How to Create a Dataset from Social Media: Theory and Demonstration

Richard N. Landers
Old Dominion University
@rmlanders | rmlanders@odu.edu
CARMA
October 2017
Agenda/Learning Objectives

1. Foundational Questions
   - Why scrape social media?
   - What are the pros and cons of social media data sources?

2. Technical Overview
   - What steps are involved in scraping social media?
   - How are Facebook and Twitter accessed?

3. Sample R Code
   - Facebook
   - Twitter

4. Practical Concerns
   - How to learn this skillset
   - Ethical concerns and legal risks
Foundational Questions

Why scrape social media?
What are the pros and cons of social media data sources?
What is machine learning and how do I use it?
Why scrape social media?

- **What is social media?**
  - A consequence of the Web 2.0 movement toward interactivity on the internet
    - “user generated content”

- **What does user-generated content entail?**
  - **purposive data**
    - user profiles
    - content
  - **incidental metadata**
    - trail of breadcrumbs
What are social media data?

- So psychologically, what are social media data?
  - behaviors, the products of person-situation interactions
  - You can also conceptualize social media data as measurement occasions, similar to a survey scale

- Instead of a "survey" stimulus, you have a "social media website" stimulus
What can I do with scraped data?

- Profile data are easy: mostly demographics.
- Text data is commonly subjected to follow-up data complexity reduction techniques
  - Or don’t reduce, if you have enough data and don’t want to.
What can I do with text data?

- **Linguistic Inquiry and Word Count (LIWC)**
  - Outputs an enormous variety of summary statistics about text, including linguistic (types of words), psychological (traits), high-level (e.g., authenticity, emotional tone)
  - See Tausczik & Pennebaker (2010)

- **Sentiment**
  - Uses existing lexica to classify words as positive or negative (such as LIWC)
  - The Harvard General Inquirer (from Stone, Dunphry, Smith & Ogilvie, 1966)

- **Topic Analysis**
How do I justify using social media data?

- Develop a **data source theory**: a list of your assumptions about a data source:
  - **Data origin/population characteristics**
    - Why does this website exist?
    - Why would someone visit?
    - Why would someone contribute?
    - What type of data is created?
    - Do users pay to participate?
    - Are users restricted in the kind of content they can contribute?

- Data source theories are the core concept in **theory-driven web scraping**

- **Data structure**
  - How are target constructs represented visually and in code?
  - Is there inconsistency in how target constructs are represented?
  - Where do data appear?
  - How is content represented?
  - Is content consistently available?

Data Source Theories Imply Testable Predictions

- Make predictions based upon what you think must be true to create a complete data source theory with testable predictions (i.e., hypotheses).

- Example
  - RQ: How is political engagement represented in tweets?
  - H: Twitter posts containing the names of politicians represent political engagement.

- In traditional data collection, we have these same assumptions but they are generally difficult or impossible to test.
  - Content validation is relatively easy.
Creating a Facebook Data Source Theory

- **Facebook**
  - The data you can scrape vary based upon who you are and what access you have obtained for yourself.
  - In practice, there are two ways to get around this:
    - Scrape content from public groups/pages
    - Create an app that people sign up for and scrape profile content
  - There are **time limitations**.
  - 58% of all US adults have a Facebook account; 81% of Internet-users have one.
Creating a Twitter Data Source Theory

- **Twitter**
  - Almost all profiles are public, so that’s much easier.
  - Birthdays may be available.
  - Geographic data is available, sort of.
  - Search tools don’t allow unrestricted access; there are per-query access limits.
  - 7% of US adults, but 79% of users are outside the US
Technical Overview

What steps are involved in scraping social media?
Five Steps to Execute a Web Scraping Project

1. Identify and pre-emptively evaluate potential sources of information
   - Assumes you already have a RQ/H and some constructs in mind
   - Don’t necessarily limit yourself to Twitter and Facebook – any webpage can potentially be used
   - Consider construct validity at every step
   - Create a data source theory
     - Think counterfactually: “If X isn’t true, my conclusions from this data source will be invalid.”
     - Write it down.
     - Develop specific hypotheses that your theory suggests and figure out which ones you can test (specifying assumptions vs. hypotheses).
Five Steps to Execute a Web Scraping Project

2. Develop a coding system
   a) Identify the specific constructs you want to assess
   b) Determine how those constructs are represented from a technical standpoint
      a) Are they recoded from text?
      b) Are they structured pieces of information?
      c) Where are they? How are they represented?
3. Code a scraper and probably a crawler
   - **Scrapers** convert raw output into datasets.
   - **Crawlers** locate more data based on scraped content.

   - If an API is available, you want to scrape API responses
     - Returns **structured** data with variables pre-defined
     - Will probably need multiple calls to grab large datasets
     - Legally unambiguous

   - If an API is not available, you’ll scrape HTML directly
     - Returns **unstructured** data, and is a lot more work
     - Legally ambiguous in some cases
So what’s an API?

- **API: Application Programming Interface**
  - A data gateway into someone else’s system, created by the owner of those data
  - Almost universally intended for real-time access by other websites, but you can take advantage of it too
  - Requires learning API documentation – they’re all different

- Let’s start easy. I’ve created an that adds two numbers, x & y.
  - [http://scraping.tntlab.org/add.php](http://scraping.tntlab.org/add.php)
  - [http://scraping.tntlab.org/add.php?x=1](http://scraping.tntlab.org/add.php?x=1)
  - [http://scraping.tntlab.org/add.php?x=1&y=muffin](http://scraping.tntlab.org/add.php?x=1&y=muffin)
  - [http://scraping.tntlab.org/add.php?x=1&y=8](http://scraping.tntlab.org/add.php?x=1&y=8)
  - [http://scraping.tntlab.org/add.php?x=1&y=8&format=csv](http://scraping.tntlab.org/add.php?x=1&y=8&format=csv)
API Call Example #1

scraping.tntlab.org/add.php

You didn't set x and y!
API Call Example #2

```
scraping.tntlab.org/add.php?x=1
```

You didn't set x and y!
API Call Example #3

scraping.tntlab.org/add.php?x=1&y=muffin

x and y must be numbers!
API Call Example #4

scraping.tntlab.org/add.php?x=1&y=8

The sum of 1 and 8 is 9!
API Example Call #5

scraping.tntlab.org/add.php?x=1&y=8&format=csv

1,8,9
What format of data do APIs provide?

- The output of an API can be in essentially any format, but JSON is the most common: JavaScript object notation

- Both Facebook and Twitter return JSON files

- These APIs also have rate limits in terms of the number of requests you are allowed to send and how quickly
  - Twitter limits to 180 calls every 15 minutes for simple requests and 15 calls every 15 minutes for complex ones. For example, only 25 tweets can be returned per simple call, so up to 4500 tweets per 15 minutes
JSON Output from Facebook API
Experiment with the Facebook API

- Go to [http://developers.facebook.com/tools/explorer](http://developers.facebook.com/tools/explorer) (you’ll need to be logged into Facebook)
- Generate a token for yourself (“Get Token”)
  - This token will have the permissions that your Facebook account has
- Craft a request using the Explorer, such as:
  - 853552931365745/feed
- Create this same request in your web browser by going to:
  - [https://graph.facebook.com/853552931365745/feed?access_token=xxxx](https://graph.facebook.com/853552931365745/feed?access_token=xxxx) (but replace xxxx with the copy/pasted token you generated)
Five Steps to Execute a Web Scraping Project

4. Clean the data and revise the data source theory
   - Once you have your data in hand, run all hypothesis tests possible from your data source theory
   - You will almost certainly identify problems with your coding system at this stage; time to revise
Five Steps to Execute a Web Scraping Project

5. Analyze!

- Natural language processing
- Data simplification
- Predictive modeling
- Simple profile reporting / qualitative analyses
Sample R Code

Facebook
Twitter
Facebook: Step 1

- Open Facebook retrieval library

```
library(Rfacebook)
```

- At this point, go grab an API access token from the Graph API Explorer
- Go to https://developers.facebook.com/tools/explorer/ and click "Get Token"
- Store this value in the next variable

```
token <- "EAACEdEose0cBAOXvI1Mww11ntZCWBeTW50ksizBfVfW9Qks0jhhKlQyikHHLloj0p4ybEB/
```

- Grab public group or page names so that you can determine Facebook's ID number
- The group you actually want to grab data from (must be OPEN)

```
ids <- searchPages("advancement of research methods and analysis", token=token)
```
Facebook: Output from Step 1

```r
> glimpse(ids)
Observations: 1  
Variables: 16
$id  <chr>  "109746019101267"
$about <chr>  "CARMA"
$category <chr>  "Educational Research Center"
$description <chr>  "CARMA"
$general_info <lgl>  NA
$likes <dbl>  734
$link <chr>  "https://www.facebook.com/NebraskaCARMA/"
$city <chr>  "Lincoln"
$state <chr>  "NE"
$countr $latitude <dbl>  47.87679
$longitude <dbl>  -97.11993
$advancement of Research Methods...
$name <chr>  "Consortium for the Advancement of Research Methods"
$website <chr>  "http://business.unl.edu/outreach/carma/"
```
Facebook: Step 2

```r
> group <- getGroup(group_id=109746019101267, token=token, n = 200)

> glimpse(group)
Observations: 200
Variables: 11
$ from_id <chr> "109746019101267", "109746019101267", "109746019101267", ...
$ from_name <chr> "Consortium for the Advancement of Research Methods and ..."$ message <chr> "We're 10 days out from the next CARMA webcast! Dr. Dona...
$ created_time <chr> "2017-10-10T13:27:40+0000", "2017-09-22T14:05:59+0000", ...
$ type <chr> "link", "photo", "video", "status", "status", "status", ...
$ link <chr> "http://business.unl.edu/outreach/carma/webcast-programs..."
$ id <chr> "109746019101267_1492747520801103", "109746019101267_147...
$ story <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ likes_count <dbl> 2, 15, 24, 0, 0, 0, 0, 12, 3, 0, 0, 0, 2, 7, 3, 9, 6, 5, ...
$ comments_count <dbl> 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ shares_count <dbl> 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
```
Complete Twitter Program

# Open Twitter retrieval library
library(twitteR)

# At this point, create an "App" on Twitter after logging in by going to
# http://apps.twitter.com and "creating an application"
# Once you've created an application, open its settings, go to Keys and
# Access Tokens, generate, then copy/paste the four strings required here
consumer_key <- "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
consumer_secret <- "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
access_token <- "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
access_secret <- "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

# Let's grab all Twitter posts about #carma that we can and convert to a data
# frame
busiSearch <- searchTwitter("#carma", n=200)
busiSearch_df <- twListToDF(busiSearch)
Practical Concerns

How to learn this skillset
Ethical concerns and legal risks
Why Do This Yourself?

- The old way
  - URAs hand-coding text (~2 minutes per subject; with 2 coders, at 60 per hour, coding 500 entries would take 8.3 hours of coding time)

- The new way
  - In ~8 hours, we captured >100,000 text entries

- If you don’t want to code, you can’t use APIs
- If you already know R, you’ll find API calls fairly easy

- You should really learn R anyway
How to Learn This Skillset

- There are two major skillsets involved:
  - HTML, to know how web pages are structured (for page scraping)
  - Statistical programming (e.g., in R or Python) in general, to be able to run algorithms
    - Web scraping libraries in R or Python, to run specific extraction algorithms
    - Machine learning libraries in R, Python, SPSS, etc to run analytic algorithms

- To learn R, Python, and their libraries:

- To learn natural language processing fundamentals:
Ethics and Legal Risks - Hacking

- Don’t look like a hacker and you won’t be treated like one (honeypots)
- Remember to read API documentation (and to authenticate)
- Look for tutorials/examples of those that have done this before
- Don’t go hunting for statistical significance with the standard stats toolkit
Ethics and Legal Risks – Fair and Commercial Use

- **Fair use**: Often unclear what is usable
  - *Harvesting data when a policy is in place explicitly forbidding it* is definitely unethical and probably illegal (see eBay v Bidder’s Edge, 2000 and Ticketmaster Corp vs Tickets.com, 2000)

- *Harvesting data behind a login wall without a policy* is probably unethical and probably illegal (APIs protect you from this)

- *Harvesting public data that is not explicitly linked anywhere* is probably unethical and probably illegal (see the story of Andrew Auernheimer, aka weev)

- *Harvesting public social media data that is plainly visible* through simple web browsing might be ethical or unethical but is *probably legal*

- A case to watch: HiQ Labs v LinkedIn
Thank You!

For easily digestible descriptions of new talent analytics technology, see my column in the Industrial-Organizational Psychologist!

For example, natural language processing:
http://www.siop.org/tip/april17/crash.aspx

Richard N. Landers
Old Dominion University

@rmlanders | rmlanders@odu.edu

CARMA
October 2017