Text Mining Project: Presentation 1

Evan Bradley

Oakland University

edbradley@oakland.edu

November 2016

Overview

Twitter Queries

- #Election2016
- #NotMyPresident
- #ElectionNight
- #ElectionFinalThoughts
- “clinton”
- “trump”

Collected Data

- Around 1,000,000 tweets
- Drawing from hashtags on previous slide.
- Collected from Twitter's stream API
- Roughly 750,000 tweets per 24 hours.

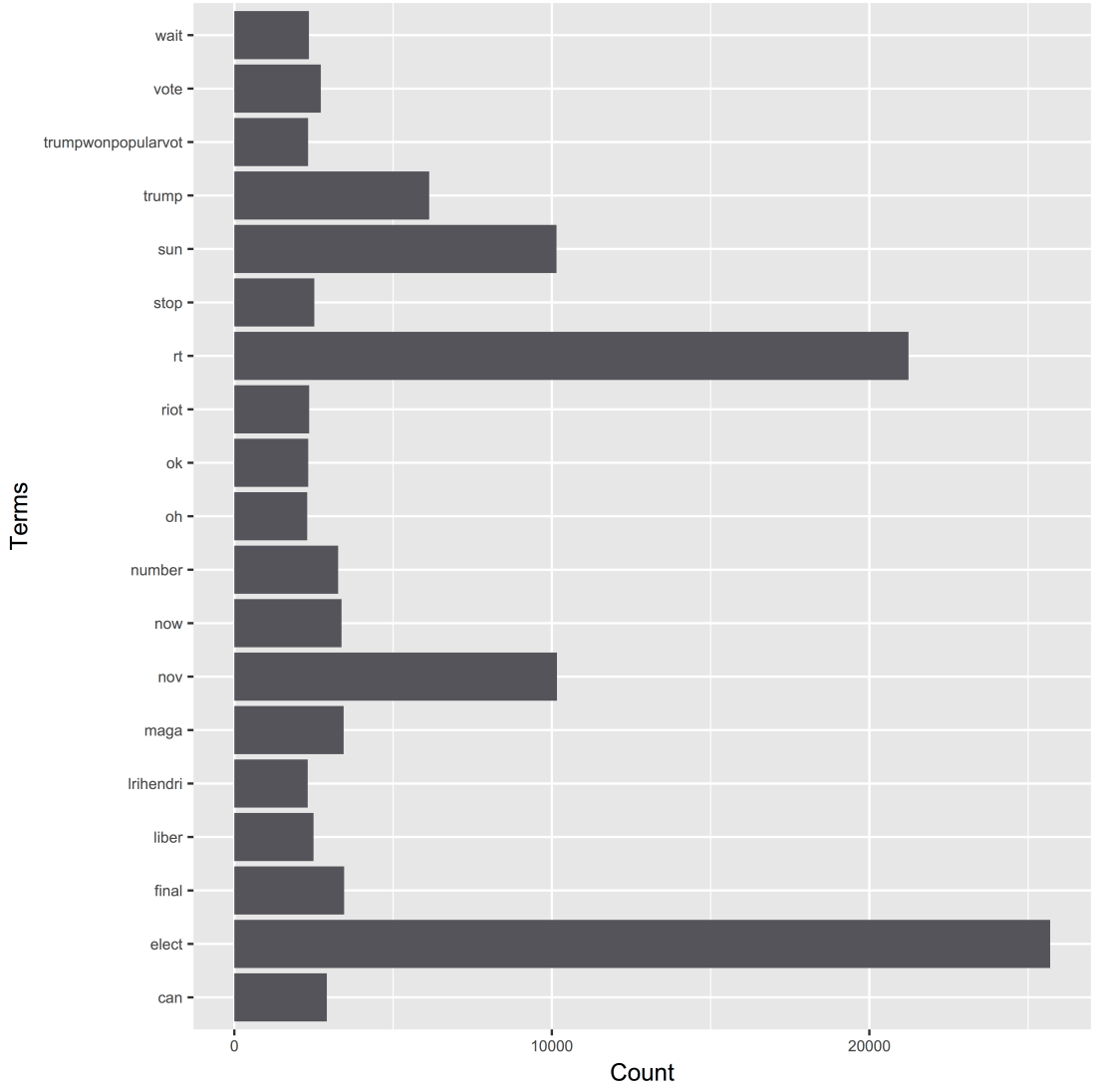
Collection Process

- Using Node.js with the Twit module.
- Responses are written to an output JSON file.
- Node.js also used to trim data down and convert to CSV.
- Database should probably be used for data collection (MongoDB).
- Further pre-processing may be necessary before exporting to CSV.

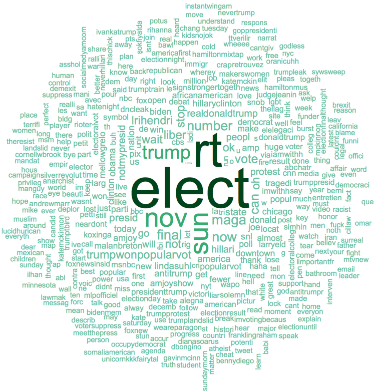
R Packages

- tm: provides text mining processes (cleaning, TDM, DTM).
- SnowballC: word-stemming using Porter's algorithm.
- ggplot2: Plotting library.
- wordcloud: Produces a wordcloud visualization of word frequency.
- sentiment: Performs rudimentary sentiment analysis on data.

Word Frequencies



Wordcloud


elect
rt
nov
sun
trump
maga
final
now
number
realdonaldtrump
trumpwonpopularvot
vote
wait
liber
riot
hillari
america
presid
can
oh
protest
lrihendri

Word Associations

```
> findAssocs(tdm, "trump", 0.2)
$trump
  election  clinton  popularvote  donald electoralcolleg
       0.31     0.28         0.26    0.25            0.22
       won
      0.22

> findAssocs(tdm, "clinton", 0.2)
$clinton
    popularvote electoralcolleg  number  trump  final
           0.55            0.47    0.43   0.28   0.20
```

Sentiment Analysis

```
> sentiments <- sentiment(tweets$text)
> table(sentiments$polarity)

negative  neutral positive
       2      423       63
```

Tasks for Text Mining

- Observe hashtag and topic trends over time.
- Look at sentiment analysis of election over time.
- Observe association of terms between Clinton, Trump, and possibly other figures.
- Observe association between hashtags over time.
- Emoji Analysis