Analyzing Twitter Data

ANALYZING SOCIAL MEDIA DATA IN PYTHON

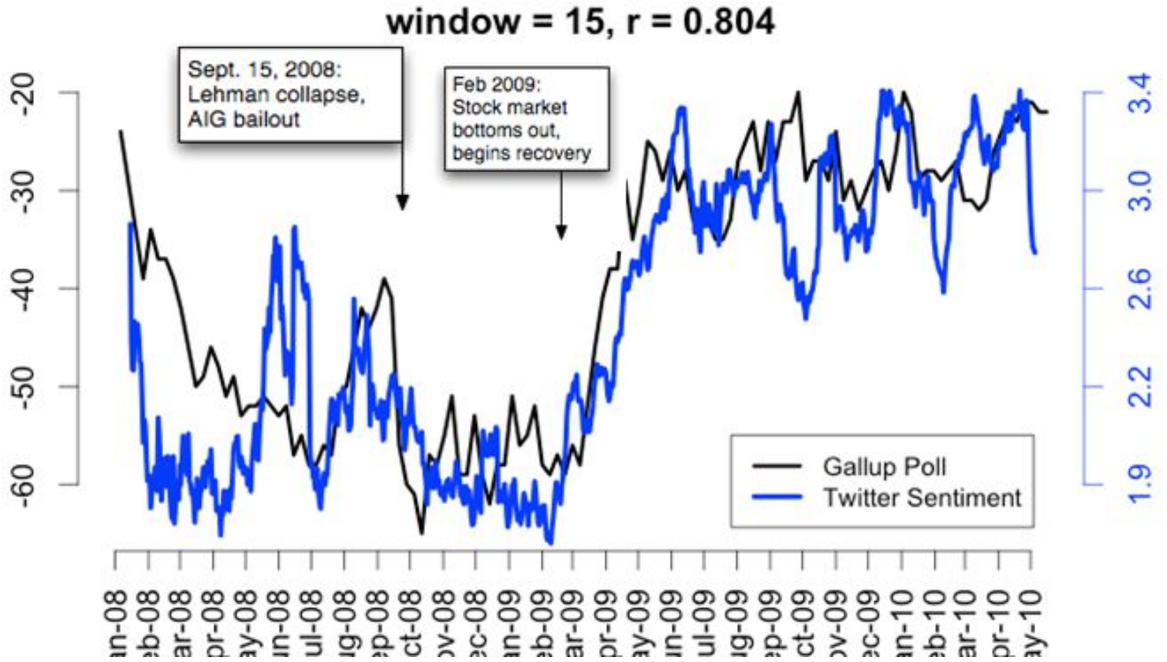
Alex Hanna
Computational Social Scientist

datacamp

Why Analyze Twitter Data?

Twitter sentiment versus Gallup Poll of Consumer Confidence

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010.
From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In ICWSM-2010



Why Analyze Twitter Data?

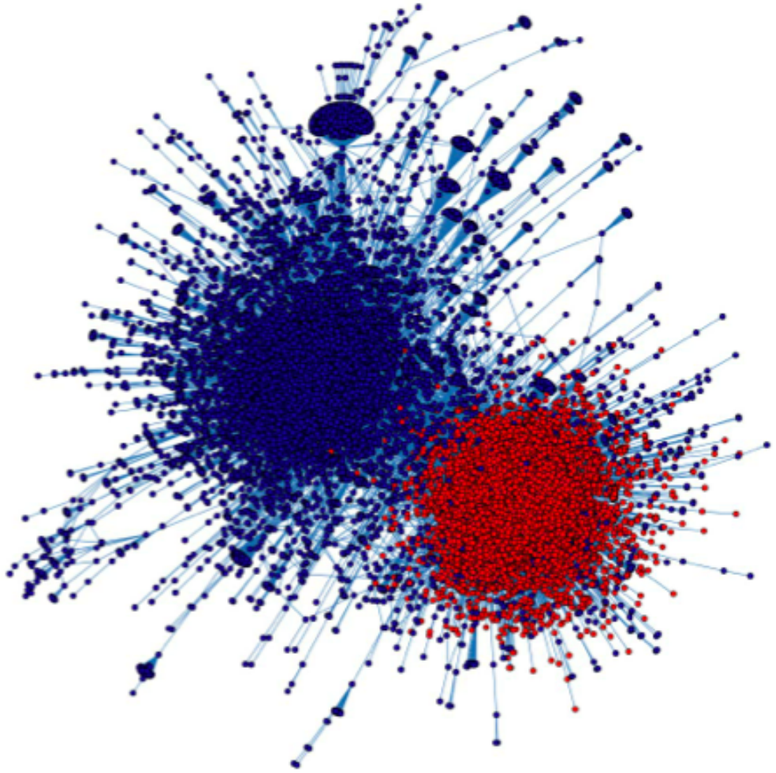


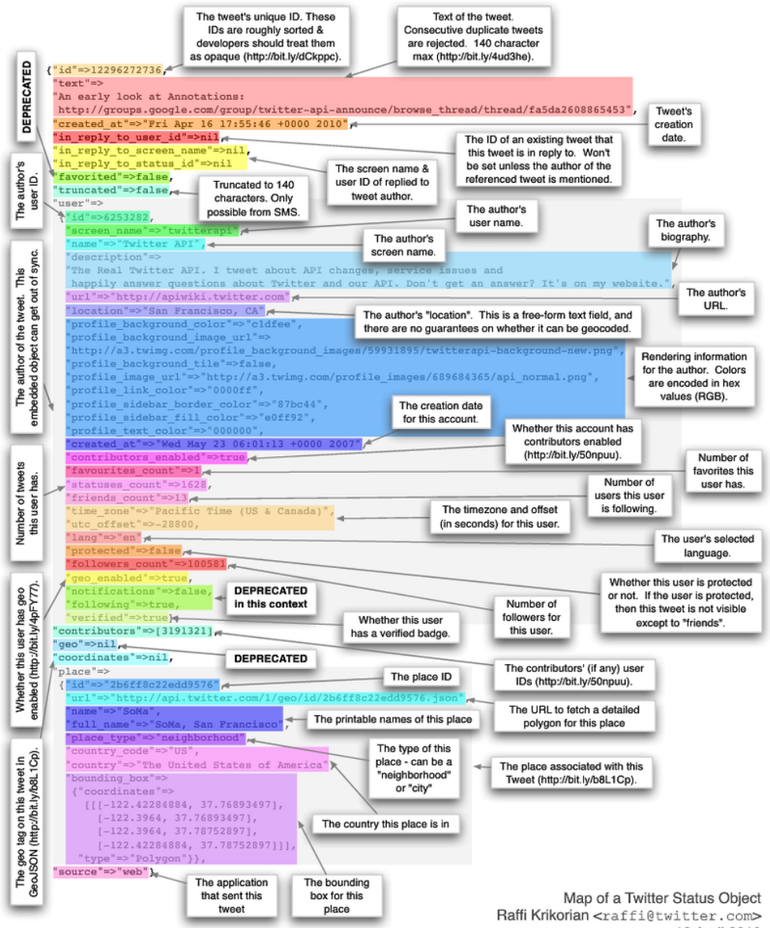
Fig. 2. The political retweet network, laid out using a force-directed algorithm. Node colors reflect cluster assignments (see text).

Source: Conover et al. (2011)

What you can't analyze

- Can't collect data on observers
- Free-level of access is restrictive
  - Can't collect historical data
  - Only a 1% (unverified) sample

What you can analyze



- 1% sample is still a few million tweets
- Within a tweet
  - Text
  - User profile information
  - Geolocation
  - Retweets and quoted tweets

Let's review!

ANALYZING SOCIAL MEDIA DATA IN PYTHON

Collecting data through the Twitter API

ANALYZING SOCIAL MEDIA DATA IN PYTHON

Alex Hanna
Computational Social Scientist

datacamp

Twitter API

- API: Application Programming Interface
  - Method of accessing data
- Twitter APIs
  - Search API
  - Ads API
  - Streaming API

Streaming API

- Streaming API
  - Real-time tweets
- Filter endpoint
  - Keywords
  - User IDs
  - Locations
- Sample endpoint
  - Random sample

Using tweepy to collect data

- `tweepy`
  - Python package for accessing Streaming API

SListener

```
from tweepy import Stream
import time

class SListener(Stream):
    def __init__(self, api = None):
        self.output  = open('tweets_%s.json' %
            time.strftime('%Y%m%d-%H%M%S'), 'w')
        self.api = api or API()
    ...
```

tweepy authentication

```
from tweepy import OAuthHandler
from tweepy import API

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = API(auth)
```

Collecting data with tweepy

```
from tweepy import Stream


listen = SListener(api)
stream = Stream(auth, listen)
stream.sample()
```

Let's practice!

ANALYZING SOCIAL MEDIA DATA IN PYTHON

Understanding Twitter JSON

ANALYZING SOCIAL MEDIA DATA IN PYTHON

Alex Hanna
Computational Social Scientist

datacamp

Contents of Twitter JSON

```
{  "created_at": "Thu Apr 19 14:25:04 +0000 2018",
   "id": 986973961295720449,
   "id_str": "986973961295720449",
   "text": "Writing out the script of my @DataCamp class
            and I can't help but mentally read it back to myself in
            @hugobowne's voice.",
   "retweet_count": 0,
   "favorite_count": 1,
      ...  }
```

- How many retweets, favorites
- Language
- Reply to which tweet
- Reply to which user

Child JSON objects

```
{
        "user": {
            "id": 661613,
            "name": "Alex Hanna, Data Witch",
            "screen_name": "alexhanna",
            "location": "Toronto, ON",
            ...
         }
}
```

Places, retweets/quoted tweets, and 140+ tweets

- `place` and `coordinate`
  - contain geolocation
- `extended_tweet`
  - tweets over 140 characters
- `retweeted_status` and `quoted_status`
  - contain all tweet information of retweets and quoted tweets

Accessing JSON

```
import json


tweet_json = open('tweet-example.json', 'r').read()
tweet = json.loads(tweet_json)
tweet['text']
```

Child tweet JSON

```
tweet['user']['screen_name']
tweet['user']['name']
tweet['user']['created_at']
```

Let's practice!

ANALYZING SOCIAL MEDIA DATA IN PYTHON