Anne LoVerso and Raagini Rameshwar
Software Design Homework 5
March 3, 2014
Textual Analysis of Sherlock Holmes

Project Overview: What were you trying to accomplish? What was your general approach?

Our project involved a textual analysis of Sherlock Holmes novels. Our two major functions include graphically representing keyword frequency throughout the book, and sentiment analysis. We then combined these two function to look at the sentiment of each instance of the keyword search. Since we had two people working on it, we started working in parallel, with Raagini working with the sentiment analysis side and Anne working with the frequency analysis side. We each got a working program that outputted the correct graphics. Then, we worked on integrating the two sides and creating the combination function that mapped both keyword frequency and sentiment. At this point, we found that some of our approaches had diverged. In particular, we had some very similar functions written, where some of them worked for lists of sentences, and some for lists of words. At that point, we decided to consolidate our code by standardizing the inputs we were working with and making a coherent workflow. Then, we partner coded to edit our existing functions and debugged.

Implementation: How does your code work? What libraries did you use? How would someone (for instance a NINJA) run your code? What data structures (e.g. lists, dictionaries) did you use in your program and why?

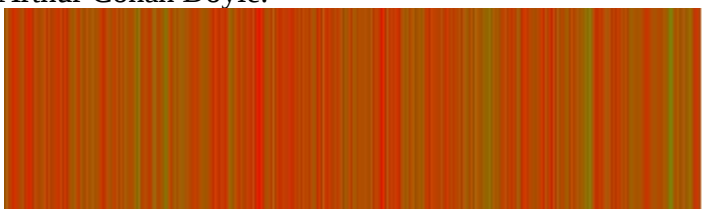
The libraries we used were pattern, Image, and re. We used pattern's sentiment function. re was used for parsing the words from the full text, and Image was used to generate the results graphics.
The general abstracted flow of our program is:

We have a series of functions that take the code through each step of the process. Then, our main() function actually calls each function in the right order and follows through in generating the three graphics, given a searchterm. The code is run by simply running the program, and the search query can be changed in the searchTerm assignment in the main() function, and the text to process can be changed in the same place. We used lists for the organization of the data. We originally had dictionaries, but found that they were unnecessary, given that our keys were all ascending integers to represent the page numbers, so the dictionary was essentially serving the purpose of a list. We stored a single page as a list of sentences, and the full text was represented as a list of sentences. When we ran the combination frequency/sentiment analysis, we stored the data as a list of tuples. This is because we needed a total of three pieces of information for each slot: a page number, the number of times the search query appeared on that page, and the average sentiment of each sentence on that page containing a search query.

For purposes of running the code, only the final_analysis.py file contains the final program. The other programs were starter code that we then combined into the final file.

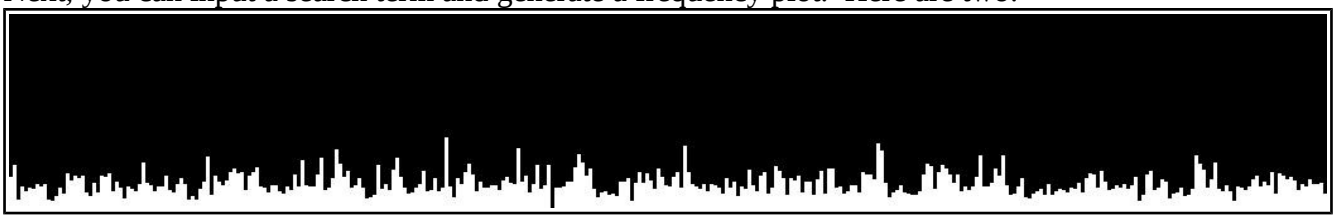
Results: If you did some text analysis, what interesting things did you find? If you created a program that does something interesting (e.g. a Markov text synthesizer), provide a few interesting examples of the program's output.

First, our program produced a graph that plotted the average sentiment per page of *The Adventures of Sherlock Holmes* by Sir Arthur Conan Doyle.

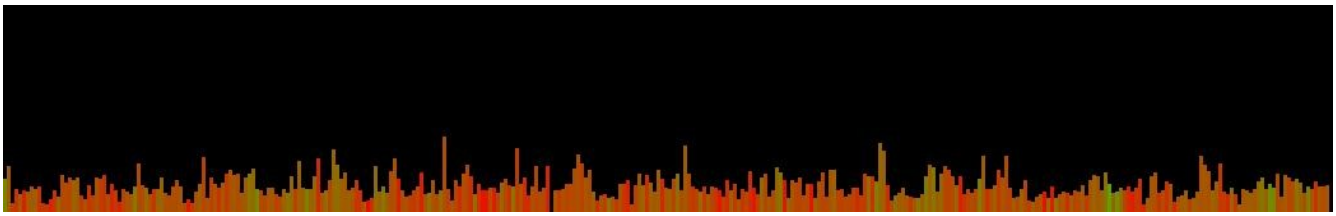


The redder the lines, the more negative the sentiment. The greener the lines, the more positive the sentiment. We neglected the subjectivity/objectivity of the sentiment analysis. As you can see, Sherlock Holmes is a pretty negative book. We think the lines of green correlate with the end of each short story, where the mystery gets solved, hence the semi-periodic habits of the lines. The x-axis represents the page that is analysed.

Next, you can input a search term and generate a frequency plot. Here are two:



The first plot shows the frequency of the word “the,” the second is for “holmes” (note the lower case search terms, as we lower-cased the entire text). The x-axis again corresponds to page number and the height of each bar is the number of times the search term appears on that page. We used these two search terms because “holmes” is an interesting and relevant metric, and “the” simply makes an interesting graphic because it appears so often.
Finally, we put the two functions together to create a combined sentiment/frequency chart



Here, the heights of the bars correspond to the frequency again, and the colors indicate the sentiment. Since each bar represents a single page, to give each bar a color we looked at the sentiments of each sentence on the page that contained the search query, and then averaged them together. The colors are

different than the originals sentiment graph because it only looks at sentences on the page that contain the word, not every sentence.

Reflection: from a process point of view, what went well? what could you improve? For instance, were there specific strategies for better coordinating on a software project with a partner? Was your project appropriately scoped? Did you have a good plan for unit testing?

Overall, we worked pretty well. There were a few rough points that could have been smoothed out. At the beginning of the project, we discussed the abstractions of what we wanted each part to do, but never went into the specifics of implementation and data structure usage. This created the confusion of Raagini having used lists of sentences and Anne having used lists of words, which we ended up having to swap and revamp so that they were compatible. In addition, we found slight inconsistencies that we didn't really anticipate, like coding conventions that didn't impact the functionality of the code, but the readability. Specifically, Raagini needs to remember to name her variables and functions starting with lower-case letters. And Anne needs to make her functions in the correct order so that they are easier to read and understand. Overall, we enjoyed working together and think we worked well together as partners.