Text Mining Project

Project Reflection

Project Overview – The Big Picture

I used free e-books **from Project Gutenberg** (https://www.gutenberg.org/). The first part of the program uses links for e-books that are available on the PG website, **scrapes and pickles the file**, and outputs a text file that is usable for the other parts of the program. The analyses is carried out using dictionaries as primary data structures – **word frequency analysis, markov analysis, and sentiment analysis**. I hoped to track patterns in texts (both original and program generated) through the three techniques. I learned the three techniques along with scraping text from any web link – these have an infinite number of applications, so much so that they exist around us and we use some of them every day.

Implementation – System Architecture Level

Once I import all the required packages, I pickle the required files and do operations on these files to output files that are required by the rest of the program. As an example, I create a file that contains all the three texts that are in the input – It is useful for creating markov chains later in the program. I achieve this by using functions like open and write (for text files) and for loops to traverse through each line of the file.

For word frequency analysis, I use for loops to iterate through each word in the given file and remove all the punctuation marks. I use string operations to manipulate the marks specifically. After this processing, I create a dictionary and add all the words to the dictionary. They are keyed to how frequently they appear in the text, using for loops, again. They are simply sorted in the descending order for the sake of easy data analysis of the output using list operations.

For markov analysis, I operate the text file so that I have each word of the file as a separate string. Then I make a dictionary that maps all the prefixes with its suffixes. I use an except case here as there can be many suffixes for the same prefix. If that is the case, the function should be able to append multiple suffixes to the prefix in the dictionary. To change frames and move the function forward in terms of the words that are being processed, I use list operations on the frame. Finally, I use a 'for' loop to iterate over the dictionary and randomly pick user defined number of words. I switch to using strings at this point in the program so as to output an easy to analyze text.

For sentiment analysis, I obtain sentiment scores (compound, negative, neutral, and positive) for a user defined text by iterating over each line of the given text using for loops and analyzing it. The output for this is an averaged score across the whole text for the sake of making the score easy to analyze. I obtain the output using some level of string operations and concatenations.

A major design decision that I made was about storing prefixes and suffixes in dictionaries. My initial idea was to make separate dictionaries to account for the case of multiple suffixes for the same prefix. Due to the complexity of the code (and the level of debugging that was required), I pivoted to using a single dictionary, and appending the suffixes to the same prefix.

Results

The book, 'Voyage to Jupiter,' is based on a voyager that goes to the planet Jupiter. This is the summary of the book in a single line – the results prove this fact as 'Jupiter' and 'voyager' are the most frequently used words (subtracting the articles and grammar). The book, 'Alice in Wonderland,' is quite small to draw on word frequency results. We can definitely observe the nuances of Alice's life through the histogram: 'Alice', 'little', and 'project'.

On performing Markov analysis on the book, 'Voyage to Jupiter,' the generated text is:

"One of the coloring agents; they may be the result of interactions with the planet's shadow as they would be seen to overtake one another and gobble each other were used to select the color of the older, darker surface, giving the appearance of a hundred has actually seen Amalthea." Larry Soderblom summarized the satellite a cracked egg in this two-week period, and finally into the spacecraft proper and the left, while a train of small spots moved eastward at approximately latitude 80° S. In addition to the spacecraft kept nearly the same size and weight of a cloudy planet like Jupiter, the Voyager scientists at JPL seemed quieter than it had been unknown for so long, yet had become a thin crescent could be seen, shrinking as the spacecraft themselves. Voyagers 1 and 2 are grouped together. [260-678A]] [Illustration: A special color reconstruction was made to save a part of the Moon (Europa) to nearly as black as the Pasadena night, for there, glittering against the spacecraft trajectory with small bursts of Jovian particles are probably the result of new data began to feel the same basic composition as the magnetospheric boundary flopped in."*

(*I punctuated the first and the last couple of characters)

Most of the text makes a little less than average amount of sense (not as much astronomically as grammatically).

Similarly, on performing Markov analysis on the book, 'Dracula,' the generated text is:

"During the night, or at its worst, for the defeat of the volcanic cloud. Early Monday morning, other Imaging Team members Tobias Owen and Hal Masursky, were given some clue to his feet. "Is anything wrong?" he asked, hoarsely. "I would; if there was nothing' else in it." I felt under obligation to meet Van Helsing thinks he knows, and will be comfort to him with his sunshine, his fair places, and his song of birds, his music and his face sternly set. Lord Godalming started for the present for one orbit, is the end!" He turned as he spoke coherent words for the ship. No other form of heat; theorists calculated that the driver of the important studies of the starlight was seen, yielding an initial value of about 0.1 percent of the brooding weight off his head at once sent up a huge circular feature on Callisto. They therefore conclude that the smuts in London from Whitby. The steamers _Emma_ and _Scarborough_ made trips up and flung it over the surface. A large quantity of the terrestrial magnetosphere—a large, dynamic region around the Sun. "*

(*I punctuated the first and the last couple of characters)

Analyzing this text, it makes less amount of sense than the text generated in the previous analysis. I conclude that markov analysis makes more sense on scientific books, as compared to fictional literature.

Last, but not the least, I performed markov analysis on the three texts combined. It generated some promising results in terms of what I had expected out of it –hilariousness.

Example 1:

“**My very soul** of the Jovian radiation belts of Jupiter. **The Probe** must strike the atmosphere of the inn-yard and its fierceness is abating; crowds are scattering homeward, **and the young ladies!** He has told is the most truly exploratory of the data streams pouring in from the fact that it was in terrible plight. The dilemma had me between his finger and thumb closed on her pillow asleep; she did not seem to realize, or at any moment. * * _31 October._--Still hurrying along. The customs and manners. There were dark, and I mistrusted myself. Doctor, you don't care about life on this occasion. My expectation was not then but that you will come to Exeter yesterday, and said, suddenly but quietly: -- "But dear Madam Mina. This time she saw maps and pictures hung upon pegs. She took but a little, and the other patients who were unmounted jumped upon the poor thing became quiet and fell asleep instantly and neither of them is aware that I knew that the cords with which we know, so that **our exploration program**. For the wide-angle pictures were taken at a distance of 5.9 R_J from the Jovian system and its Moon.” *

(*I punctuated the first and the last couple of characters)

Example 2:

“How I slept, but did not wake me. He paused and raised him up. "Come," I said softly to him: -- "And now, Dr. Seward, humanitarian and medico-jurist as well as to facts of his own room. As soon as possible. And then begins our great quest. But first came the answer: "darkness and the solar wind. Between April 24 and May 27, Voyager 2’s radio reception was switched to its own magnetic field, appears to be the **blackest things** that we are all the way **he lifts his 'at as perlite as a man must not live**, lest I harm her; for I feared for Harker, though I was bewildered, and, strangely enough, I did not seem to have unchecked sway--a blue flame is seen over any place where my friend John here, who has been a report of the body of Szgany have come without special reason, but just begun. **Those children whose blood she suck are not to notice**, but remarked that the driver jumped again into our own feelings, but the exact moment that it didn't seem half so hard to refuse would be our undoings." *

(*I punctuated the first and the last couple of characters)

The sentiment analyzer strikes the relation between the generated text and the books that went into making the text. For a particular case, the scores were as follows:

Sentiment Analysis for **Voyage to Jupiter**

Compound score = 0.039946479660259354

Neg score = 0.01827626285203398

Neu score = 0.7912093205185508

Pos score = 0.042436857398301366

Sentiment Analysis for **Dracula**

Compound score = 0.027268709697614746

Neg score = 0.056639829712640116

Neu score = 0.7038449884179531

Pos score = 0.07110730607900832

Sentiment Analysis for **Alice in Wonderland**

Compound score = 0.025042714453583997

Neg score = 0.03727144535840193

Neu score = 0.6495834312573444

Pos score = 0.04875264394829614

Sentiment Analysis for **Markov Analysis generated text**

Compound score = 0.9117

Neg score = 0.052

Neu score = 0.833

Pos score = 0.115

I was able to conclude that the Markov Analysis generated text for that particular case was sentimentally most similar to Dracula, according to the score solely. The sentimentality relation depends a lot on the percentage of sentimental words randomly chosen from the books. I observed that the score was in congruence with the actual text only some of the times.

Alignment

I find that the data that I used for a pretty good fit to my program – it was mostly unrelated ('Voyage to Jupiter' contains a lot of astronomical text, whereas 'Dracula' can be classified as a mystery/thriller story.) This would have paved my way for using the sentiment analyzer, but the scores did not differ by a very large percentage. The tools that I used processed the data as they were designed to do. However, there was some level of discrepancy between the actual sentimentality of the generated text and the sentiment score.

I set out to explore if I could use sentiments to prove correlation between randomly generated text and the original text that was fed to the random text generator. I had imagined that the result of Markov Analysis would make very little sense, if any. And that if the books that I feed in are very different, the result would be hilarious. I predicted the sentiment analyzer to work at an average level. I think that the data source that I used worked very well for me, because it allowed me to choose from a wide variety of categories and types of books. For my analysis, I needed files that differ so much so that the generated text brings out differentiable qualities and sentiments.

As for the tools, the histograms were very promising and accurate. The results perfectly matched my expectations. The markov generated texted worked better than I had expected it to (after the design decision pivot that I discussed in the previous section). It made almost complete grammatical sense, as I used Microsoft word's 'review' tool to cross-verify it. Lastly, for the sentiment analysis, I am only slightly confident. Because, as I noted before, the results were not in line with what I expected to get.

Reflection

From a process point of view, defining new goals for the functions for making the output easier for the data analyst as I progressed through the program worked well for me. I could improve on how I visualize the program that I want to write – if I straight away dive into writing code, it takes hours for me to debug it. Quite a few times, I also have to search for similar applications of the program that I want to write. I could mitigate the tool-choice problems with my project by diving deep into sentiment analysis using text, and the forces that work around it. I could use the research to write my own program for it, rather than using the package that I import for this project. I think that my project was appropriately scoped – it used analysis prompts that sounded interesting to me. I unit tested functions that did not involve markov analysis – it would've been incredibly hard to test randomly generated text. I can envision a good number of applications for the functions that I came across during this project – use of randomization, text analysis, scraping, etc. I did not leave myself with a lot of room to experiment with the functions related to markov analysis. I played around with it for a major portion of the time that I spent doing this project – in the end, what worked the best was based on what I learnt from the book. One iteration for the project that I can think of is that I can make my results more visually appealing – graphs, histogram 2D plots, etc. It would translate to taking the next step for easing the process of data analysis.