W205 Summer 2017 Course Project

Andrew Kabatznick, Clay Miller, Himal Suthar, Ji Zheng

Introduction

When events happens, people usually talk about them. Some people write about them. A select few even get paid to write about them to let others know about the event. For this project, we wanted to study the relationship between news stories and the public's reaction to what the news story is about – not to be confused with the public's reaction to the news story itself. We also wanted to study the variation in this relationship depending on the type of event in question. For example, do people start talking about an entertainment event (for which they do not need a news story to know that it happened) immediately after it occurred, or after it has been written about? How does this compare to a political event (for which the general public does not know when or what happened)?

To accomplish this, we will be looking at the relationship between when a New York Times article regarding a certain event was released, and the number of tweets containing words related to that event in the same time period. For this study, we will be analyzing three keywords that correspond three different kinds of events: "Game of Thrones" for entertainment, "Bannon" for politics, "Charlottesville" for national news, and "Kaepernick" for sports news.

How it Works

For every article the New York Times releases, there are a set of facets associated with the article – tags and keywords that can identify the article such as "Game of Thrones (TV Series)" or "Donald Trump." We first pull all articles from the New York Times, along with their facets, to our database every day automatically. Next, we query all facets that are less than two days old and use those facets as they keywords in our twitter query, taking care to not write duplicate tweets to postgres. This will allow us to get enough tweets for a baseline for any event as well as tweets post-event without having to rely on streaming twitter data.

After we've pulled the appropriate data, we are able to search our tweet database by keywords that we selected and obtain the number of Tweets for that keyword per hour per day using Tableau. Simultaneously, we can search our database for articles that have facets matching the keyword to obtain when NYT articles containing that facet were released per hour per day. Finally, an overlay can be created manually in which one can view both the article release times and tweet volumes on the same timeline to analyze any trends.

Technology Decisions and Consequences

A few key decisions were made regarding the system architecture that affects the overall usability and status of system that we would like to point out.

First was our decision to use an EBS volume for storage. While this drastically decreases the scalability of our system, it greatly increase the ease of use and decreases level of effort to get the system up and running.

Second was our decision to store our data in a Postgres database as opposed to using a distributed file system such as HDFS. Again, this was done to increase the ease of use and decrease time to set up, although this was done at the sacrifice of ability to handle larger volumes of data efficiently.

Third was our decision to pull data using chron jobs, as opposed to importing streaming data into our database. While it's possible that we will lose out on some data and that our data will not be in real-time, this also greatly decreases our storage requirements.

Installation Instructions

Prerequisites:

Before we get into installation instructions, please make sure you have the following:

1. An UCB MIDS W205 EX2-FULL instance
2. You will also require a working postgress installation
3. A GitHub Account
4. Tableau
5. A working New York Times API key to interact with the NYT API. An API key can be obtained here: https://developer.nytimes.com/signup. Make sure you have selected "Top Stories API. Keep your NYT key handy for the future.
6. Twitter Credentials which will allow you to use the Twitter API and import tweets. To obtain the credentials:
    a. Login to Twitter: https://www.twitter.com
    b. Create a New Twitter App: https://www.apps.twitter.com/
    c. Click on "Keys and Access Token", and then "My Access Token"
    d. You should now have four keys – Consumer Key, Consumer Secret, Access Token, and Access Secret. Keep these handy for the future
7. Your machine is running python 2.7
8. Python Packages:
    a. Tweepy. If you do not have this, it can be installed in your EC2 instance with:
    $ pip install tweepy

Installing:

Initializing and Populating Database:

1. Connect to your EC2 instance
2. Launch Postgres
3. Clone the application directory on your machine (make sure you are in /data):
   $ git clone https://github.com/amkabatznick/MIDS-w205-summer-17-Course-Project.git
4. Run the following:
   export TWITTER_CONSUMER_KEY="twitter_consumer_key"
   export TWITTER_CONSUMER_SECRET="your_twitter_consumer_secret"
   export TWITTER_ACCESS_TOKEN="your_twitter_acess_token"
   export TWITTER_ACCESS_SECRET="your_twitter_access_secret"
   export NYTimesAPI="your_NYT_account"

5. Run the following to initiate the one databases we will be using:
   cd /data/MIDS-w205-summer-17-Course-Project
   $ python create_twitter_db.py
6. Run the following to obtain NYT article metadata. This script parses through each section of the JSON that is returned and stores it in our database:
   $ python NYT_API_Parser.py
7. Run the following:
   $ python twitter_search.py

Connecting to Tableau and Viewing data:

1. Set Postgres permissions to allow external connections:
   $ cd /data/pgsql/data
   $ vim pg_hba.conf
   * Navigate to "IPv4 local connections" and enter (remember to hit "I" beforehand"):
   host all all 0.0.0.0/0 trust
   * Hit ESC, then type ":wq"

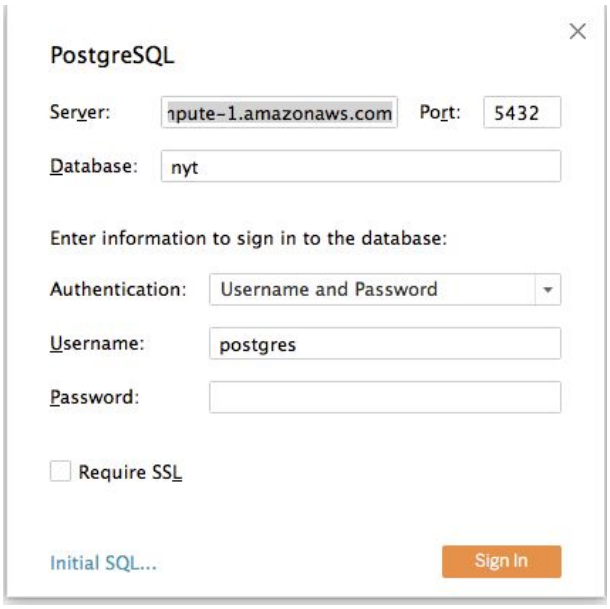
   $ vim postgresql.conf
   * Remove the "#" before "port = 5432" (again, remember to hit "I" before")
   * Hit ESC, then type ":wq"

```
# - Connection Settings -

#listen_addresses = 'localhost'         # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost', '*' = all
                                        # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)
# Note:  Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3     # (change requires restart)
#unix_socket_directory = ''             # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                        # (change requires restart)
#bonjour_name = ''                      # defaults to the computer name
                                        # (change requires restart)
```

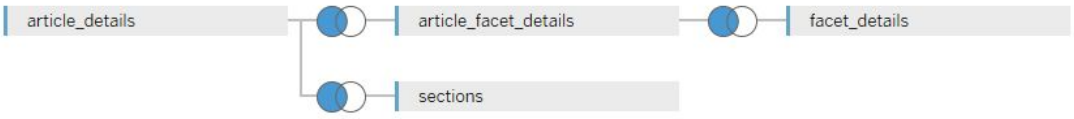
2. We've now allowed connections to Postgres on port 5432. Ensure that your security settings for your instance have port 5432 open.

3. Start Tableau
4. Under Connect, click "To a Server" and click Postgres
5. In the window, populate it so it looks like the following:



Password should be "pass" and the Server is the IPv4 Public IP address

6. Setup table linkage based on the data view you want to produce

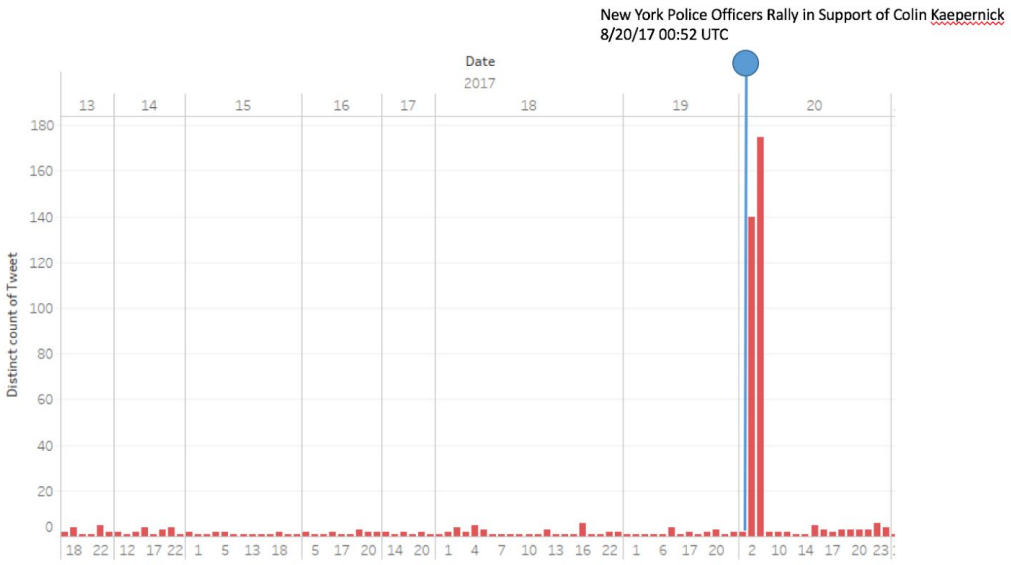


7. Create views on datasheet based on needs

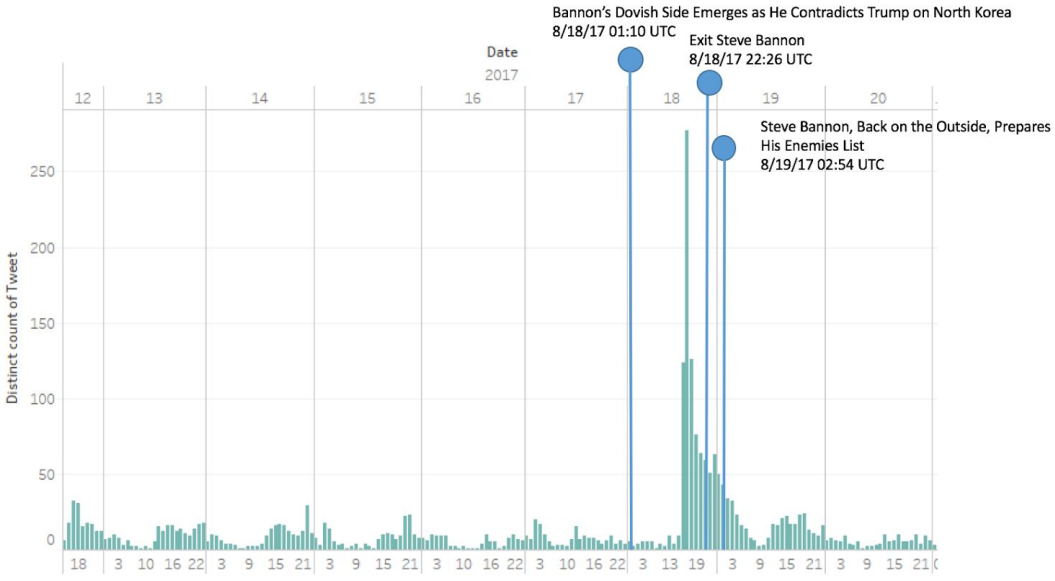
Results

Below are the resulting visuals. The vertical blue lines with bubbles represent the time at which an NYT article was published online, and the various bars represent the Tweet quantity which contained the keywords of interest (e.g. if the article is about Kaepernick, the bars represent Tweets containing "Kaepernick," within 1 Levenshtein distance).

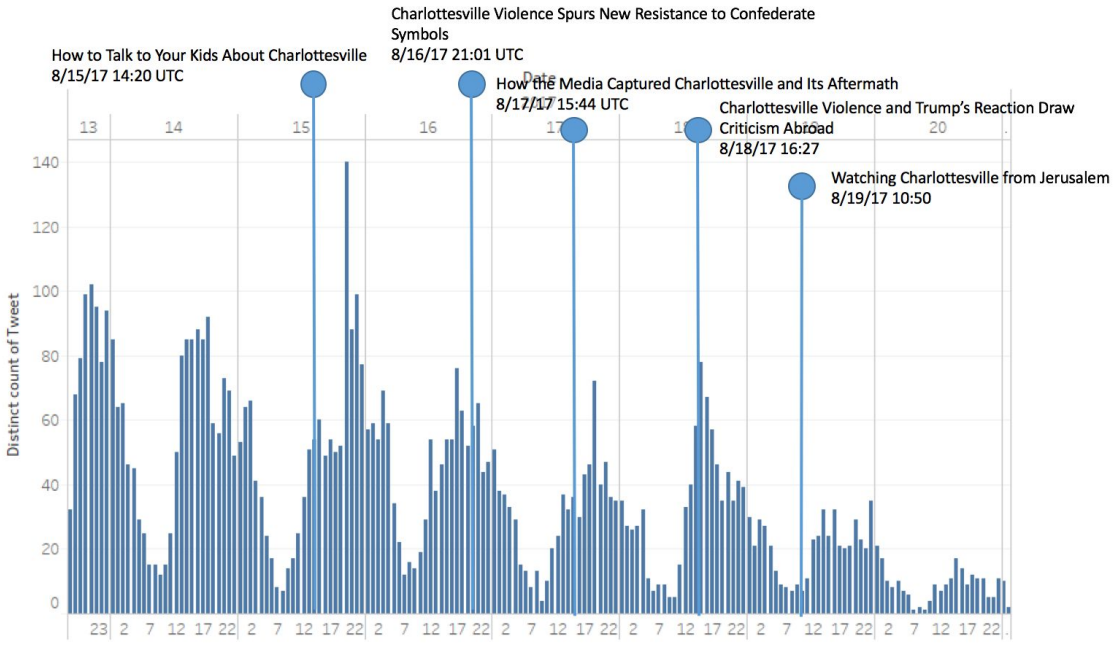
Kaepernick (Sports):



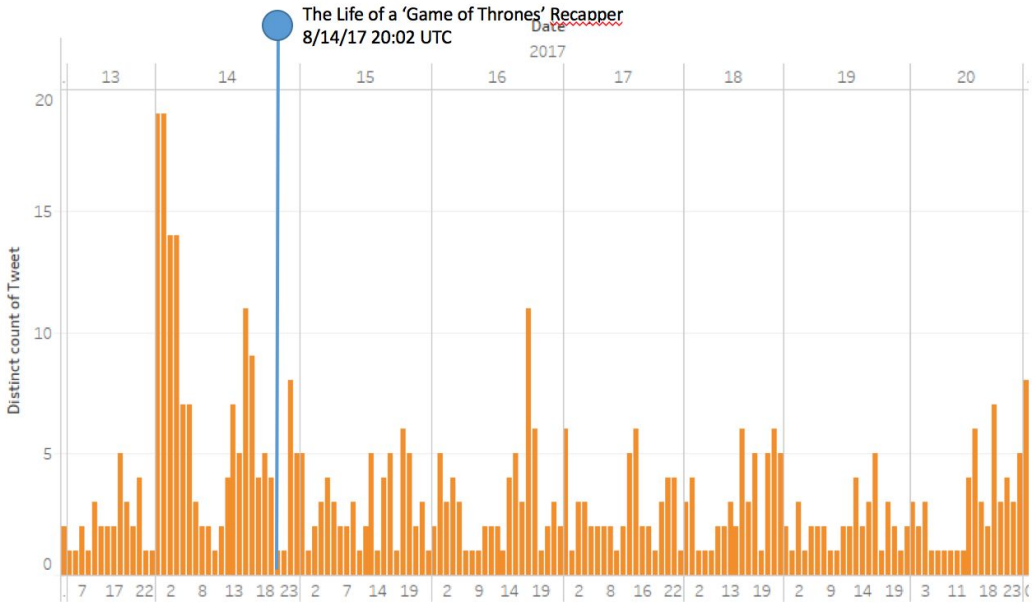
Bannon (Politics):



Charlottesville (National News):



Game of Thrones (Entertainment):



In the case of Kaepernick, who represents the sports events, we see that there is relatively little talk about him on Twitter until the news breaks on NYT that the NYDP police officers support his cause, at which point there is a 2 hour spike before things return to normal, which is comparatively low with respect to the other three keywords.

In the case of Bannon, who represents politics, there is a cyclical pattern in the number of Tweets that contain "Bannon" over a 24 hour cycle up until the point that some NYT articles are released announcing that he was fired, at which point there is a sharp spike before business as usual.

In the case of Charlottesville, which represents national news, there is again a cyclical pattern in Tweets that contain "Charlottesville," although the volume is much higher than that of Bannon's. Additionally, each spike seems to decrease slightly with each day, which makes sense as the event in question occurred over a week ago with little follow up.

In the case of Game of Thrones, which represents entertainment, there is no distinct cyclical pattern, although there is a spike very early Monday (right after a new episode aired), with "normal" being a fairly consistent volume with a few spikes here and there.

With the data we have now, we can see that 1) with regards to individual sports players, the general population does not discuss much about them unless something out of the ordinary happens 2) with regards to political figures, the general population discusses some about them (although even this is hard to tell because Bannon is particularly infamous figure) and much more when a major event occurs involving that figure 3) with regards to national events, there seems to be a lot of talk about them initially, but it seems to steadily decrease unless further developments occur and 4) with regards to entertainment events, there seems to be consistent talk about them, and even more when developments occur, which in the case of this show happens on a weekly basis.

If we were to make a hypothesis explaining our results, it would be that people tend to discuss (at least on Twitter) things that pertain to them the most. In the case of sports figures and political figures, the two don't have too much of an impact on one's everyday life. On the other hand, national news and entertainment are two aspects of life which are seemingly ubiquitous.

Future Considerations

While we were able to successfully initiate this study and build a system to facilitate it, there are a number of improvements that we hope to make in the future.

Firstly, our Twitter API license only allows us pull tweets from one week prior. To build a robust database for any kind of study, we would like to parse and analyze all tweets in the tweet stream – not just 1000 from one city.

With regards to the study itself, we plan to increase the number of news sources so that we more accurately model when an event occurred. At present, we assume that the NYT is a fairly accurate measure of when an event actually occurred, but with more news sources this will be even more accurate, as it's possible that reporters other than the NYT are closer to the source of information.

Next, our current code pulls data from Twitter containing keywords that we need to manually input. Not only does this limit our data pool as we only have one keyword representing a category (e.g. "Bannon" representative of all political figures at this point in time), but also requires a greater level of effort as one needs to look at the current facets on NYT, then choose keywords accordingly. In a future iteration, we plan to create keyword searches automatically based on facets found in news articles. This will increase the automation in our system and also give us a more accurate representation of keywords for each event category.

Finally, while we initially started this study to determine variation in how people talk about different kinds of events, we believe that we have created an initial version of a tool that can also allow researchers to study what it is people like to discuss, and perhaps why people choose to discuss these things (entertainment) rather than others (political figures).