Training session: Collecting Twitter Data

Pablo Barberá
Networked Democracy Lab
University of Southern California

January 31, 2017

materials: github.com/NetDem-USC/training

Collecting Social Media Data

Two different methods:

1. Screen scraping: extract data from source code of website
2. Web APIs (application programming interface): use a set of structured https requests that return JSON or XML files

Collecting Social Media Data

Two different methods:

1. Screen scraping: extract data from source code of website
2. Web APIs (application programming interface): use a set of structured https requests that return JSON or XML files

Types of APIs:

1. RESTful APIs: queries for static information in current moment (e.g. user profiles, posts, etc.)
2. Streaming APIs: changes in users' data in real time (e.g. new messages, deletions, etc.)

Collecting Social Media Data

Two different methods:

1. Screen scraping: extract data from source code of website
2. Web APIs (application programming interface): use a set of structured https requests that return JSON or XML files

Types of APIs:

1. RESTful APIs: queries for static information in current moment (e.g. user profiles, posts, etc.)
2. Streaming APIs: changes in users' data in real time (e.g. new messages, deletions, etc.)

Potential issues

1. Rate limits: restrictions on number of API calls by user and period of time (APIs are expensive!)
2. Ongoing debate on replication of social science research using social media data

Connecting with an API

Constructing a REST API call

- Baseline URL: http://graph.facebook.com/
- Parameters: ?ids=barackobama,johnmccain

Connecting with an API

Constructing a REST API call

- Baseline URL: http://graph.facebook.com/
- Parameters: ?ids=barackobama,johnmccain

Response often in JSON format. (example)

Connecting with an API

Constructing a REST API call

- Baseline URL: http://graph.facebook.com/
- Parameters: ?ids=barackobama,johnmccain

Response often in JSON format. (example)

Authentication

- Most common is an open standard called OAuth
- Connections without sharing username and password, only temporary tokens that can be refreshed
- httr package in R implements most cases (examples)

Interacting with social media APIs

R packages

- Twitter: streamR for Streaming, netdemR for REST,
- Facebook: Rfacebook

Interacting with social media APIs

R packages

- Twitter: streamR for Streaming, netdemR for REST,
- Facebook: Rfacebook

Why R? Most common programming language in data science, 5,000+ packages, great documentation, “it just works”

Interacting with social media APIs

R packages

- Twitter: streamR for Streaming, netdemR for REST,
- Facebook: Rfacebook

Why R? Most common programming language in data science, 5,000+ packages, great documentation, “it just works”

Other APIs: CRAN Task View for Web Technologies and Services

Interacting with social media APIs

R packages

- Twitter: streamR for Streaming, netdemR for REST,
- Facebook: Rfacebook

Why R? Most common programming language in data science, 5,000+ packages, great documentation, “it just works”

Other APIs: CRAN Task View for Web Technologies and Services

Equivalent libraries for python, java, ruby... whatever works for you!

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
    - Queries for specific information about users and tweets

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
   - Queries for specific information about users and tweets
   - Examples: user profile, list of followers and friends, tweets generated by a given user ("timeline"), users lists, etc.

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
   - Queries for specific information about users and tweets
   - Examples: user profile, list of followers and friends, tweets generated by a given user ("timeline"), users lists, etc.
   - R library: netdemR (also twitteR, rtweet)

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
   - Queries for specific information about users and tweets
   - Examples: user profile, list of followers and friends, tweets generated by a given user ("timeline"), users lists, etc.
   - R library: netdemR (also twitteR, rtweet)
2. Streaming API:

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
   - Queries for specific information about users and tweets
   - Examples: user profile, list of followers and friends, tweets generated by a given user ("timeline"), users lists, etc.
   - R library: netdemR (also twitteR, rtweet)
2. Streaming API:
   - Connect to the "stream" of tweets as they are being published

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
   - Queries for specific information about users and tweets
   - Examples: user profile, list of followers and friends, tweets generated by a given user ("timeline"), users lists, etc.
   - R library: netdemR (also twitteR, rtweet)
2. Streaming API:
   - Connect to the "stream" of tweets as they are being published
   - Three streaming APIs:

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
   - Queries for specific information about users and tweets
   - Examples: user profile, list of followers and friends, tweets generated by a given user ("timeline"), users lists, etc.
   - R library: netdemR (also twitteR, rtweet)
2. Streaming API:
   - Connect to the "stream" of tweets as they are being published
   - Three streaming APIs:
     2.1 Filter stream: tweets filtered by keywords

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
   - Queries for specific information about users and tweets
   - Examples: user profile, list of followers and friends, tweets generated by a given user ("timeline"), users lists, etc.
   - R library: netdemR (also twitteR, rtweet)
2. Streaming API:
   - Connect to the "stream" of tweets as they are being published
   - Three streaming APIs:
     2.1 Filter stream: tweets filtered by keywords
     2.2 Geo stream: tweets filtered by location

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
   - Queries for specific information about users and tweets
   - Examples: user profile, list of followers and friends, tweets generated by a given user ("timeline"), users lists, etc.
   - R library: netdemR (also twitteR, rtweet)
2. Streaming API:
   - Connect to the "stream" of tweets as they are being published
   - Three streaming APIs:
     2.1 Filter stream: tweets filtered by keywords
     2.2 Geo stream: tweets filtered by location
     2.3 Sample stream: 1% random sample of tweets

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
   - Queries for specific information about users and tweets
   - Examples: user profile, list of followers and friends, tweets generated by a given user ("timeline"), users lists, etc.
   - R library: netdemR (also twitteR, rtweet)
2. Streaming API:
   - Connect to the "stream" of tweets as they are being published
   - Three streaming APIs:
     2.1 Filter stream: tweets filtered by keywords
     2.2 Geo stream: tweets filtered by location
     2.3 Sample stream: 1% random sample of tweets
   - R library: streamR

Twitter APIs

Two different methods to collect Twitter data:

1. REST API:
   - Queries for specific information about users and tweets
   - Examples: user profile, list of followers and friends, tweets generated by a given user ("timeline"), users lists, etc.
   - R library: netdemR (also twitteR, rtweet)
2. Streaming API:
   - Connect to the "stream" of tweets as they are being published
   - Three streaming APIs:
     - 2.1 Filter stream: tweets filtered by keywords
     - 2.2 Geo stream: tweets filtered by location
     - 2.3 Sample stream: 1% random sample of tweets
   - R library: streamR

Important limitation: tweets can only be downloaded in real time (exception: user timelines, $\sim$ 3,200 most recent tweets are available)

Anatomy of a tweet

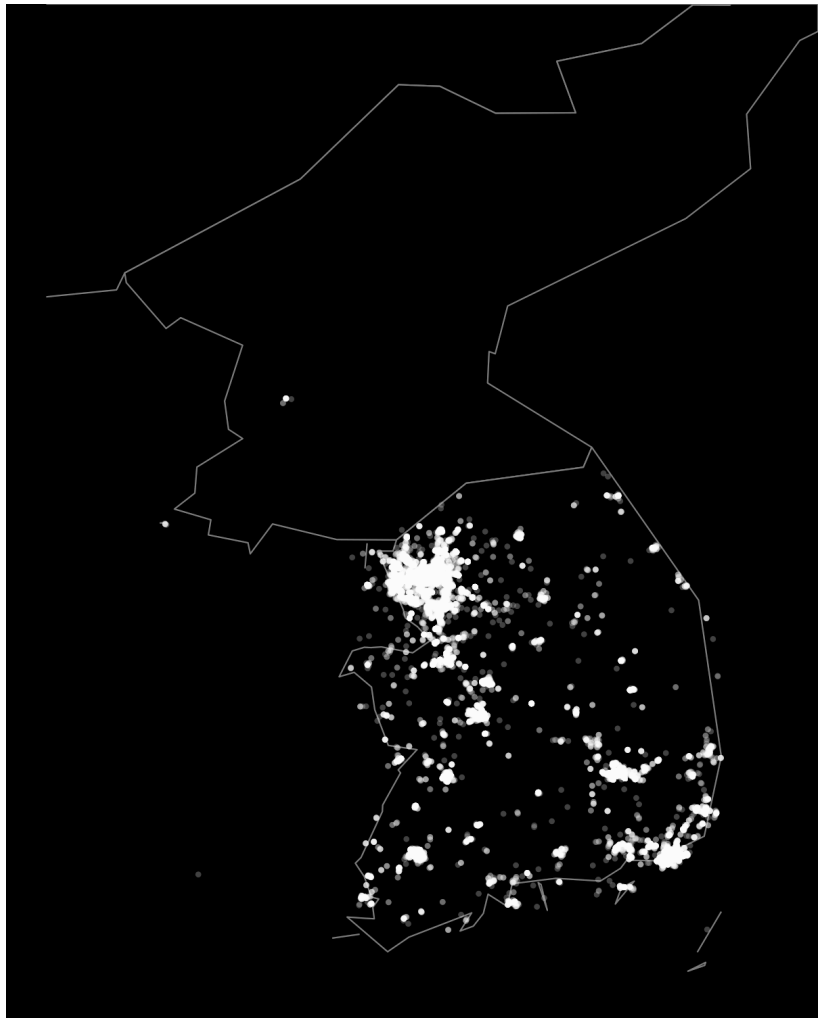
Anatomy of a tweet

Tweets are stored in JSON format:


```
{ ”created_at”: ”Wed Nov 07 04:16:18 +0000 2012”,
  ”id”: 266031293945503744,
  ”text”: ”Four more years. http://t.co/bAJE6Vom”,
  ”source”: ”web”,
  ”user”: {
    ”id”: 813286,
    ”name”: ”Barack Obama”,
    ”screen_name”: ”BarackObama”,
    ”location”: ”Washington, DC”,
    ”description”: ”This account is run by Organizing for Action staff.
       Tweets from the President are signed -bo.”,
    ”url”: ”http://t.co/8aJ56Jcemr”,
    ”protected”: false,
    ”followers_count”: 54873124,
    ”friends_count”: 654580,
    ”listed_count”: 202495,
    ”created_at”: ”Mon Mar 05 22:08:25 +0000 2007”,
    ”time_zone”: ”Eastern Time (US & Canada)”,
    ”statuses_count”: 10687,
    ”lang”: ”en” },
  ”coordinates”: null,
  ”retweet_count”: 756411,
  ”favorite_count”: 288867,
  ”lang”: ”en”
}
```

Streaming API

- Recommended method to collect tweets
- Potential issues:
  - Filter streams have same rate limit as spritzer (1% of all tweets)
  - Stream connections tend to die spontaneously. Restart regularly.
  - Lots of invalid content in stream. If it can't be parsed, drop it.
- My workflow:
  - Amazon EC2, cloud computing
  - Cron jobs to restart R scripts every hour.
  - Save tweets in .json files or in MongoDB.
  - For large .json files, preprocess with python (see: github.com/pablobarbera/pytwools)

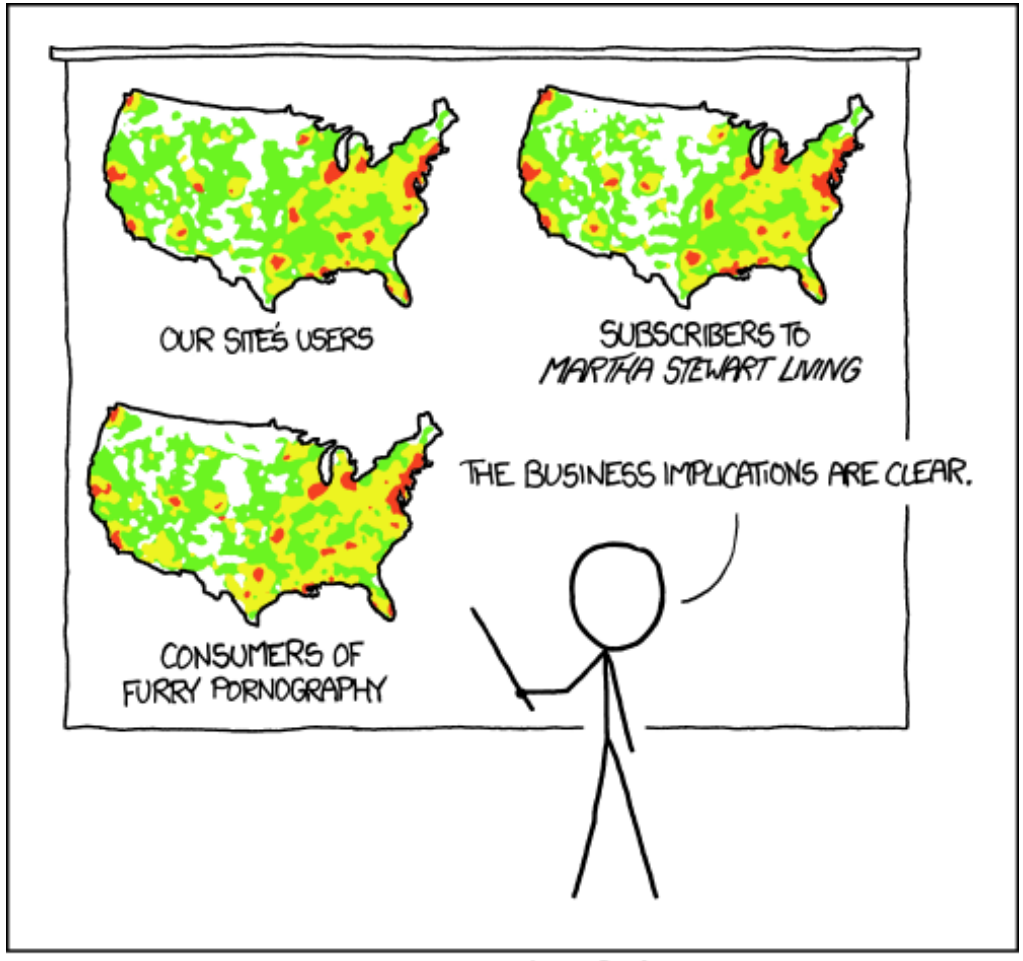


Tweets from Korea: 40k tweets collected in 2014 (left)
Korean peninsula at night, 2003 (right). Source: NASA.

Who is tweeting from North Korea?

Twitter user: @uriminzok_engl

But remember...



PET PEEVE #208:
GEOGRAPHIC PROFILE MAPS WHICH ARE BASICALLY JUST POPULATION MAPS