Sophia Li
Software Design

Assignment 10: Text Analysis

Project Overview:
Inspired by the anime “Psycho Pass”, this Python script pulls data from your Facebook and Twitter feed and does sentiment analysis on your posts after some cleanup of the raw input received. After sentiment is generated, it is mapped to a color, the hue of which is roughly determined by the polarity and shade determined by the subjectivity.

Implementation:
One can run this code through the terminal by running python hue_generate.py in the directory where the file is located. On screen prompts will tell the user what to input next. In order to pull data from Facebook, a Facebook key is required.

Libraries this code uses:
pattern.en → sentiment analysis
pattern.web → acquiring data from social media
webcolors → translating RGB color values to English color names
random → shuffling values in generate color step to randomize color spectrum
Image → for PIL

This code has 4 major steps.

1. Pull data from Twitter and/or Facebook
At this point, using pattern.web’s search abilities, we pull data from Twitter and Facebook using hashtags and a specific Facebook validation key. This data is added to a list and is passed on for cleaning.

2. Clean input
Since prepositions like “and”, “or”, “it” appear frequently in texts and do not contribute particularly to the tone of a post, we strip out all of these bad words from the data we get from social media. This is done by looping through each element of the list from the previous step, and creating a new list of individual words from each string. We can then do a comparison between the word list and the bad words to create a new string only containing words not in the bad list.

3. Sentiment Analysis
Sentiment analysis using pattern.en’s abilities is done on the clean input. The result is averaged to create a final value representing average sentiment across all inputs.

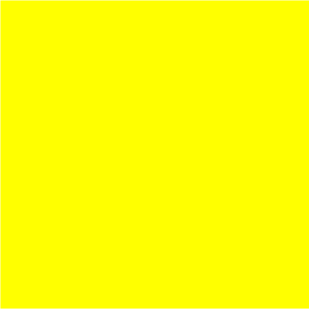
4. Generate color
Using the results from sentiment analysis, we remap the numbers to be within 0-255 range. Two of the RGB values are set to the polarity portion of the analysis, while the last value is set to the subjectivity portion of the sentiment analysis. This roughly results in color being determined by polarity while shade being determined by subjectivity. This color is displayed via PIL, and a text output translates the RGB value to an Engsh color name.

Steps 1-3 are all done in one function, (facebook_call, twitter_call)

The main() function initializes when the user runs the script, and walks him through the information needed to generate his hue. In addition to searching Twitter usernames, this script can also search hashtags to find general Twitter sentiment on a certain topics.

Results:
It’s clear that the more positively polarized we are, the brighter the color. For example, searching these Twitter hashtags generates:

‘best day’



his individual's hue is: clear yellow
Sentiment analysis shows that this individual or topic is 0.72 positive and 0.37 subjective

In the words of Charlotte Bronte :
Happiness quite unshared can scarcely be called happiness; it has no taste

Continue keeping bright things around.

‘fml’

This individual's hue is: steel dimgrey
Sentiment analysis shows that this individual or topic is -0.15 positive and 0.62 subjective

Remember the words of Abraham Lincoln who said:
Folks are usually about as happy as they make their minds up to be.

Look to happier things.

And searching my posts, I find

This individual's hue is: clear cadetblue
Sentiment analysis shows that this individual or topic is 0.29 positive and 0.6 subjective

In the words of Anne Frank :
Whoever is happy will make others happy.

Continue keeping bright things around.

Interestingly enough, I found my polarity was 0.29. I think it’s kind of creepy but really cool that I wrote a program that could examine my life like this and provide insight on the kind of person an impartial judge (pattern.en) thinks I am. Compared to the polarities of my friends when I ran this code, I’m actually a pretty negative person. I can think of two reasons for this:

Firstly, I don’t use the traditional smile :) but rather C: or :3 or even (ﾉ´ヮ´)ﾉ*:･ﾟ✧. Sentiment analysis can’t recognize these emoticons, and assigns them values of 0 instead of the 1 it actually means. Additionally, sentiment analysis really stutters at processing sarcasm, which is understandable, so more comments are tagged as negative than necessary.

Second, I need to be more positive haha.
Haha.
;____;

Reflection:
I probably should have worked with a partner. I created working code fine on my own, but I feel like it might have been more fun if I worked with someone else and they would have had interesting ideas to contribute to this project. However, being able to work on the code whenever I felt like it was a pretty edifying experience. Because I worked alone, I was able to start the project early and got the bulk of the coding done on Friday and Saturday. I think I was successful in managing my time and I didn’t have to pull an all-nighter!

I wish I had more time to work on the specifics of the project though. I had initially planned to make a GUI, but found the learning curve too steep for the 3 days I dedicated to this project. Additionally, I spent a lot of time flailing around before I decided on the path I wanted to take with my functions. In the future, I’ll draw out a diagram of the way all the functions link together first. Unit testing wasn’t really considered, but print declarations were used liberally when debugging. These are all gone in the final code, of course, since I feel like it makes the final script too heavy.