Text Mining Reddit

Project Overview:

The point of this project was to create a text mining bot that would scour the most popular buzzwords from Reddit and accessing it through the API.

Implementation:

The methods I used to create my reddit bot were relatively simple, but the difficulty was in the experimentation required to get things working together.

The first major portion of the code was the Reddit Scraper. Accessing Reddit anonymously (without logging in), I used the hot topic generator function in order to come up with the most trending reddit posts of a specific subreddit (in my case, I chose the Fire Emblem subreddit). After I received a list of the three hottest posts, I filter out the "stickied" posts, because stickied posts automatically are at the top of a reddit, and thus may not accurately represent what else may currently be trending. Basically, the script goes through a logic loop that identifies the first non-sticky post and uses that as the post to scrape.

After the post is selected, I have the bot parse all the comments using the methods from the API. In this sense, I get an entire list of comments. After I separate out the "more comments", I pull the text from all the comments and split it into a large list of individual words.

Next, I have the script loop through the entire list and create a dictionary for the words, taking the instance of how many times a certain word has appeared. On the side, there are two functions. One is a list of words I deemed not helpful, which are practically particles, pronouns, prepositions, and anything that wasn't a noun. I guess in a way I made a noun finder to see how many times my favorite character's name appeared in a reddit thread. The second part I recently implemented (aside from improving the not helpful list) was adding a search function that allowed the user to view how many times the string they're looking for appears in the post.

Reflection:

This iteration of the reddit text miner wasn't too bad, I guess. I wanted to implement some sort of noun-detection API, but the prerequisites were complete sentences in order for the API to properly identify nouns. The other problem was that since it was an online forum, it might not be easy to find complete sentences depending on the subreddit. Additionally, misspelled words might not work very well with that system. Since the first iteration, I only had it give the top ten words to find out what was popular, but the level of implication is very superficial. I guess adding a search function improved functionality for a user to see how relatively popular their query is, but on the whole, the basis of seeing how popular something is based on word composition percentage is very inaccurate, and does not reflect well so much on trends, but rather the most common structural words people use in a particular subreddit.