W205 FINAL PROJECT

Spring 2016

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document. When you're ready to add your content, just click here and start typing.]

Sally Hong, Mary Lewis

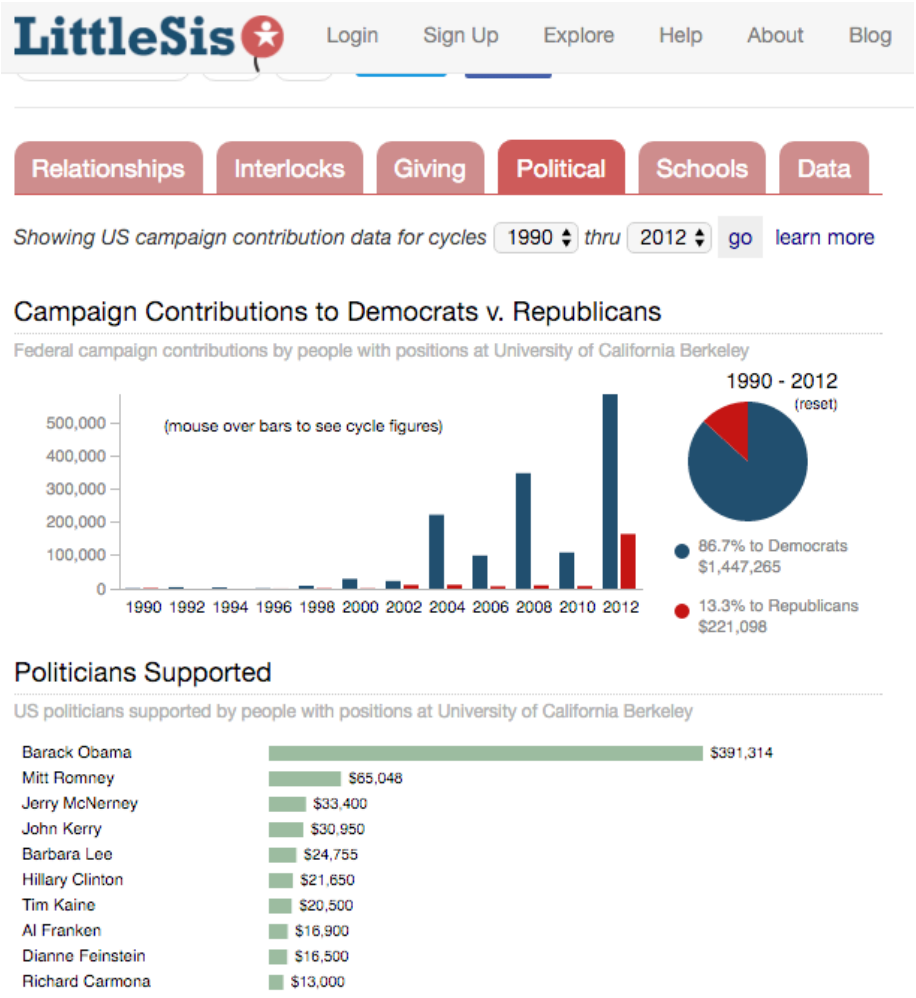
Table of Contents

I. Introduction and Motivation

LittleSis is an online free database that details the connections between powerful people and organizations. It is a play on the words “Big Brother”.

This site brings all the publicly available information into one place, allowing users to easily track key relationships of politicians, business leaders, lobbyists, financiers, and their affiliated institutions.

As LittleSis already has extensive research on various queries. For example, when looking for donations trends about a specific school, LittleSis API already lists information such as “Campaign Contributions to Democrats vs. Republicans” and “Politicians Supported”.



Source: http://littlesis.org/org/14980/University_of_California_Berkeley/political

Motivation

The questions we wanted to explore were as follows:

1) What are the connections between graduates of Top US schools and US politicians?
2) How much money do graduates of top schools donate and to which politians/parties?
    a. What are the total dollars donated to politicians and how are dollars split by party?
3) What is the current real time social media sentiment during the current election season?

The purpose was to reformulate the question to a larger scope—a more aggregated outlook of the donations by school as well as the top donors of a school. In other words, following the money with a school considered.

II. Getting the Data from Little-Sis

Obtaining the Complete Data Set from LittleSis.org

LittleSis makes its entire database freely available through an accessible API with a user key.

- Calls are HTTP calls; limited to 10,000/day

The original plan was to obtain data via loops calls to the API or webcrawler, but we were unsure of the total DB size and the time required.

Therefore, we reached out and contacted LittleSis support for a direct transfer of the complete DB dump. We received it in a form of a link to a S3 bucket on Amazon.

Making MySQL Format Compatible with PostgreSQL

We selected to convert to PostgreSQL because it is:

- Open source, non-proprietary
    - PostgreSQL has an application you can download on your local machine to use.
- Simple table structure and types
- Pre-installed and configured on the EC2 instance for w205 AMI
- Hive SQL had limitations with SQL queries (as experienced in Exercise 1)

Migrating, Transforming, Loading Local 4GB MySQL DB to EC2/Postgres DB

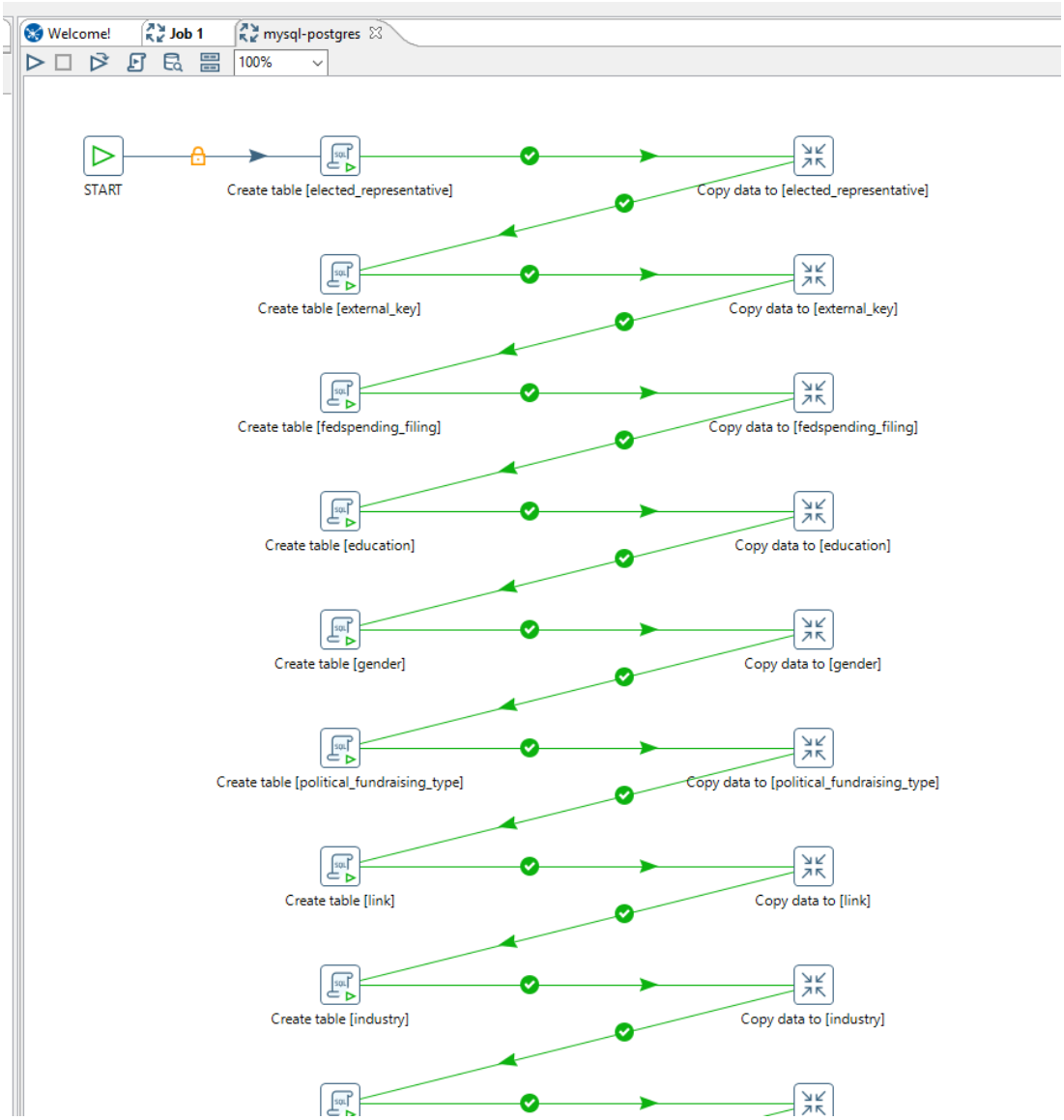
The plan was to:

1) Generate schema from MySQL
2) Export MySQL tables to CSVs
3) Migrate CSVs to EC2

The challenge was that (mentioned earlier) MySQL CSV export format is incompatible with PostgreSQL.

Alternative Method: Pentaho

Pentaho Business Analytics offers products that provide data integration, OLAP services, reporting, dashboarding, data mining and ETL capabilities. For our project, we used Pentaho to convert MySQL DB to PostgreSQL DB.



1. Download the SQL file and create MySQL database:
   `wget https://s3.amazonaws.com/littlesis/public-data/littlesis-data.sql`
2. Convert MySQL → PostgreSQL
   Use Pentaho (http://www.pentaho.com/)
   This converts the MySQL schema to be compatible with PostgreSQL
3. Create PostgreSQL database for LittleSis data

The Data

The entire data consisted of 39 tables:

```
littlesis=# \dt
                  List of relations
 Schema |            Name             | Type  |  Owner
--------+-----------------------------+-------+---------
public | alias                        | table | postgres
public | business                     | table | postgres
public | business_industry            | table | postgres
public | business_person              | table | postgres
public | custom_key                   | table | postgres
public | degree                       | table | postgres
public | domain                       | table | postgres
public | donation                     | table | postgres
public | education                    | table | postgres
public | elected_representative       | table | postgres
public | entity                       | table | postgres
public | extension_definition         | table | postgres
public | extension_record             | table | postgres
public | external_key                 | table | postgres
public | family                       | table | postgres
public | fec_filing                   | table | postgres
public | fedspending_filing           | table | postgres
public | gender                       | table | postgres
public | government_body              | table | postgres
public | industry                     | table | postgres
public | link                         | table | postgres
public | ls_list                      | table | postgres
public | ls_list_entity               | table | postgres
public | membership                   | table | postgres
public | org                          | table | postgres
public | ownership                    | table | postgres
public | person                       | table | postgres
public | political_candidate          | table | postgres
public | political_fundraising        | table | postgres
public | political_fundraising_type   | table | postgres
```

```
public | position                  | table | postgres
public | professional              | table | postgres
public | public_company            | table | postgres
public | reference                 | table | postgres
public | relationship              | table | postgres
public | relationship_category     | table | postgres
public | school                    | table | postgres
public | social                    | table | postgres
public | transaction               | table | postgres
(39 rows)
```

III. Working with LittleSis Tables

Due to the large amount of data, it was a challenge to understand the actual data content and respective locations. Tables were highly indexed and required multiple, iterative joins to achieve digestible content.

These were the common obstacles dealing with big data:

- Desire to use as much data as possible for the project
    - Too much data but too little time dilemma
- Siphoning some information required advanced programming
- For next/future steps:
    - Lobby groups/ PACs : these groups are not tagged to a political party in LittleSis
    - Multiple recursive joins though the “relationship” table may yield a political association
    - Number of iterations not known
        - Political association not guaranteed.

The solution was to identify all people (not organizations) classified as “elected representative” or “politician” and examine donations made to these people by graduates of top schools.

IV. REST API as a Presentation Layer Tool

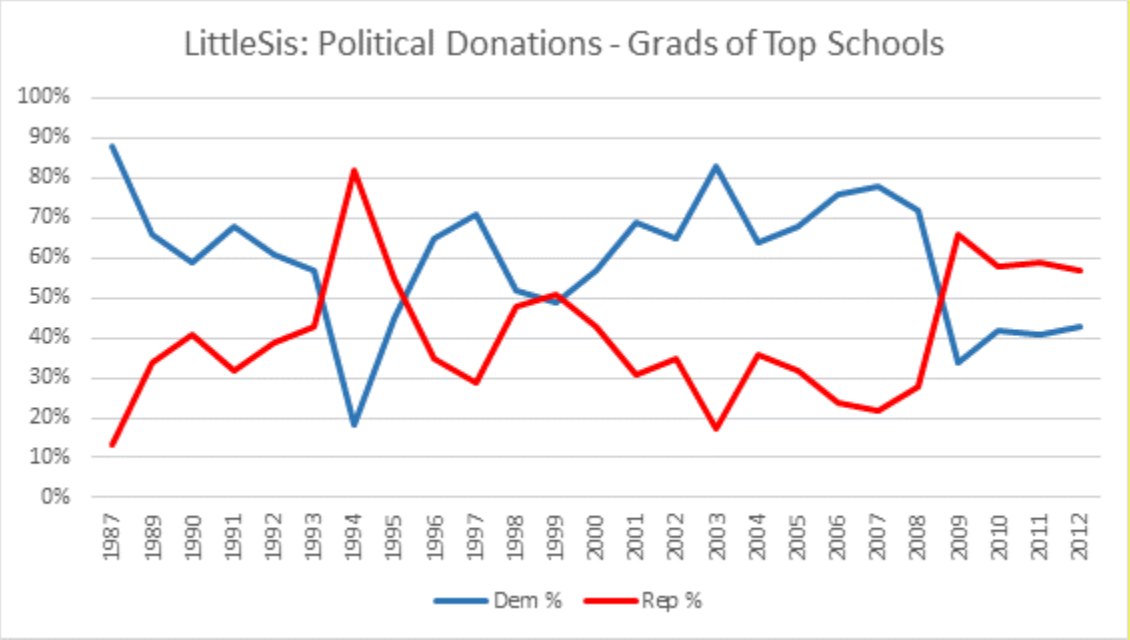
We decided to use REST API as the presentation layer tool for these following reasons:

- RESET API lends itself to “walking a graph” of relationships
    - Queries linked to the results of the previous query
- Similar in concept to following links through web pages
- Similar to the LittleSis.org website itself

Final results using REST API request:

curl http://<hostname>.compute-1.amazonaws.com:8080/donationsummaries/all/all

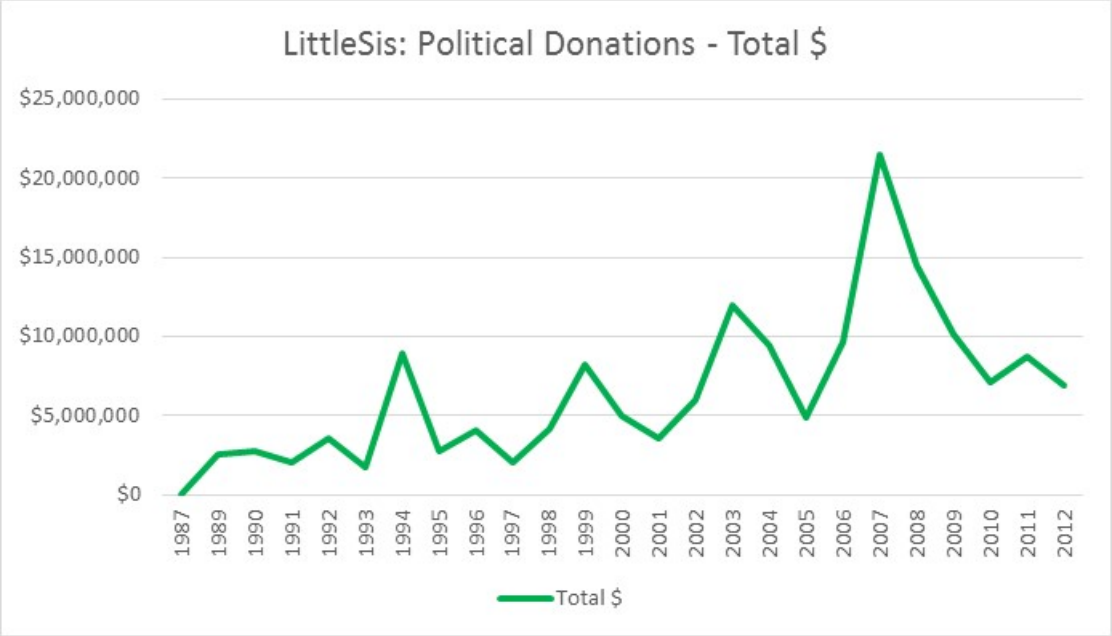
V. Static Database : Interesting Findings and Discussions



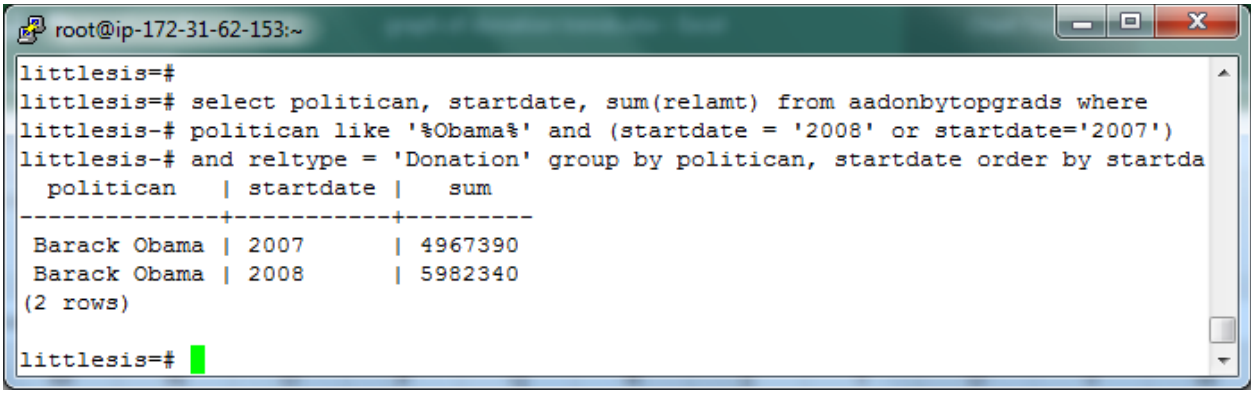
Post 9/11 donations were much higher to Democratic candidates /politicians. This changed after Obama became president, after which Republican donations surpassed Democratic donations. The 2010 elections were a direct result of this support.

Republicans swept the 2010 elections, breaking many records, from Wikipedia:

> Approximately 82.5 million people voted.[2] The Democratic Party suffered massive defeats in many national and state level elections, with many seats switching to Republican Party control. Although the President's party usually loses congressional, statewide and local seats in a midterm elections, the 2010 midterm election season featured some of the biggest losses since the Great Depression. The Republican Party gained 63 seats in the U.S. House of Representatives, recapturing the majority, and making it the largest seat change since 1948 and the largest for any midterm election since the 1938 midterm elections. The Republicans gained six seats in the U.S. Senate, expanding its minority, and also gained 680 seats in state legislative races,[3][4][5] to break the previous majority record of 628 set by Democrats in the post-Watergate elections of 1974.[5] This left Republicans in control of 26 state legislatures, compared to the 15 still controlled by Democrats. After the election, Republicans took control of 29 of the 50 State Governorships.



A significant portion of the total dollar spike in 2007-2008 (seen above) is driven by the Obama campaign:



VI. Streaming Twitter Setup

Modifying Exercise 2, tweets were streamed altogether (with the removed restriction of symbols such as @) and stored into a database called ‘tweet’.

Due to the limitations of the Storm architecture (more in detail in a later section), an infinite while loop was implemented.

The base code[1] was found from the internet. However, further modification was needed to store in the database for the project as well as fit the scope of the project. If there was anything to take away from "streaming" in general is—store now and filter later.

The streaming duration was around one week with intermittent stops here and there to check the status of the database.

Official Twitter Names of Schools

Here are the official twitter names of the various colleges.

University Name	Twitter Name
Princeton	@princeton
Harvard	@harvard
Yale	@yale
Columbia	@columbia
Stanford	@Stanford
University of Chicago (UChicago)	@Uchicago
Massachusetts Institute of Technology (MIT)	@MIT
Duke	@DukeU
University of Pennsylvania (UPenn)	@Penn
California Institute of Technology (CalTech)	@CalTech
Johns Hopkins University (JHU)	@johnshopkins
Northwestern	@NorthwesternU
University of California - Berkeley (UC Berkeley)	@UCBerkeley
Dartmouth College	@dartmouth
Unvirsity of Virginia (UVA)	@UVA
New York University (NYU)	@nyuniversity
University of Michigan - ann Arbor (Umich)	@UMich

Noise Cleaning

The biggest hassle of dealing with Twitter streams, compared to a static database, is the amount of unstructured data. Even though the keyword list were set up, there would be cases to "fix".

For example, filtering by "%@mit%" drew up "@mitchell" which is wrong! Therefore, implementing "%@mit %" AND "%@mit" circumvented this problem.
This was mostly trial and error to figure out which filters were working and which were bringing in noise.

[1] https://github.com/alexfrancisross/TwitterStream/blob/master/twitterstream.py

Database Creation and Selection

Either one can use the database provided (`xx_sample_tweet_database.sql`) in the Sample DB folder, which is a filtered version of tweets streamed for a week in April, or one can refer to the `README.txt` for a step-by-step process to generate one's own tweet database.

VII. Streaming Twitter Challenges

For Exercise 2 and Lab 6 we were introduced to the Storm architecture for streaming purposes. For the scope of the assignments (which were relatively short-term), this was sufficient. However, for this project, while trying to implement the similar architecture, we ran into several problems.

We were running this at full capacity on local commodity hardware. (It's crazy to think that we were only streaming only 1% of tweets in English!)

Empty Queue Exception

Streaming would work initially. However, after 10 minutes to an hour, the stream started spouting out "Empty Queue Exception".

After googling our problems, we realized that we were getting throttled. Streamparse by nature (due to the Storm architecture) keeps trying when it fails to retrieve a tweet as it wants to process everything at least once.

This was counterproductive to our project as we didn't care about getting everything but rather it not crash at all.

Cannot Connect to Host

During streaming, other devices connected to the WiFi (such as mobile devices and tablets) could not use the WiFi network simultaneously.

At first, we thought it was bandwidth issue; however, it was that because streaming made constant requests to the router. We were essentially running a DOS (Denial of Service) attack on our own router.

Too Many Accesses to a File

This is a common problem found in Hadoop as well—basically, anywhere where multiple accesses are required and it exceeds a certain limit.

To address this, we set the “no file limit” to an arbitrarily high number.

- “ulimit \n” from 1024 (default) to 60,000

VIII. Twitter Sentiment

To analyze the sentiment, we used an API specialized in natural language processing from the Python library called TextBlob. This application is commonly used for part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, etc.

For our specific project, we used sentiment analysis.

Given the tweet, it calculates the polarity, which can range from -1 to 1.
Then, it classifies the tweet as:

- Positive sentiment, if the polarity is positive
- Neutral sentiment, if the polarity is zero
- Negative sentiment, if the polarity is negative

IX. Sample Queries and Results

Included are six different serving script queries one can perform.
Descriptions and screenshots are provided for each query.

Query 1 : 01_candschoolcount.py

Description: See how many times candidate comes up at each school
Input: candidate
Output: candidate, school, count

```
D:\twitter\serving scripts>python 01_candschoolcount.py hillary
  "hillary": yale - 203
  "hillary": harvard - 75
  "hillary": berkeley - 12
  "hillary": upenn - 6
  "hillary": stanford - 4
  "hillary": mit - 2
  "hillary": umich - 2
```

Query 2 : 02_toptweetsentiment.py

Description: See the top tweets of X candidate and count by sentiment
Input: positive or negative, candidate
Output: tweet, count, sentiment

```
D:\twitter\serving scripts>python 02_toptweetsentiment.py positive hillary
positive
  "hillary": RT @PaulBegala: .@HarvardIOP poll: @HillaryClinton beats @realDonaldTrump among young voters by more than @POTUS' margin over Romney https:â€¦ | positive | Count: 50
```

Query 3 : 03_toptweetscandschool.py

Description: See the top tweets for THAT school about X candidate
Input: candidate, school
Output: tweet, count, sentiment, school

```
D:\twitter\serving scripts>python 03_toptweetscandschool.py hillary harvard
  "hillary", "harvard": RT @PaulBegala: .@HarvardIOP poll: @HillaryClinton beats @realDonaldTrump among young voters by more than @POTUS' margin over Romney https:â€¦ | positive | Count: 50
  "hillary", "harvard": Millennials favor @HillaryClinton in Harvard poll https://t.co/YGKGhhpk47 via @bpolitics | neutral | Count: 2
  "hillary", "harvard": Cry baby cry, another Harvard Lawyer Kept out of the WHITE @HouseGOP @sevigny_rob @oreillyfactor @HillaryClinton  https://t.co/S32PXCxiiG | neutral | Count: 2
  "hillary", "harvard": .@HillaryClinton in commanding lead over @realDonaldTrump among young voters - Harvard poll: https://t.co/0LZ7pad5Vy https://t.co/PRWvPz79rz | positive | Count: 2
  "hillary", "harvard": Bloomberg: @HillaryClinton Beats @realDonaldTrump for Millennials in Harvard Poll https://t.co/ryAjROMz9m via @bpolitics #ImWithHer | neutral | Count: 2
  "hillary", "harvard": From @moody Harvard University Institute of Politics survey 61% voters age 18-29 say vote for @HillaryClinton https://t.co/WuYds5006u | neutral | Count: 2
  "hillary", "harvard": @Morning_Joe @morningmika Also from the Harvard poll @HillaryClinton DOESNT HAVE problem w/ millennials against GOP: https://t.co/A6VhPNULsQ | neutral | Count: 2
  "hillary", "harvard": @HillaryClinton Madame Secretary, I'm a supporter of you and President Clinton. Will you or he comment on the Harvard U Tziip Livni matter? | neutral | Count: 2
  "hillary", "harvard": @PaulBegala @HarvardIOP @HillaryClinton @realDonaldTrump @POTUS    ... but Trump has enormous lead with the OldWhiteRacists voters! | neutral | Count: 2
  "hillary", "harvard": RT @upayr: Harvard Inst.of Politics:Poll of 18-29yo,Hillary has commanding lead against Trump among millennials.@Obamaispres @GQue242 @docdâ€¦ | neutral | Count: 2
```

Query 4 : 04_schoolcandsentpct.py

Description: Show breakdown of candidates, sentiment, and count
Input: school
Output: candidate, sentiment, count, percentage

```
D:\twitter\serving scripts>python 04_schoolcandsentpct.py harvard
  harvard | @berniesanders | positive | 331 | 98.22 pct
  harvard | @berniesanders | neutral | 4 | 1.19 pct
  harvard | @berniesanders | negative | 2 | 0.59 pct
  harvard | @hillaryclinton | positive | 120 | 40.40 pct
  harvard | @hillaryclinton | neutral | 117 | 39.39 pct
  harvard | @hillaryclinton | negative | 60 | 20.20 pct
  harvard | @realdonaldtrump | positive | 149 | 45.99 pct
  harvard | @realdonaldtrump | neutral | 150 | 46.30 pct
  harvard | @realdonaldtrump | negative | 25 | 7.72 pct
  harvard | @tedcruz | positive | 343 | 22.91 pct
  harvard | @tedcruz | neutral | 802 | 53.57 pct
  harvard | @tedcruz | negative | 352 | 23.51 pct
```

Query 5 : 05_schoolsentpct.py

Description: Show breakdown of sentiments of a school
Input: school
Output: sentiment, percentage

```
D:\twitter\serving scripts>python 05_schoolsentpct.py harvard
  harvard | positive | 459 | 61.45 pct
  harvard | neutral | 238 | 31.86 pct
  harvard | negative | 50 | 6.69 pct
```

Query 6 : 06_highestsentschoolcand.py

Description: See which schools have the highest sentiment % for a candidate
Input: positive or negative, candidate

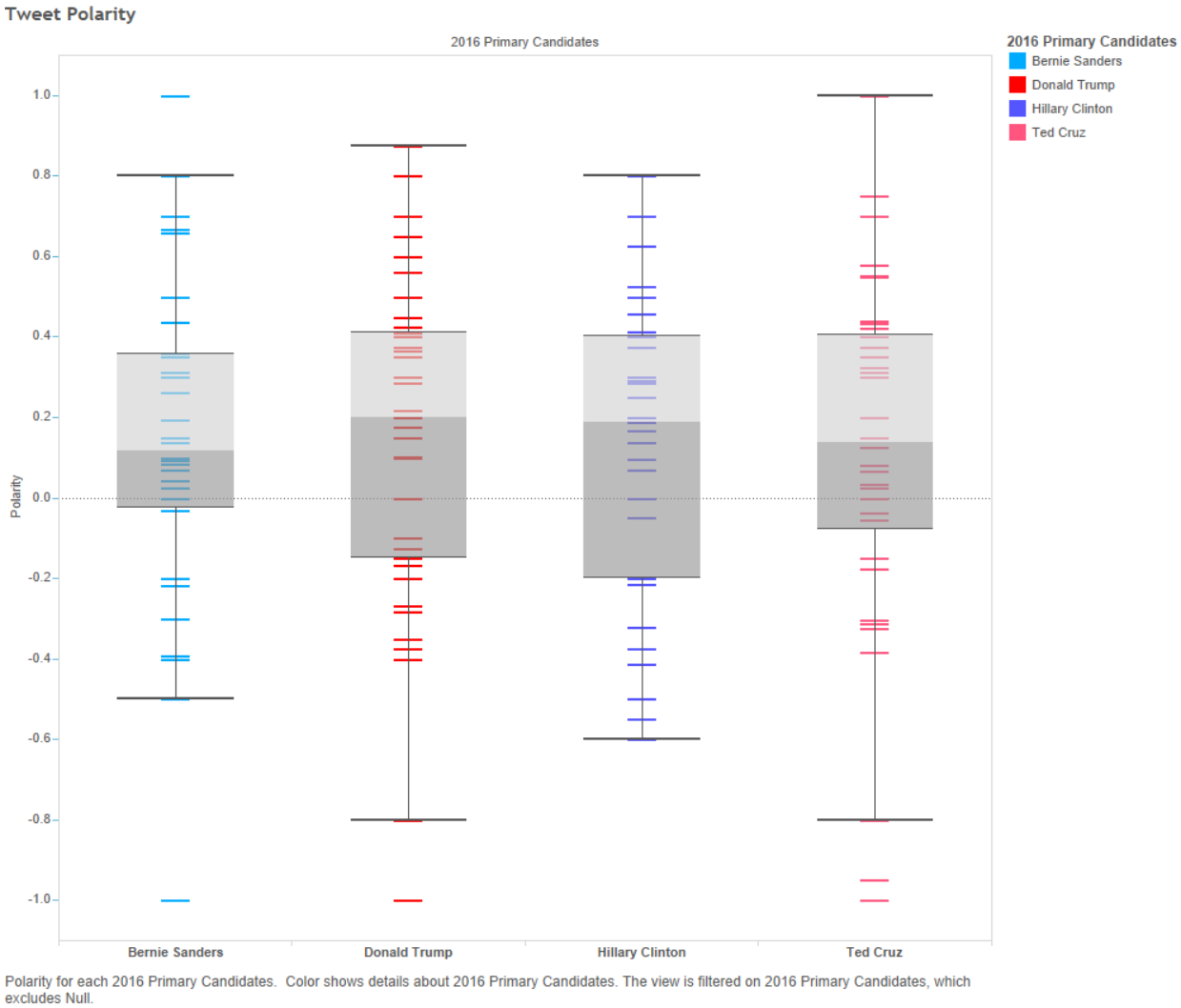
Output: school, percentage

```
D:\twitter\serving scripts>python 06_highestsentschoolcand.py positive hillary
  hillary | positive | yale | 59 | 50.00
  hillary | positive | harvard | 52 | 44.07
  hillary | positive | stanford | 3 | 2.54
  hillary | positive | umich | 2 | 1.69
  hillary | positive | berkeley | 2 | 1.69
```

X. Streaming : Interesting Findings and Discussions

Boxplots of Overall Polarity amongst 2016 Primary Candidates

This is a boxplot that demonstrates the polarity using Twitter Sentiment analysis out of the database where a school was mentioned. Out of this environment, it seems that Ted Cruz (especially) as well as Donald Trump had wide boxplots. Bernie Sanders had the smallest boxplot. Surprisingly both Hillary Clinton and Trump had a high median polarity. However, this should be taken with a grain of salt as it is near 0.2. Tests should be run to verify its statistical significance.



Top Retweets from College Campuses Regarding 2016 Primary Candidates

This is a visualization of the top retweets in college campuses regarding the 2016 primary candidates. The largest bubble mentions Bernie Sanders at Yale. At a glance it seems that there are some questionably negative things about Ted Cruz being said in association with Princeton, calling him a “A walking personality disorder who ran for president”.

Tweet. Color shows details about Tweet. Size shows count of Tweet. The marks are labeled by Tweet. The view is filtered on Tweet, which keeps 15 of 354 members.

2016 Primary Candidates and their Appearance Count in Tweets by School

Lots of case statements were needed to made to complete this query. From the screenshot below, there is an abnormal surge for Bernie Sanders at Yale. This was due to him visiting Yale recently for the Connecticut primaries.

candidate	school	c
clinton	yale	203
clinton	harvard	71
clinton	stanford	4
clinton	berkeley	4
clinton	upenn	4
clinton	umich	2
clinton	mit	2
cruz	harvard	263
cruz	yale	33
cruz	stanford	14
cruz	upenn	10
cruz	uchicago	2
cruz	jhu	2
sanders	yale	1555
sanders	harvard	337
sanders	berkeley	22
sanders	upenn	6
sanders	jhu	2
sanders	uchicago	2
trump	harvard	146
trump	yale	89
trump	upenn	55
trump	stanford	6
trump	berkeley	3
trump	mit	2
trump	jhu	2

(26 rows)

Note: This shows that more Twitter noise cleanup is needed. “jhu” is shown as part of a URL link. This only accounted for 2 tweets but still “noise” regardless.

```
schoolstream=# select tweet from schoolpol where lower(tweet) like '%jhu %';
                                   tweet
------------------------------------------------------------------------------------------------------------
 #Bernie Sanders to Fight All the Way to Philadelphia Convention Good Morn... https://t.co/25cZf0kjHu via @YouTube @berniesanders #primaryday
 #Bernie Sanders to Fight All the Way to Philadelphia Convention Good Morn... https://t.co/25cZf0kjHu via @YouTube @berniesanders #primaryday
(2 rows)
```

An interesting tweet from Berkeley regarding Bernie Sanders.

```
@HillaryClinton #BernieBlackout?! The remainder of my #Cal #Berkeley scholarship will go to @BernieSanders campaign. Thanks, #imwithhillary
```

Here is a heatmap version of the table above for visualization purposes.

XI. Future Applications

Scaling Out and Improvements in Twitter Streaming

This can be scaled to include other schools in the future (as well stream more). Though we only streamed for one week, the database was getting really big. However, in an ideal world, it would be nice to stream for longer periods of time and see “long-term” trends.

Furthermore, it would be nice to keep on top of all the trending hashtags to not miss a tweets that refers to a specific event.

Moreover, noise reduction can always be performed. Sometimes one can prevent it beforehand, but other times one has to comb through the current database to see if there are mistakes. In the future, better filters should be implemented to get cleaner data.

Scaling Out to Incorporate Other Datasets

Another interesting approach to explore is analyzing the sentiments of a politician's profile on Wikipedia and tracking the traffic. Along with real-time news sources, one could potentially observe how a news event affects interest in the candidate along with the sentiments that follow in social media.

Statistical Analysis to Complement Data Analysis

Including more covariates about the individuals (such as graduation year, major, hometown), one can run some sort of regression where given an individual and their alma mater (amongst other information), how likely and how much would they support a political candidate?

XII. Other

Prezi presentation : https://prezi.com/uoro8xc0mbbr/205-final-project/

Pentaho : http://www.pentaho.com/download

TextBlob : http://textblob.readthedocs.io/en/dev/