

Software Design Homework 5

Griffin Tschurwald and Sawyer Vaughan

Project Overview: Our project is a fake celebrity tweet generator. Our project takes as an input a Twitter handle, and utilizes Markov analysis to generate a fake tweet from that account. The user can specify the length of the output tweet in words as well as the length of the prefixes that the Markov analysis uses.

Implementation: Our project takes a Twitter handle as an input. It then mines Twitter for the tweets that have been sent from that account from roughly the last two weeks, and saves them into a text file. From that text file, we create a dictionary mapping all of the prefixes (tuples of length specified by user) to their possible suffixes (a list of all possible suffixes for that prefix). Then, we randomly pick a prefix to start, and it randomly picks one of the suffix options for that prefix. It then adds the word it picked and omits the first word from the previous prefix to create the new prefix it will analyze. It continues in this manner until the length specified by the user is reached. Unfortunately, not many people will be able to use our code. If you have a text file that is the tweets that you want to do Markov analysis on, it is easy for anybody to run our Markov synthesis function. However, the Twitter text mining functions would not be able to be used easily. I had to obtain my own API key in order to access Twitter's data. I had to input this key into the pattern.web API and reinstall pattern in order to make the Twitter search function work.

Results: Our biggest obstacle was Twitter's archiving scheme. The way that Twitter archives tweets means that it was impossible for us to mine tweets from more than two weeks ago. This meant that the amount of tweets that we could gather was too little to really do a comprehensive Markov analysis on the tweeter's vocabulary, syntax, and content.

We were successful at composing tweets from the dictionary created by the Markov analysis. Our output was highly adjustable. We generalized the functions in order to made the length of tweets and the depth of the Markov analysis adjustable. The lower the depth of the Markov analysis was, the more that the generated tweet would seem unique, but the less coherent the tweet would be. Unfortunately (mostly because of the lack of text that we could mine), a sufficiently high depth to cause complete coherency simply caused the generated tweet to be identical to the user's real, previous tweet.

A few examples of our program's output:

- @kellyoxford (depth 1): "photo sums everything up, forever. I hope Vince Gilligan is as it just to waking yourself up with Alzheimer's patients. Proud of the only people up with you. ♥ it"
- @kellyoxford (depth 2): "did we go wrong?? Watching his red carpet interview I felt like Jared Leto has the perfect ombre just to taunt women. #oscars unless he went blind, yes. frankly, the"
- @kellyoxford (depth 3): "pyrogy are eastern euro.. It's flour and potato and onion... Ukrainian, Polish, German etc good. I'm glad Ukraine has no exports. Pyrogy. why obvious? Congratulations to for bringing to LA"
- @Oprah (depth 1): "my favorite! #theOscars agree is always required. Sipping now! Morning all think? #LifeClassSingleMoms we all thank you want me if you're a double standard. Single dads are heralded. #lifeclasssinglemoms Mark"

- @Oprah (depth 2): “Trying to get home in time for tweet party. #HavesAndHaveNots thanks Britt. ? .@OWNing_JOY thank you for the eye-candy #HavesAndHaveNots do real mothers behave that way. So cold blooded Here”
- @Oprah (depth 3): “is it society or your own judgement? Do u feel you're judging yourself harshly? #SingleMoms Do u all think we as society judge single moms harshly? #lifeclasssinglemoms you'll get a”

Reflection: I wish that we would have begun work sooner. I also wish that there were a better way to coordinate changes to a common code. We were forced to implement separate codes and combine them by emailing our code to each other and copy-pasting each other's code into our own. Because of our limitation of not being able to work on the code at the same time, a better strategy would have been to include a specific plan for which functions each of us was going to write. We also could have improved our code by pair-programming.

Our project was appropriately scoped. We could have had a more specific goal, because our project was so open-ended that we had a limited scope. We also didn't have a good plan for unit testing, although we did test the Markov function and Twitter function separately. We did declarative testing, using print statements in order to tell what was happening inside our functions. Our outputs did seem reasonable, because we could parse through the text file that was the raw data from the tweets and look at our generated tweets in order to make sure that the Markov synthesis was working properly.