Predicting the 2014 Midterm Election Results

Liani Lye

Due 09.30.14

1 Overview

There are 36 U.S. Senate seats and the entire House of Representatives to be determined in the upcoming Midterm Elections. With that in mind, I chose to predict the outcome of a few of these races by gauging the sentiment, or feeling, around certain candidates.

2 Implementation

2.1 Libraries

codecs, used to log tweet text, author follower count, and resulting prediction value.

date and **datetime**, used to save log file names based on time and date.

indicoio, used to calculate sentiment of tweet text. Results range from 0 to 1, with 1 being more positive. A more accurate and contextual sentiment package than **pattern.web**'s.

pattern.web, a python wrapper for the Twitter API. Used to search for tweets. Does not require a unique API key.

tweetpony, another python wrapper for the Twitter API. The **common** submodule is specifically used to execute Twitter GET requests for author information. Requires a unique Twitter API key.

2.2 Code Base

pobama.py consists of two functions, **findFollows()** and **search()**.

findFollows() is based off *common* to execute GET requests and return a user's follower count. Without GET requests, only Twitter information relating to my Twitter profile, since Twitter API keys are associated with the applicant's Twitter account, can be mined.

search() does three things: first, this function acquires query-based tweets through *pattern.web*. Then, a sentiment rating is assigned to each tweet with *indicoio*. Finally, the sentiment rating is multiplied by the tweet author's follower count from **findFollows()**. The rationale behind this multiplicative scaling is that a user with a higher follower count has the potential to influence more people. A summation of purely sentiment would ignore this important factor.

The result is a value that one can use to gauge candidate performance. The higher the number, the more likely the candidate is to poll well.

2.3 Running the Code

The repository can be cloned through:
https://github.com/lianilychee/SoftwareDesign/tree/master/hw3

Pattern and TweetPony must also be installed through sudo apt-get.

3 Results

According the the Federal Election Commission, there are four candidates for a single Delaware Senate. In the interest of staying within Twitter's API call rate limits, the fewer candidates to analyze, the better.

Five tweets were scraped per candidate using the query "[first_name last_name] senate." The results are as follows:

Candidate	**Score**	**Comments**
Coons, Chris A	3005.34	(D) incumbent
Groff, Andrew R	0	0 search results
Smink, Carl Robert	988.96	2 search results
Wade, Kevin L	1397.02	(R)

Basing purely off of the above numbers, I predict that Chris Coons will be re-elected for the 2014 to 2020 cycle.

4 Reflection

Over the summer, I used JavaScript and D3 to create visualizations for tweet information analysis. In a sense, though in a different language, this project is an extension of my summer work. I wanted to explore tasks that I previously hadn't been able to: working with APIs and sentiment analysis. I also dabbled with unit testing, which was performed using sentiment analysis of my own tweets (a grand total of seven).

Future iterations of this project would involve:

1. Finding an API with a higher rate-limit ceiling (the JS one caps at 180 calls every 15 minutes)
2. Implementing objects so the user can search for multiple candidates at once
3. Have files named by query; scrape total value across multiple files for a larger sample size