qualities.



```
The most common words in the text (without stop words) are:
said          136
one           72
came          63
fox           53
soon          49
lion          48
man           40
upon          36
could         35
wolf          33
```

Next, I analyzed what the top 5 most common animals in the text are. I had created a list of all the names of animals on another text file and compared it with that in the Aesop text to see what animals match. When the output came out, I realized that a couple of the results are names of animals that are not really seen in the fables' titles. So I decided to run the matching analysis on the book's table of contents, which includes all the titles of the 82 fables in the collection. The results came out as I had expected. The 3rd most and the 4th most common animals in the text were not the same as those in the titles, which I find to be very interesting. It's clear that mouse

and ass appears more frequently in the writings, but dogs and serpents tend to play bigger roles; hence the placement in the titles. Later I also tallied up all the appearances of animals in the text, which gives me a total of 410 appearances out of 11890, which means animals make up on 3.45% of the entire Aesop's fables text file.

-----	-----
The 5 most common animals in the entire text:	The 5 most common animals in the Aesop's Fables' titles:
fox 53	fox 11
lion 48	lion 7
wolf 33	wolf 6
mouse 20	dog 4
ass 19	serpent 3
-----	-----

Total number of words in Aesop's Fables collection: 11890
Percentage of total words in text that are animals: 3.45%

Given all the 82 fables, I chose 3 that are Aesop's most popular fables: The Hare and the Tortoise, The Wolf in Sheep's Clothing, and The Fox and the Grapes and analyzed the sentiment of each individual fable. The results weren't what I expected. I initially thought there would be higher numbers on the negative end, but neutral looks to dominate both positive and negative. The compound results came out to be very close to each other, all not 0.2 away from each other. The compound results are interpreted by being the most positive as it gets closer to 1. That means "The Fox and the Grapes" came out to be the most positive text amongst the three.

Sentiment analysis results of The Hare and The Tortoise:
{'neg': 0.046, 'neu': 0.867, 'pos': 0.087, 'compound': 0.6908}
Sentiment analysis results of The Wolf in Sheep's Clothing
{'neg': 0.041, 'neu': 0.87, 'pos': 0.089, 'compound': 0.7579}
Setiment analysis of The Fox and the Grapes
{'neg': 0.058, 'neu': 0.824, 'pos': 0.117, 'compound': 0.8625}

Lastly, I analyzed the three fables' similarity to each other. As I head mentioned, fables generally follow the same structure. They're all very short, have the same characters (animals), and have morals as endings. The results below were a big surprise considering how alike they are in structure. However, it is worth noting that the similarity analysis programmed mostly accounts for words appeared and their frequencies. So given that each fable is only about a paragraph long, it makes sense that they would be dissimilar to each other since there are less words that intersect.

```
-----  
The Hare and The Tortoise and The Wolf in Sheep's Clothing is 0.26% similar.  
The Hare and The Tortoise and The Fox and the Grapes is 0.32% similar.  
The Fox and the Grapes and The Wolf in Sheep's Clothing is 0.42% similar.  
-----
```

Reflection

The toughest part of this project for me was to just think about what I wanted to do especially since I don't know the full capabilities of python and how I'm able to utilize it. Once I was able to pick a text I was interested in and list out what I plan to accomplish with my analysis, the assignment became a lot more doable. My plan of attack throughout the entire coding experience was to write very specific functions that does something for one thing, and then make them more generalized throughout the iteration. I think that worked really well in helping me understand what can be done with each function and what can be done with the text that I have chosen. Although it would have been beneficial to have met with Professor Li to discuss plans for the project, individual exploration on stackoverflow forums was definitely very exciting and enlightening to understand different people's logic and coding styles. I did struggle with having to download the nltk module and it took me hours to finally be able to make it run on my file. It would definitely have been helpful to have learned about downloading such packages like nltk and twython in class. Although not difficult, learning it in class could have prevented from a lot of unnecessary trial and error.