

Data-driven: the humanities get digital

Katie Ireland Kuiper
5/12/2021

<https://orcid.org/0000-0002-4725-9525>

Analyze your data with R

For tips and how-to install R and R Studio, please check out this [document](#).

Today we will learn how to create a corpus, get data from Twitter (and Reddit) and perform frequency and other types of analyses. To prepare, please make sure you have the following R packages installed and loaded: tidyverse, tidytext, rtweet, quanteda, and ggplot2.

Find all information and handouts for our tutorial today [here](#).

#load the libraries

```
library(rtweet)
library(quanteda)
library(quanteda.textatata)
library(quanteda.textplots)
library(ggplot2)
library(tidytext)
library(dplyr)
library(tidyverse)
library(igraph)
library(ggraph)
library(RedditExtractor)
```

Getting Twitter data

Today we will create a corpus based on the keyword #janeusten. Rtweet is just one way to get data from Twitter. In order to get a large(r) amount of data, you will need to apply for a Rtweet developer account. Check out the application for becoming a developer and utilizing the Twitter API [here](#). Twitter is making it easier for researchers to access and work with data so definitely check it out! Twitter has recently updated rules and regulations for researchers interested in utilizing their data.

Using rtweet, users have two options for getting data. The first way utilizes the search_tweet function to get tweets, and a pop-up browser window will ask the user to authenticate the request. With this method, there are stricter limits to how many tweets and data the researcher can obtain. After authorizing the pop-up window, the authorization token will be stored in the user's .Renviron file.

The other method involves creating a Twitter developer account (see above), which is recommended for researchers and for obtaining more data.

```
#getting data using rtweet's search_tweets function
janeusten<- search_tweets("#janeusten", n = 35, include_rt = FALSE)
View(janeusten)

#Note: Rtweet includes multiple options for search functions including phrases

#search for a keyword
keyword <- search_tweets(q = "Emma")

# search for a phrase
phrase <- search_tweets(q = "Reader, I married him")

#search for multiple keywords
manykeywords <- search_tweets(q = "#janeusten AND #bronte")
```

Accessing metadata

```
#use this syntax to get the number of different locations (or other metadata options)
length(unique(janeusten$location))

#plot the different locations
janeusten %>%
  ggplot(aes(location)) +
  geom_bar() + coord_flip() +
  labs(x = "Count",
       y = "Location",
       title = "Locations in #janeusten Tweets")
```

Basic frequency analysis & implementing stopwords

```
#use the unnest function to get all the words separated out for frequency analysis
this format is function(nameofoutputcolumn, inputcolumn, optional tokenizer settings)
tidy_tweets <- janeusten %>% unnest_tokens(word, text, token = "tweets")
#check out the data!
View(tidy_tweets)
tidy_tweets$word

#subset the data by screen name, counts, and word tokens
groups <- tidy_tweets %>% group_by(screen_name, word) %>% summarize(count=n())
View(groups)

#get frequency counts
frequency <- tidy_tweets %>% count(word, sort = T)
frequency

# a few more cleaning options: lowercase all, implement stopwords
withstopwords <- tidy_tweets %>% filter(!word %in% stop_words$word, !word %in% str_remove_all(stop_words$word, ""))
str_detect(word, "[a-z]")

withstopwords <- withstopwords %>% count(word, sort = T)
withstopwords

#now plot it
withstopwords %>% filter(n > 1) %>% ggplot(aes(x = reorder(word, -n), y = n)) +
  geom_col() +
  labs(x = "word",
       y = "count",
       title = "Top words") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

#note: these settings can be adjusted in order to get more or less top words
withstopwords %>% filter(n > 4) %>% ggplot(aes(x = reorder(word, -n), y = n)) +
  geom_col() +
  labs(x = "word",
       y = "count",
       title = "Top words") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Ngram analysis

```
#get bigrams
bigram_tweets <- janeusten %>% unnest_tokens(bigram, text, token = "ngrams", n = 2, collapse = F)
View(bigram_tweets)

#prep for visualization
bigram_tweets <- bigram_tweets %>% separate(bigram, c("word1", "word2"), sep = " ") %>% count(word1, word2, sort = T) %>%select(word1, word2, n)
View(sep_bigrams)

#visualize it
sep_bigram %>%
  filter(n > 1) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  # geom_edge_link(aes(edge_alpha = n, edge_width = n))
  # geom_edge_link(aes(edge_alpha = n, edge_width = n)) +
  geom_node_point(color = "darkslategray4", size = 3) +
  geom_node_text(aes(label = name, vjust = 1.8, size = 3) +
  labs(title = "Word Network of #janeusten",
        subtitle = "Optional subtitle here",
        x = "", y = "")
```

Using quanteda

```
janeocorpus <- quanteda::corpus(janeusten)
summary(janeocorpus)
View(janeocorpus) #each tweet is separated out as its own text
janeocols <- quanteda::collocations(janeocorpus)
#options for visualizing the data
#search for a particular token, phrase
#get top words
janeocorpus_tokens <- tokens(janeocorpus)
kwic_jane <- kwic(janeocorpus_tokens, pattern = "jane")
View(kwic_jane) #this will open in viewer window

#Note: the pattern can be adjusted to include different options
#multiple keywords can be searched for, as below:
kwic_multiple <- quanteda::kwic(janeocorpus_tokens, pattern = c("jane", "the"))
kwic_multiple

#use window argument to adjust number of words on either side
kwic3 <- kwic(janeocorpus_tokens, pattern = "life", window = 4)

#use pattern = phrase("insert phrase") to look for different
#phrases, or add valuetype = "regex" to search for additional variations of a token
kwic(janeocorpus_tokens, pattern = "jane", valuetype = "regex")

#create an xray plot visualizing the distribution of a token
kwic(tokens(janeocorpus_tokens), pattern = "jane") %>%
  textplot_xray()
```

DFM with quanteda

```
#To create a document feature matrix, use the quanteda dfm function.
#dfm's are useful and help with a variety of options for different analyses.

#make a dfm
janedfm <- dfm(janeocorpus)

# make a dfm, removing stopwords and (optionally, applying stemming by adding stem = T
DFM <- dfm(janeocorpus, remove = stopwords("english"), remove_punct = TRUE)
DFM[, 1:5]
#get the top features
topfeatures(DFM, 20)

#visualize the frequencies and distribution of words using the word cloud
#create a dfm object to prepare for a wordcloud visualization
dfm_austen <- corpus_subset(janeocorpus) %>%
  dfm(remove = stopwords('english'), remove_punct = TRUE) %>%
  dfm_trim(min_termfreq = 3, verbose = FALSE) #adjust these settings to your needs

#word cloud time
set.seed(100)
textplot_wordcloud(dfm_austen)
#adjust settings, add in specific colors
textplot_wordcloud(dfm_austen, min_pcount = 25,
  color = c('red', 'pink', 'green', 'purple', 'orange', 'blue'))

#topic modeling with quanteda
#adjust the dfm to prep it for topic modeling
dfm_adjusted <- dfm_trim(DFM, min_termfreq = 4, max_docfreq = 10)
dfm_adjusted

# topic models package (Grün et al. 2021) uses latent dirichlet allocation for topic modeling; it also supports
different methods for model fitting and sampling, including Gibbs sampling

library(topicmodels)
LDA_20 <- convert(dfm_adjusted, to = "topicmodels") %>%
  LDA(k = 20) #k is the number of topics
View(LDA_20)
# get top five terms per topic
get_terms(LDA_20, 5)

#note: other packages support topic modeling in R, using DfMs. One such package is the stm package, which also takes into account document and corpus metadata.
```

Your turn! Use the RedditExtractor to get data from Reddit. Feel free to adjust the settings and search terms below according to your research interests. Note that this package has limits due to the Reddit API, including only 500 maximum comments per thread.

```
janeoreddit <- get_reddit(search_terms = "Jane Austen", regex_filter = "", subreddit = "janeusten",
  cn_threshold = 1, page_threshold = 1, sort_by = "comments",
  wait_time = 2)
#turn reddit posts into quanteda object
View(janeoreddit)

#excluding the post text bc every comment has post attached
janerreddit_cleaned <- janeoreddit %>% select(-post_text)

#creating the quanteda corpus object
redditcorpus <- quanteda::corpus(janerreddit_cleaned, text_field="comment")
summary(redditcorpus, 5)

#Now try the functions we used earlier on the Twitter corpus on your Jane Austen Reddit corpus!
```

Additional Resources

Data:

[Linguistic Data Consortium](#)

[Kaggle.com](#): many pages of social media datasets, including tweets, and others: example: disaster tweets dataset, Instagram data, emojis, reddit, and many many others.

[Stanford SNAP](#): large network dataset collection, including data from amazon, social media, Wikipedia and others

[Network Repository](#): combines social networks, biological, graph data and tools for analyzing and comparing available datasets

Web-Based Resources

[iScience Maps](#): web-based option for getting Twitter data, with options for sorting and analyzing the data

[Naoyun](#): software for connecting Twitter data with Gephi, with options for visualizing "live Twitter activity"

[Netlytic](#): uses APIs to collect public data from Twitter, YouTube, and RSS feeds. Includes free and paid user options, with network and text analytics

[Socioviz](#): get and analyze Twitter data in this web-based environment

[The Chorus Project](#): free web-based option for analyzing and obtaining Twitter data; based out of the UK

[Webometric Analyst](#): free Windows-based program for gathering data, including Social out, the Statistical Cybermetrics Research

[Digital Footprints](#): obtain and analyze Facebook data; web-based service available for researchers, based out of Aarhus University

[InfoExtractor](#): no longer maintained, but offers options for getting data from different URLs

[Snooportal](#): free for researchers; focus on obtaining Instagram data

R packages

[streamR](#): Access to Twitter Streaming API via R

[twittr](#): also useful for getting twitter data in R

[rfacebook](#): Rfacebook: Access to Facebook API via R

[instaR](#): access Instagram data via the Instagram API; an approved developer account is required

[RedditExtractor](#): utilizes Reddit API to obtain posts, comments, and subreddit information

[rtweet](#): useful package for getting Twitter data, with options for accessing followers, retweets, geolocation, and additional metadata.

[xml2](#) and [rvest](#) work well together for harvesting web data

[Rcurl](#) & [RSelenium](#)

Python Libraries

[spaCy](#): pos tagging, tokenization, dependency parsing, etc. Check out this [tutorial](#) for more about NLP with spaCy

[CoreNLP](#): lemmatization, pos tagging, tokenization, named entity recognition

[NLTK](#): Natural Language Toolkit; contains over 50 corpora, includes options for tokenization, tagging, parsing, document classification

[Gensim](#): useful for various types of topic modeling

[PyNLP](#): open-source NLP library; great for of tasks ranging from building simplistic models and extraction of n-grams and frequency lists, with support for complex data types and algorithms

[Pattern](#): useful for web-crawling (webscraping) for creating your own corpora; includes options for tokenizing, pos tagging, etc

[Polylot](#): NLP pipeline for multilingual applications, includes options for preprocessing, analysis of sentiment, morphological features, and more.

[TextBlob](#): includes options for tokenization, pos-tagging, noun phrase extraction, classification, translation and sentiment analysis

Thanks for listening!

Feel free to reach out with questions or comments:

Emily McGinn
mcginn@uga.edu

Katie Ireland Kuiper
katherine.kuiper25@uga.edu

All materials and resources from today's seminar are available [here](#) courtesy of The Digital Humanities Lab, University of Georgia.

Twitter:

- @DigiLab_UGA
- @EmMcGinn
- @Kannireland

Works Cited

Al-Rifou, Rami. 2015. [Polyglot](#).

Barbera, Pablo. 2018. [Package 'streamR'](#).

Barbera, Pablo, Michael Piccirilli, Andrew Geisler, and Wouter van Atteveldt. 2017. [Rfacebook: Access to Facebook API via R](#).

Barbera, Pablo, Tiago Dantas, Jonne Guyt. 2016. [Package 'instaR'](#).

Beckman, Matthew, Stéphane Guerrier, Justin Lee, Paulo Molinari, Samuel Orso & legor Rudnytskyi. 2020. [An Introduction to Statistical Programming with R](#).

Benoit, Kenneth, Kohel Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, Akitaka Matsuo, William Lowe. (2018). [quanteda: An R package for the quantitative analysis of textual data](#). Journal of Open Source Software, 3(30), 774. doi: 10.21105/joss.00774

Bird, Steven, Ewan Klein, and Edward Loper. 2019. [Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit](#).

Brown, Simon. 2016. [Tips for Computational Text Analysis](#).

Bussiere, Kirsten. 2018. [Digital Humanities - A Primer](#).

Cribbin, Timothy, et al. 2021. [The Chorus Project: making sense from Twitter/http://chorusanalytics.co.uk/about-us/](#)

Csardi G, Nepusz T (2006). "The [igraph software package](#) for complex network research." InterJournal, Complex Systems, 1695.

De Smedt, Tom. 2018. [Pattern 3.6](#)

Digital Footprints Research Group. 2021. [Digital Footprints](#). <https://digitalfootprints.dk/about> Aarhus University.

Evert, Stefan. 2007. [Corpora and collocations](#).

Feinerer, Ingo. 2020. [Introduction to the tm Package: Text Mining in R](#).

Gentry, Jeff. 2015. [twittrR: R Based Twitter Client](#).

Grün, Bettina & Kurt Hornik. [topicmodels: An R Package for Fitting Topic Models](#).

Gruzd, Anatolij, & Philip Mai. [Netlytic: Making sense of public discourse online](#).

Han, Na-Rae. [Python 3 tutorials](#).

Harrison, John & Ju Yeong Kim, 2020. [RSelenium](#).

Honnibal, Matthew and Montani, Ines and Van Landeghem, Sofie and Boyd, Adriane. 2020. [spaCy: Industrial-strength Natural Language Processing in Python](#).

Kearney, Matthew. 2018. [R: Collecting and Analyzing Twitter Data: featuring \(rtweet\)](#) NICAR 2018.

Kearney, Matthew, Andrew Heiss, and Francois Briatte. 2020. [Package 'rtweet'](#)

Kross, Sean et al. 2020. [swirl: Learn R, in R](#).

Kuiper, Katie Ireland. 2021. [Text Analysis Glossary](#) DigiLab.

Leskovec, Jure, and Krevl Andrej. 2014. [SNAP Datasets: Stanford Large Network Dataset Collection](#). <http://snap.stanford.edu/data>

Loria, Steven. 2020. [TextBlob: Simplified Text Processing](#).

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. [The Stanford CoreNLP Natural Language Processing Toolkit](#) In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60.

Pederson, Thomas. 2021. [ggraph: an implementation of grammar of graphics for graphs and networks](#)

Radim, R. & P. Sojka. 2010. [Software Framework for Topic Modelling with Large Corpora](#). Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. 45-50.

Reips, U.-D., & Garaziar, P. (2011). Mining Twitter: Microblogging as a source for psychological wisdom of the crowds. [Behavior Research Methods](#), 43, 635-642. doi <http://dx.doi.org/10.3758/s13428-011-0116-6>

Rivera, Ian. 2019. [package RedditExtractor](#).

Roberts et al. 2019. [stm package for structural topic modeling](#). <https://www.jstatsoft.org/article/view/v09i02>

Rossi, Ryan, & Nesreen Ahmed. 2021. [Network Repository: An Interactive Scientific Network Data Repository](#).<http://networkrepository.com/>

Rüdiger, Sophia, and Daria Dayter. 2020. [Corpus Approaches to Social Media](#). In [Studies in Corpus Linguistics](#).

Schradang, N. 2015. [Intro to NLP with spaCy](#).

Shah, Chirag. [infoextractor](#). [infoextractor.org](#)

Silge, Julia, and David Robinson. 2017. [Text Mining with R: A Tidy Approach](#).

Theilwall, M., & Sud, P. 2012. [Webometric research with the Bing Search API 2.0](#). Journal of Informetrics, 6(1), 44-52.

Totet, Matthieu. 2021. [Naoyun- Visualize Live Twitter Activity](#).

van Gompel, Maarten. 2016. [PyNLP](#).

Wasser, Leah, & Carson Farmer. 2020. [Twitter Data in R Using Rtweet: Analyze and Download Twitter Data](#). Earth Data Science.

Watanabe, Kohel. 2021. [Example: social media analysis](#).

Wickham et al. 2019. [Welcome to the tidyverse](#). Journal of Open Source Software, 4(43), 1686. <https://doi.org/10.21105/joss.01686>

Wickham et al. 2020. [xml2: Parse XML](#).

Zonin, A. 2015. [SocioViz: A Free Social Network Analysis Tool for Twitter \[Software\]](#). Available from <https://socioviz.net/>