

Application programming interfaces for social scientists: A collaborative review

A roundtable organized by Camille Landesvatter, Lion Behrens and Paul C. Bauer

MZES Social Science Data Lab

September 21, 2022, 13:45-15:15

Hybrid event [A5, 6, Room A231 + Zoom]

Structure of today's event

- Part 1: General Information on APIs and the collaborative review
- Part 2: Introduction of individual chapters and their authors
- Part 3: The Use of APIs for Social Science Research: Opportunities and Limitations
- Other Questions

Part 1: General Information on APIs and on the collaborative review

Web APIs in the context of Social Science Research

(Web) API = Application Programming Interface; a technology that includes a set of tools allowing users to send and receive data or functionality through a documented interface

APIs in the context of Social Science Research

- Many today famous APIs originated in context of Web 2.0
 - Youtube (2005) and Twitter API (2006)
 - Facebook API (2004, API: 2006)
- increasing number of API-based research in social science disciplines
 - main use case: data access (“data collection”)
 - also, innovative research questions that require creative solutions for the operationalisation of theoretical concepts
- APIs allow flexible and customizable access to data
- before: manually write web scraping code (e.g. with RSelenium) to collect unstructured data from the web

APIs in the context of Social Science Research: Twitter API



Developer Platform

Products ▾

Docs ▾

Use Cases ▾

Community ▾

Do research > Academic research

Academic research

From social science to computer science, Twitter data can advance research objectives on topics as diverse as the global conversations happening on Twitter.

Website project:
APIs for social scientists:
A collaborative review

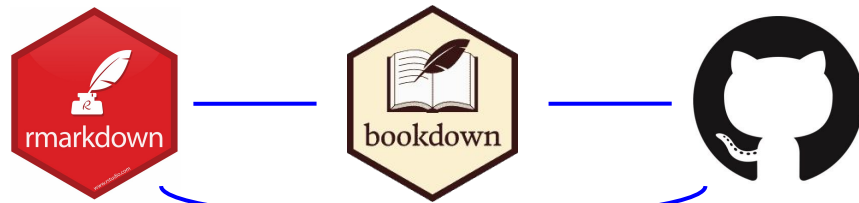
Website project: APIs for social scientists: A collaborative review

[https://bookdown.org/paul/
apis_for_social_scientists/](https://bookdown.org/paul/apis_for_social_scientists/)



Website project: Technical Details

Website project: Technical details



How is the book set up?

This book is set up as a **collaborative project** using work organization methods that were initially developed in the **open source software community** (Konrad, 2022)

- based on one [R Markdown](#) file for each individual chapter
- hosted via [R bookdown](#) which knits all chapters together
- stored as a public repository on [Github](#):
github.com/paulcbauer/apis_for_social_scientists_a_review

This means:

- ➡ open source
- ➡ everyone can adapt chapters
- ➡ everyone can contribute

Website project: Technical details

Issue: Authentication

How can an **author** push a reproducible [Rmd](#) file to [GitHub](#) without exposing sensitive credentials? (API keys)

- ➡ Option 1: store keys locally in [JSON](#) format
 - from Chapter 9 Google Natural Language API

```
gl_auth("./your-key.json")
```

- ➡ Option 2: define user-level environment variables
 - from Chapter 2 Best Practices

```
Sys.setenv(MYSECRET = "MbOS6cQhhFkwETXKur-L9rN")
```

```
Sys.getenv("MYSECRET")
```

```
## [1] "MbOS6cQhhFkwETXKur-L9rN"
```

Issue: Reproducibility

How can a **reader** reproduce the code of a chapter without signing up for the API?

- ➡ Local copy of API response is stored under `apis_for_social_scientists_a_review/data`
- ➡ See “Caching” in Chapter 2 Best Practices

Website project: Technical details

Reproducibility: A closer look

Challenge: 21 chapters introduce different APIs using [R](#) Coding.

- R *packages* get updated (-> change in functionality)
- R *version* gets updated (-> packages incompatible)
- *API functionalities* get updated
 - access restrictions
 - changes in rate limit

Solution 1: Use checkpoint package from [Reproducible R Toolkit](#)

```
p_load('checkpoint')  
checkpoint("2022-08-03")
```

- daily snapshots of CRAN which are mirrored by the RRT-team on a separate server
- command will install package snapshots in their version of last reproducibility check

Solution 2: *error_control.R*

Website project: Technical details

Reproducibility: A closer look

- ➡ R script to
 - loop over all .Rmd files
 - extract raw R code
 - output 1 and [error message] if there is an error

```
# identify errors in script
for (file in 1:length(Rmdfiles)) {

  # extract R code from Rmd, run R code
  tryCatch({
    knitr::purl(Rmdfiles[file]) # compress .Rmd to .R
    source(substring(Rmdfiles[file], 1, nchar(Rmdfiles[file])-2)) # run .R
  },

  # store error and error message
  error = function(e) {
    row_id <- which(error_control[, "file"] == Rmdfiles[file])
    error_control[row_id, "error"] <- 1
    error_control[row_id, "error_message"] <- list(e[1])
  }) # end tryCatch
} # end for
```

Figure: Key operations in error_control.R

Website project: Technical details

Reproducibility: A closer look

	file	error	error_message
1	Chapter_Best_practices.Rmd	0	NA
2	Chapter_Ckan_api.Rmd	0	NA
3	Chapter_Crowdtangle_api.Rmd	1	object 'listids' not found
4	Chapter_Facebook_ads_library_api.Rmd	1	
5	Chapter_Genderize_api.Rmd	0	NA
6	Chapter_Github_api.Rmd	0	NA
7	Chapter_Google_news_api.Rmd	0	NA
8	Chapter_Google_nlp_api.Rmd	1	lexical error: invalid char in json text. [...]
9	Chapter_Google_places_api.Rmd	0	NA
10	Chapter_Google_speech.Rmd	1	lexical error: invalid char in json text. [...]
11	Chapter_Google_translation_api.Rmd	1	lexical error: invalid char in json text. [...]
12	Chapter_Googletrends_api.Rmd	0	NA
13	Chapter_Instagram_basic_display_api.Rmd	0	NA
14	Chapter_Instagram_graph_api.Rmd	1	URL using bad/illegal format or missing URL
15	Chapter_Internet_archive.Rmd	0	NA
16	Chapter_Introduction.Rmd	0	NA
17	Chapter_Mediacloud_api.Rmd	0	NA
18	Chapter_Twitter_api.Rmd	0	NA
19	Chapter_Wiki_api.Rmd	0	NA
20	Chapter_Youtube_api.Rmd	0	NA
21	index.Rmd	0	NA

Figure: Output of error_control.R

Part 2: Introduction of individual chapters and their authors

Twitter, Media Cloud, Best practices

17.4.1.3 Pull word matrices

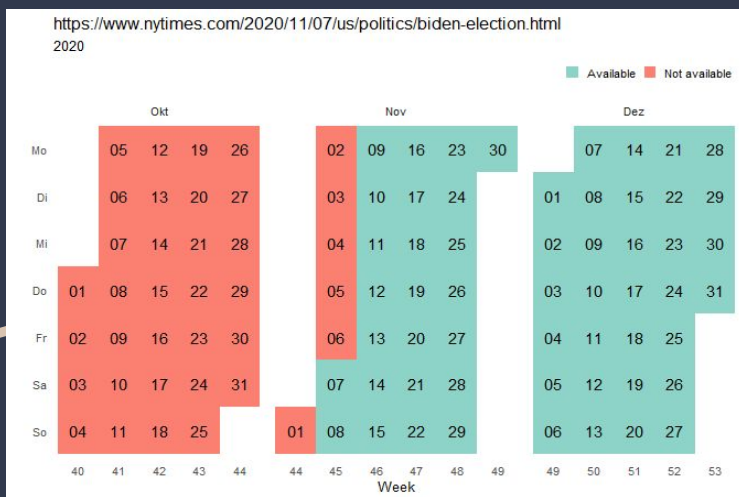
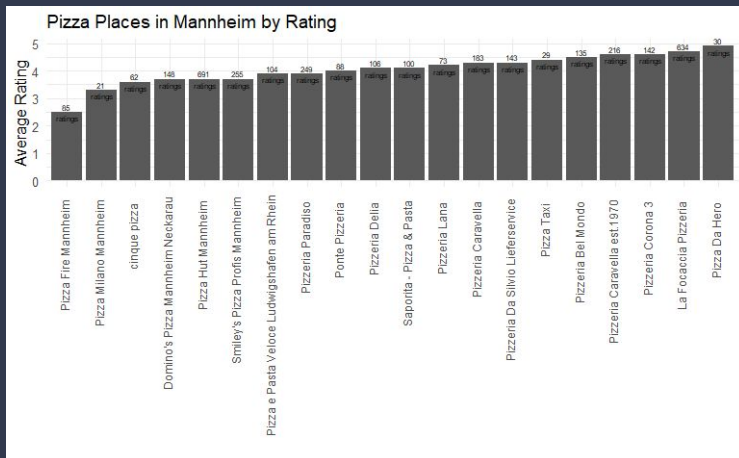
With the list of `stories_id`, we can then use the function `get_word_matrices()` to obtain word matrices. ¹¹


```
unima_mat <- get_word_matrices(stories_id = unima_articles$stories_id, n = 100)
```

```
## # A tibble: 30,353 × 4
##   stories_id word_counts word_stem      full_word
##   <chr>         <int> <chr>      <chr>
## 1 1815573995         1 deutschland deutschland
## 2 1815573995         1 befragt      befragt
## 3 1815573995         1 irgendeinem irgendeinem
## 4 1815573995         1 vorgelegten vorgelegten
## 5 1815573995         1 solo-selbstständig solo-selbstständige
## 6 1815573995         1 wert          wert
## 7 1815573995         1 zuefolge      zuefolge
## 8 1815573995         1 dpa-afx       dpa-afx
## 9 1815573995         1 universität  universität
## 10 1815573995         1 gewinnsitu    gewinnsituation
## # ... with 30,343 more rows
```

- Twitter Academic API v2, R package: **academictwitterR** (Barrie & Ho, 2021)
 - “There are so many (if not too many) social science research examples using this API.”
- Mediacloud - pulling DTMs of news articles for free, R package: **mediacloud** (Unkel, 2022)
- Best practices
 - Do Your Homework
 - Don't Hardcode Authentication Information into your R Code
 - Memoise your API Calls

Google Places API, Internet Archive API Lukas Isermann



- Part of the **Google Cloud project** and integrated in the wider universe of Google APIs
- Provided Services:
 - Place Search, Place Details, Place Photos, etc.
- integration in R via the **googleway** R-Package
- For example used to research changes in mobility during Covid-19 (Konrad, 2020)
- Gives access to the **Internet Archive**
- Provided Services:
 - Timestamps and Urls of mementos of any Url saved in the Internet Archive
- **archiveRetriever** R-Package (Gavras/Isermann, 2022) 
 - provides pipeline that facilitates scraping from the Internet Archive
 - easy overview over the availability of mementos for any Url
 - access to memento Urls and memento Urls of any subpages of a given memento
 - easy scraping of content from any memento Url
- For example used to access online newspaper articles (Gavras, 2022)

Marie-Lou Sohnus



rank	trackname	artist	track.popularity	danceability	valence
1	As It Was	Harry Styles	80	0.520	0.6620
2	Late Night Talking	Harry Styles	84	0.714	0.9010
3	Ojitos Lindos	Bad Bunny	92	0.647	0.2680
4	Me Porto Bonito	Bad Bunny	92	0.911	0.4250
5	Matilda	Harry Styles	82	0.507	0.3860

Figure 1. Simple track data

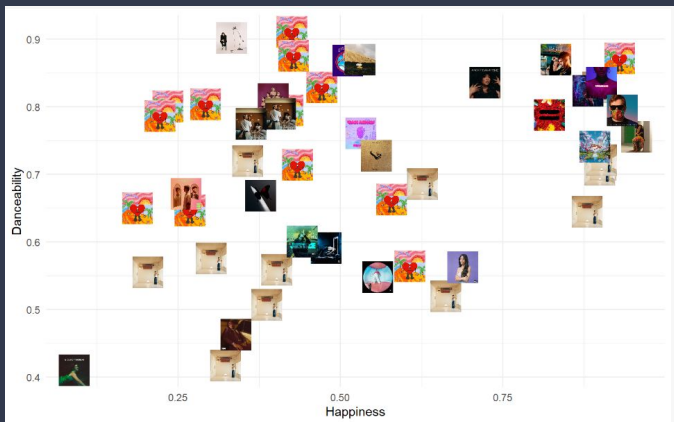


Figure 2. Visualisation of Top 50 by Happiness (x-axis) and Danceability (y-axis)

- The Spotify Web API allows you to easily pull data from the platform
- Data access requires (1) setting up a **user AND developer account** with Spotify and then (2) using a **simple API call or the spotifyr package**.
- Possible requests include
 - **General data** on
 - track audio features (e.g., danceability, score, or pace)
 - popularity metrics of single tracks and albums
 - playlist information
 - **Individual user data** (including your own) on listening behavior. This requires users to have used their Spotify account for a while to have generated any data of interest.

Reddit API

- The Reddit API is accessible freely to any non-commercial user.
 - Primarily used to get text data, but you can also get metadata for posts
- Data collection can be done with packages such as *httr* or with dedicated packages such as *RedditExtractoR*
 - *httr* is more versatile but requires a lot more work
 - *RedditExtractoR* gets you data easily but is limited in options of what data it can get

Name	Type	Value
• response	list [10] (S3: response)	List of length 10
url	character [1]	'https://www.reddit.com/r/cats/json'
status_code	integer [1]	200
• headers	list [27] (S3: insensitive, list)	List of length 27
• all_headers	list [1]	List of length 1
• cookies	list [4 x 7] (S3: data.frame)	A data.frame with 4 rows and 7 columns
content	raw [274226]	7b 22 6b 69 6e 64 ...
date	double (S3: POSIXct, POSIXt)	2022-09-07 07:36:30
• times	double [6]	0.0000 0.0138 0.0308 0.0745 0.7780 0.7880
• request	list [7] (S3: request)	List of length 7
• handle	externalptr (S3: curl_handle)	<pointer: 0x000002da2d6ffea0>

Figure 1. Raw output of the *httr* request

Domantas Undzėnas

Information	
post title	"Sister took my cat because of financial things. Covid happened. First time seeing my old bud in quite a few years."
upvotes	21346
upvote_ratio	0.97

Figure 2. Simple post data (Undzėnas, 2022)

title	comments
"5 years later, on St Patrick's Day, I return to tell Americans that they aren't Irish"	8885
"Kinkshaming should be encouraged"	3029
"It is impossible to be super fit and work a 9-6 job."	2758
"Can you explain this gap in your resume? Is totally rude and completely inappropriate to ask"	2243
"Using someone elses toothbrush is not a big deal"	2113
"Drake is an absolute shit rapper with shit lyrics and even shittier ideals and anyone who thinks otherwise has the brain/creativity of a lamp."	2075
"Girl scouts are a creepy scam"	2062
"Wearing dress clothes for work does nothing and just makes people uncomfortable all day"	1713
"Taco Bell doesn't give you the shits. You just never eat fiber."	1660
"College is a waste of time and money for most people"	1496

Figure 3. Top 10 posts from the *unpopularopinion* subreddit (Undzėnas, 2022)

APIs: CrowdTangle, Google NLP, Google Speech-to-Text, Google Translation



CrowdTangle API

- public insights tool to track social media content
- since 2016: run by Facebook (Meta)
- *platforms*: Facebook, Instagram, Reddit
- *posts*: public pages, groups or verified public person

Data Restriction

- basic functionality for everyone
- full access only to verified media publishers, content creators as well as academics

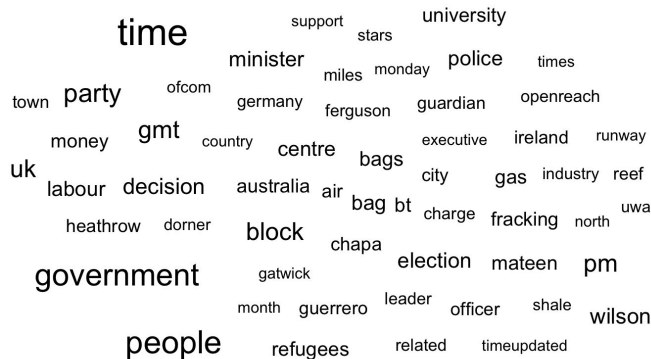
Data Access

- simple API calls available for everyone
- full access only to verified media publishers, content creators as well as academics
- simple API calls or *RCrowdTangle* package which directly produces a data frame

Google Natural Language API



- provides pre-trained models for natural language processing
- user provides (un-)labeled data
- examples of services: sentiment analysis, named entity recognition, syntax analysis (identify nouns, verbs, etc.)



Google Speech-to-Text API

- convert audio files to text using deep learning (> 100 languages supported)

Google Translation API

- translate text files into other languages (> 100 languages supported)

Part 3: Discussion – The Use of APIs for Social Science Research: Opportunities and Limitations

What are use cases and examples of unique data that social scientists can retrieve from APIs?

7,070

Views

111

CrossRef
citations to date

8

Altmetric

ARTICLES


Using APIs for Data Collection on Social Media

Stine Lomborg  & Anja Bechmann

Pages 256-265 | Received 12 Jul 2012, Accepted 03 Apr 2014, Published online: 08 Jul 2014

 Download citation

 <https://doi.org/10.1080/01972243.2014.915276>

 Check for updates

 Full Article

 Figures & data

 References

 Citations

 Metrics

 Reprints & Permissions

Get access



Abstract

This article discusses how social media research may benefit from social media companies making data available to researchers through their application programming interfaces (APIs). An API is a back-end interface through which third-party developers may connect new add-ons to an existing service. The API is also an interface for researchers to collect data off a given social media service for empirical analysis. Presenting a critical methodological discussion of the opportunities and challenges associated with quantitative and qualitative social media research based on APIs, this article highlights a number of general

Related research

People also
read

Recommended
articles

Cited by
111

[Social Media: Defining, Developing, and Divining >](#)

Caleb T. Carr et al.

Atlantic Journal of Communication

Published online: 6 Feb 2015

How does data collection via APIs, like in the aforementioned examples, compare to more traditional methods of data collection?

Other use cases where collecting data
via APIs is worthwhile?

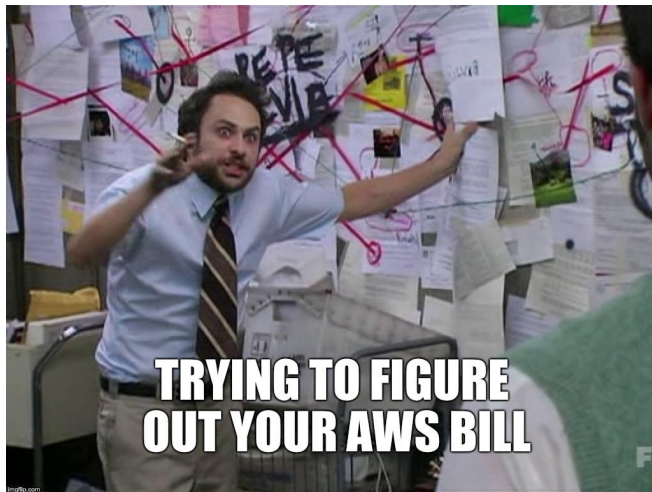
What are drawbacks of using APIs to collect or to manipulate data? Any bad experiences?

... especially in (social science) research?

Are there other dependencies or
“rules” we have to consider when
using these services?

Make sure you are familiar with the cost structure!

- What does the API service costs?
- Do you have enough funding?
- For Google APIs: [Google Cloud Research Credits](#)



What about data quality?

Can we say something about how accurate data from API calls is or about measurement error?

Accuracy (Konrad 2022)

WZB Data Science Blog

[HOME](#)[ABOUT](#)[LEGAL NOTICE / IMPRESSUM](#)

SOME THOUGHTS ABOUT THE USE OF CLOUD SERVICES AND WEB APIS IN SOCIAL SCIENCE RESEARCH

March 7, 2022 11:41 am , Markus Konrad

In the recent weeks I've collaborated on the online book *APIs for social scientists* and added two chapters: a [chapter about the genderize.io API](#) and a [chapter about the GitHub API](#). The book seeks to provide an overview about web or cloud services and their APIs that might be useful for social scientists and covers a wide range from [text translation](#) to accessing [social media APIs](#) complete with code examples in R. By harnessing the [GitHub workflow](#) model, the book itself is also a nice example of fruitful collaboration via work organization methods that were initially developed in the open source software community.



Recent posts

Some thoughts about the use of cloud services and web APIs in

Future directions for our project

- incentive structure for collaborating in our project?
- book or website?
- how can we ensure similar quality of the chapters over time?
- what APIs are readers interested in?

Your Questions?