Topic Modeling in R

Wouter van Atteveldt

CCS Hannover, Feb 2018
What, why?

- What: Topic modeling: unsupervised term/document clustering using R as a platform for text analysis
- Why: Exploration, cheap coding, semi-automatic coding
When?

Tuesday:
- AM: **Intro**, Text analysis with R
- PM: Topic Modeling: application and validation

Wednesday:
- AM: Technical Details
- PM: Structural Topic Modeling

Thursday:
- AM: Linguistic processing; visualization
How?

Each session:
- 60-90 minutes interactive lecture
- 60-90 minutes practice on your own data

Please:
- Interrupt me
- Work on your own code
- Ask me stupid, easy, difficult, and/or seemingly unrelated questions!
Who?

- Wouter van Atteveldt
- VU Amsterdam
- M.Sc. in Comp.Ling.; PhD in A.I. & CS
- Computation Methods (Interest Group)
Who? (2)

- Kasper Welbers, Post-doc VU Amsterdam
- First author corpustools, tokenbrowser, RNewsFlow
Who? (3)

• And who are you? :-)

• (e.g. name, affiliation, interest in topic modeling, current research, experience with R/Python/...)
Automatic Text Analysis

- Huge amounts of textual data exist
  - Twitter, newspaper archives, reviews, books, ...
- Impossible to read all texts
  - Methods to **explore** meaning
  - Methods to **analyse** relations
  - Visualization to **convey** interpretation
- Textual data points have **meaning**
  - (actually, they’re symbols to which we attach meaning)
Why do we analyse/visualize texts?
Why do we analyse/visualize texts?

(a) Israel subject, U.S. media

(b) Israel subject, Chinese media
Text Analysis & Digital Social Science

- Flood of digital information (online/archived)
- About (social) behaviour
- Much of it textual
- 'Measure' content of messages
- Explain/understand/predict social behaviour
Applications of Text Analysis

Bar charts showing the comparison of control and experimental groups for positive words and negativity reduced, and positivity reduced.

- Positive Words (per cent)
  - Control: 5.2
  - Experimental: 5.3

- Negativity Reduced
  - Control: 5.4
  - Experimental: 5.1

- Positivity Reduced
  - Control: 1.60
  - Experimental: 1.50
Introduction

Applications of Text Analysis

Topic Modeling in R

Wouter van Atteveldt
Applications of Text Analysis
Applications of Text Analysis

Topic Modeling in R

Wouter van Atteveldt
Applications of Text Analysis
Hospital officials in Gaza said that 390 people were killed by Israeli fighter planes.
Applications of Text Analysis

- K=10: Accidents/danger
- K=25: Chernobyl & Fukushima
- K=25: 3 mile island
- K=25: Power plant construction (Shoreham)
Applications of Text Analysis

Introduction

Text Analysis in R

R Recap

Top Topic Modeling in R

Wouter van Atteveldt
Applications of Text Analysis

![Graphs showing correlation between text analysis for different periods]

**Figure 3.** Correlation between Framing of Immigration in each Party’s Press Releases with the Manifesto of All Parties. Only Significant Correlations Are Shown.
Applications of Text Analysis

Fig. 1. Scaled Positions of Party Manifestoes.
Applications of Text Analysis

- ovation
- Johnson
- independence
- control
- immigration
- back
- campaign
- tonight
- boris
- project
- done
- plan
- fear
- thank
- save
- davidson
- truth
- won
- will
- remain
- great
- right
- now
- voted
- thanks
- referendum
- result
- see
- one
- good
- sure
- make
- take
- just
- says
- we
- know
- people
- political
- decide
- back
Recap: What is R?

Motivational example

(R in 10 seconds)
What is R?

- Open source, multi-platform
- Text/Console
- Full programming language
  - Functional, Interpreted (=slow)
  - Packages use C, Fortran (=*fast*)
- Community driven:
  - User packages are "1st class citizens"
- Decent tooling support (esp. RStudio)
Cathedral vs. Bazaar
Why use R for text analysis?

- Compared to proprietary / single task tools
  - Learn one language, gain many options
  - Easier to combine multiple methods
  - Can import/export as needed (e.g. gephi)

- Compared to python
  - Both would be fine, both have strong use in data science
  - R more geared towards stats
  - Python more geared towards general programming and web development

- Can do scraping, cleaning, analysis, visualization, ...
Everything is a variable

- Variable: data with a name
- Every variable has a type
  - Numbers, text, data frame
- Name is for your convenience
- Combine, load, save, copy, delete variables
Recap: What is R?

Reading/writing data

- R: save, load(file)
- CSV: read.csv, read.csv2, write.csv
- SPSS: + read.spss(to.data.frame=T)
  - library(foreign)
Inspecting data

- Inspecting a data frame
  - names
  - head and tail
  - RStudio: View

- Inspecting a column (vector)
  - summary
  - length
  - mean, sum, etc
Selecting data

- Selecting rows:
  - `subset(data, column==value)`
  - `subset(data, column>=value, c("col1", "col2"))`
- Selecting a column:
  - `data$column`
Transforming data

- base R: assign to columns, select with [] and subset
- reshape2 functions (melt, acast, dcast)
- dplyr functions (filter, arrange, select, mutate)
Steps in Text Analysis

*Figure 1.* Order of text analysis operations for data preparation and analysis.

Welbers, van Atteveldt, Benoit (2017), *Text Analysis in R*, CMM
Text Analysis in R

So how do we do it?
KEEP CALM
THERE'S A PACKAGE FOR THAT
KeepCalmAndPosters.com
Text Analysis in R

- readtext: text input
- stringi: basic string (character) manipulation
- quanteda: bag-of-words text analysis, frequency analyses
- corpustools: token-based text analysis
- wordcloud
- topicmodels, stm (structural topic models)
- spacyr, coreNLP, nlpiper
- (tm, tidytext, RTextTools, etc., etc.)
## Step 1

Reading texts into R
Getting texts

Goal: data frame with text column (and metadata)

- Built-in data
  - `data("AssociatedPress", package="topicmodels")`
  - gutenburgr
  - Example data in other packages (quanteda, corpustools)
- From csv: `read.csv`
- Other files: package `readtext`
- Scraping:
  - `Rvest`, `RFacebook`, `twitteR`, ...
(1) Reading Texts

## Built-in text

```r
data("AssociatedPress", package="topicmodels")

library(gutenbergr)
gutenberg_works(author == "Austen, Jane")
d <- gutenberg_download(768)
```
Readtext package

library(readtext)
d = readtext(url, text_field = "texts")
Scraping

• `rvest`: scrape HTML from R (using CSS)
• `witteR`, `RFacebook`, `nytimes`: scrape APIs (needs API key / App)

See handouts on course page
Twitter from R

• Need to create an 'app' in twitter and get tokens
  • See e.g. https://www.r-bloggers.com/
    setting-up-the-twitter-r-package-for-text-analytics

• Conduct queries using Search API
  • Can only access recent tweets
    how-to-build-a-query

• See handout "Using API’s from R"
Twitter from R

devtools::install_github("geoffjentry/twitteR")
library(twitteR)
setup_twitter_oauth(...)
tweets = searchTwitter("#Trump2016",
                   resultType="recent", n = 10)
tweets = plyr::ldply(tweets, as.data.frame)

- https://www.r-bloggers.com/
  setting-up-the-twitter-r-package-for-text-analytics
Facebook from R

- Get token at
  developers.facebook.com/tools/explorer

```r
posts = getPage(page="nos", token=token, n=25)
post = getPost(post=post_id, token=token, n=100)
comments = post$comments
```

thinktostart.com/analyzing-facebook-with-r/
developers.facebook.com/docs/graph-api
AmCAT from R

- Install AmCAT-r:
  - `install.packages("devtools")`
  - `devtools::install_github("amcat/amcat-r")`

- Connecting to AmCAT:
  - `library(amcat)`
  - `amcat.save.password(...)`
  - `conn = amcat.connect(...)`

- Querying AmCAT:
  - `amcat.aggregate(conn, sets=.., ...)`
  - `amcat.hits(conn, sets=.., ...)`
  - `amcat.hits(conn, project=.., sets=.., col=c("headline"))`
  - `amcat.getarticlemeta(conn, set=.., ...)"
Step 2

Analysing text
The document term matrix

- Main primitive for frequency based text analysis
- Huge (sparse) matrix with:
  - **Rows** are the documents
  - **Columns** are the (unique) words
  - **Cells** contain the frequency of that word in that document
Creating a DTM

- Tokenize text
- Clean text
  - Remove stop words, noise
  - Remove high/low frequency words
  - Lemmatize/stem
Quanteda

```
library(quanteda)
tokens = tokens(sotu_texts$text, remove_punct = T)
tokens = tokens_tolower(tokens)
tokens = tokens_remove(tokens, c(stopwords("english")))
tokens = tokens_wordstem(tokens, "english")
dfm = dfm(tokens)
dfm = dfm_trim(dfm, min_docfreq = 5)
textplot_wordcloud(dfm, max.words=100)
```

(or simply dfm(text, options))
Word clouds and contrast plots

textplot_wordcloud(dfm, max.words=100)
keyness = textstat_keyness(dfm, target)
textplot_keyness(keyness)
Hands-on

Go get some data and make a word cloud :)
See "Intro to text analysis" handout
Bored? Have a look at the optional handouts:
  • social media scraping
  • using corpustools