March Saper

Text Mining Mini Project Reflection

Project Overview

Louisa May Alcott's *Little Women* is a coming of age story that revolves around four main characters, each of whom can be considered an archetype. My goal was to extrapolate the adjectives used most frequently to describe each character. To do this, I examined the words surrounding each character mention.

Implementation

In order to extrapolate the adjectives used to describe each character, I chose to examine the ten words prior to a character mention. If other one or more of the other three characters were mentioned in those ten words, I did not examine the text. In this way I hoped to builds lists of words that were likely referring to or about the character I whose descriptions I was examining. I chose this method of extrapolation because it made the most sense to me, as a learner and as the writer. I wanted a way to gather likely lists of descriptive words in a simple manner. It seemed that over the course of a fairly large number of lines, inconsistencies as a result of this method would average out.

Once I had these lists of words, I used parsetree from pattern to identify the words that were (in a very loose sense, as I will explain later) adjectives. In earlier iterations of the code I chose to look at all adjectives associated with each character. In later iterations (and in the final code I submitted) I chose to look at adjectives unique to each character.

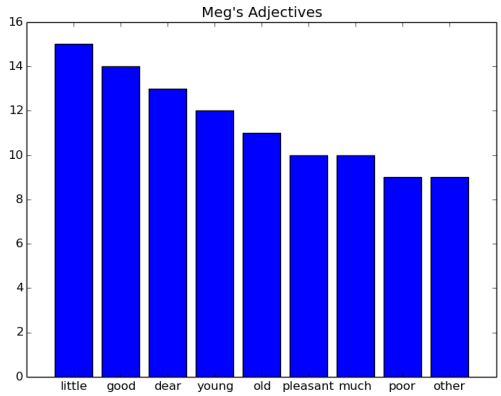
From the list of adjectives I found the most frequently used for each character. I used the decorate, sort, undecorate method from Think Python to build this, employing a dictionary for indexing purposes and tuples for their convenience as a function return value. Finally, from this information I created bar graphs of the top nine most common adjectives for each character.

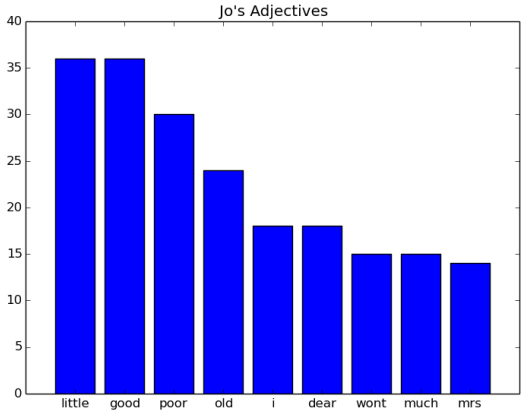
Results

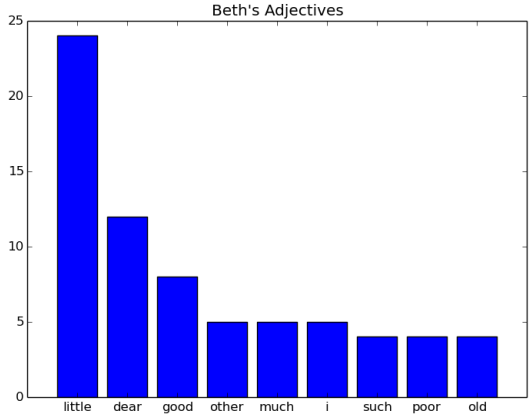
In the graphs below, one major roadblock that I encountered will be clear. Either I or parsetree from pattern malfunctioned at some point. There are definitely some words below that are not adjectives. I do not know whether this came from a misunderstanding on my part of what parsetree requires to distinguish parts of speech or whether this came from parsetree itself.

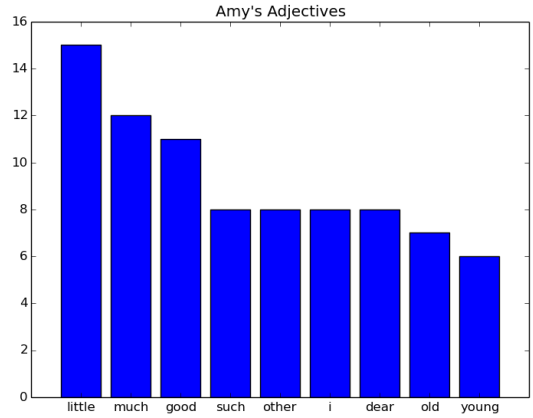
With that disclaimer, the results I found were interesting. Each character had several frequent adjectives in common. Surprisingly, for all characters "little" was used the most frequently. The adjectives "dear", "good", "poor", "young", and "old" also appeared in the top nine for all characters. In retrospect it is interesting that both "young" and "old" appear as top adjectives for all characters. This may be due to the use of "old" as an adjective of endearance. Or it may be due to the fact that Little Women spans a significant portion of life for the main characters.

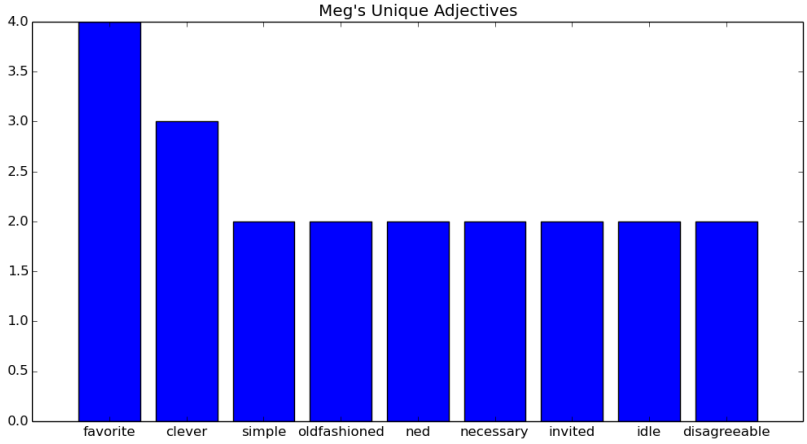
The unique adjectives fell into what I been expecting – that the adjectives used for each character would closely follow the archetype each character was assigned. What I did not expect until seeing it, though, was how infrequenty even the most common unique adjectives were used. I interpret this as a mark of a good writer. Alcott apparently used a wide and distint vocabulary when describing each of her characters. I assume this to be the mark of a good writer!

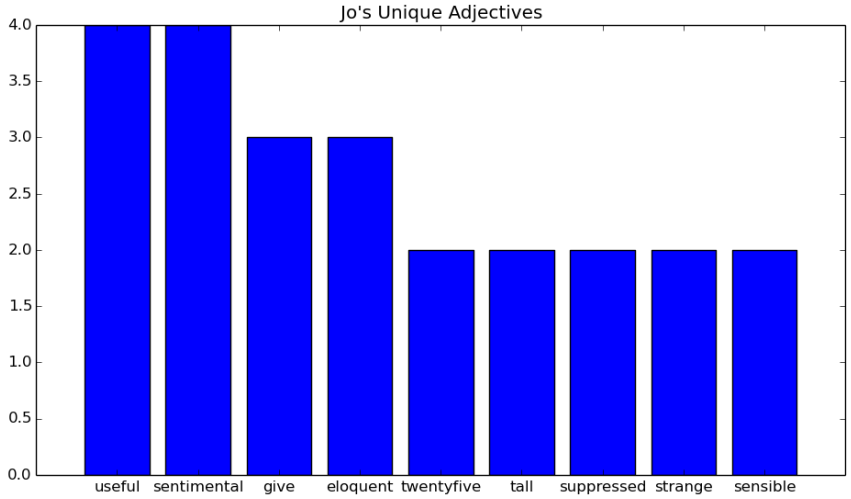


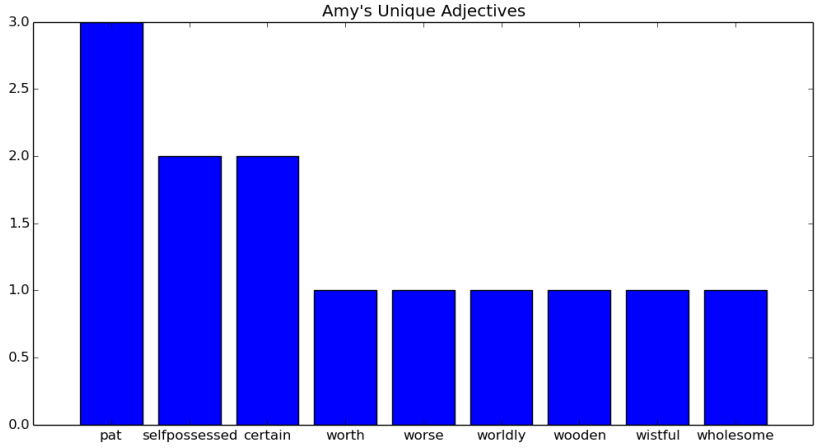


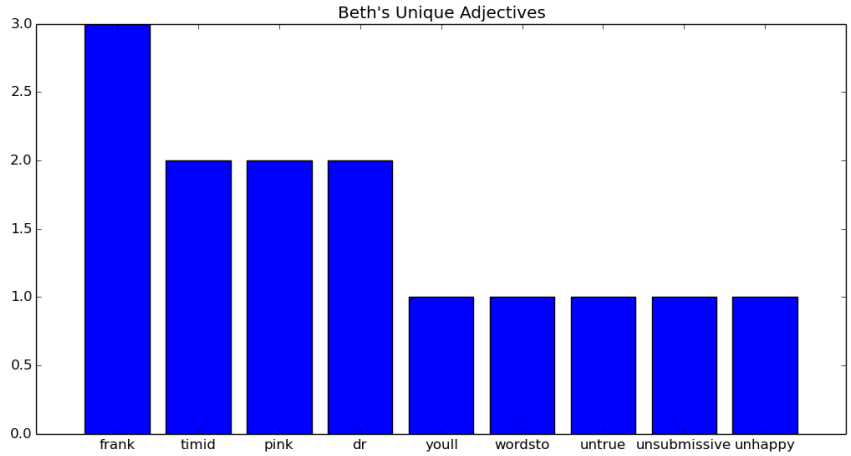







Jo's Unique Adjectives
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
useful
sentimental
give
eloquent
twentyfive
tall
suppressed
strange
sensible


Amy's Unique Adjectives
3.0
2.5
2.0
1.5
1.0
0.5
0.0
pat
selfpossessed
certain
worth
worse
worldly
wooden
wistful
wholesome


Beth's Unique Adjectives
3.0
2.5
2.0
1.5
1.0
0.5
0.0
frank
timid
pink
dr
youll
wordsto
untrue
unsubmissive
unhappy

Reflection

From a process point of view, I successfully built my program in small functions and tested each one. I feel that this was a good approach as it allowed me to be confident in how my program was behaving before I continued to the next function. It also forced me to think out how I wanted to structure the program. There are many ways of combining the operations I wanted to perform. Overall, I think the choice to use smaller function chunks was a good one. I could improve the method I use to find words that are likely descriptions. I could perhaps use a python module to examine sentence by sentence instead of by word count. I could also make my code generally more compact. Basically, I could examine the choices I made when finding likely descriptive words and ensure that I did not make any grave assumptions. I wish I had known a better way to sort out adjectives from a list of words. (Honestly, to the very best of my knowledge “I” is not an adjective.) This could have been manifest in knowing where to go to read the Pattern documentation, or in knowing what other possibilities are out there. Going forward I hope to use the web scraping skills that I have learned for future projects (though I do not know what they might yet be). I generally feel that that is a helpful skill. In addition, I feel much more confident manipulating text in lists and tuples. I also have a much clearer understanding of the nuances related to the “in” and “not in” operators.