# Analyzing texts with R

Adam Obeng

Sociology PhD at Columbia, computational social sci

Twitter!

Tells a story of how got a video game from a friend and then learned to program through messing with its code. Jedi Knight Dark Forces II.

Ken Benoit’s class “The Quantitative Analysis of Textual Data” (NYU Fall 2014) is free online! Put it on my list.

Quanteda and readtext – he’s one of the creators!

London School of Ec team worked on it. Primary is Ken Benoit.

Quantitative text analysis is more specific than the name implies.

Primarily used in poli sci.

Set of methods

QTA is a/b text as data, but not in the same way linguistics and comp sci are. In QTA, interest is the relationship b/w text and underlying behavior. We use the text to study underlying social behavior.

QTA assumptions

Texts reflect characteristics (of the authors)

Eg political views, or amt of scientific training, of the authors

Texts represented by features

Look at features like term counts rather than text as a whole

Analysis estimates characteristics

Want to use analysis of texts to get at characteristics of the people

QTA process: documents -> document-feature matrix -> analysis

In a dfm, rows are documents, columns are features

e.g., counts of particular words in each doc

A document-term matrix is a type of dfm, where the columns are *only* terms, not other features like ngrams, skipgrams

Methods include descriptive stats on words; extraction of topics; sentiment analysis; vocab analysis; classifying docs; scaling docs

ML is used in many of these

# Outline:

Loading texts (descriptive stats)

Extracting features

A particular kind of analysis: supervised scaling

Step 1: loading texts

library(quanteda)

library(readtext)

#readtext has a bunch of functionality that was in quanteda, but has been separated. Designed to make a dataframe containing texts. Single-fct package.

#We’re gonna look at presidential inauguration addresses from archive.org

inaug\_texts <- readtext(‘https://archive.org/download/Inaugural-Address-Corpus-1789-2009/inaugural.zip’)

str(inaug\_texts) #the structure of it gives you a nice idea of how it looks

Turn it into a corpus

inaug\_corpus <- corpus(inaug\_texts)

#texts(inaug\_corpus) would let you print the texts back out from the corpus; can see all the text files in the corpus, including the readme at the end

#”Docvars” – document variables – are metadata. Let’s make some based on filenames. We see the filenames have common structure – year address was made and president who made it. We’ll also include file extension in case we need it.

docvars(inaug\_corpus, c(‘year’, ‘president’, ‘extension’)) <- Stri\_split\_regex(names(texts(inaug\_corpus)), ‘[-.]’, simplify=T) #simple regex for where to split

docvars(inaug\_corpus) # to check

library(magrittr)

inaug\_corpus %>% corpus\_subset(Year ~=’README’) -> inaug\_corpus #remove readme from the corpus

docvars(inaug\_corpus) #no longer includes readme.

summary(inaug\_corpus) #gives us info on names of texts; number of docs (56); types (number of *distinct* words), tokens (number of words), sentences; year president and extension metadata we made

#Note that there are some fcts in the package for how to deal w/ punctuation, eg. If don’t want ngrams to span sentence boundaries

Keyword and context – semi-quantitative way to look at usage of words in texts. Gets context.

Let’s say we’re interested in the word “tax.”

kwic(inaug\_corpus, ‘tax’) #gives a list of all uses of the word “tax” in the docs, plus some words on either side.

#Good quick way to see what people are talking a/b when talk a/b smthng!

And can look at “taxes” “taxation” etc too with

kwic(inaug\_corpus, ‘tax\*’)

Can do regexes and such!

Lexical dispersion plot is a useful tool related to kwic – plot w/ occurrences of the word you’re looking for, and at what point in the document each was mentioned. E.g. see that in 1925 Coolidge speech there’s a cluster of mentions of taxes towards the end

plot(kwic(inaug\_corpus, ‘tax\*’))

textstat\_readbility() lets you do many diff readability measures

textstat\_readbility(inaug\_corpus, measure=c(‘Flesch.Kincaid’, ‘Scrabble’))

#Here we’re looking at reading grade level and at Scrabble score of words

inaug\_readbility[,’Flesch.Kincaid’] #another method, check the diffs

## Digression:

Risk of namespace collisions w/ other R packages. Approaches are inconsistent; came up w/ a consistent naming scheme w/in this package, but there isn’t really a general one. There’s a tension b/w usability and avoiding namespace collision.

Split out readtext from quanteda. Readtext is the only fct in titular package. Intended to give you the text from any file you point it at.

Specify what file you want to point it at; has various metadata parameters, encoding parameter

Can handle formats including pdf, json and Twitter API output, XML and html, zip tar and gz, remote files, glob paths, in addition to the obvious

Xpath is smthng handy – check what it is

readtext(‘path/to’whatever’) generally just works!

Given something like a path to a file, want to return all matching files

A path can resolve to multiple files, e.g. ‘/ath/to/\*.csv’ will match any CSV file on in that dir on any Unix system

‘/path/to/\\*.tsv’ can match diff numbers of files on Windows than on Unix

Recursive problem, and has a recursive solution!

If it’s a remote file, download it

If an archive, extract it, glob the contents

If a dir, glob the contents

Then call listMatchingFiles() on the result, and then repeat!

Termination condition: was it a glob last time? A glob can’t resolve to a glob.

Can’t tell just by looking at a path whether it’ll refer to more than one file, in general, if it’s a glob expression.

QTA STEP 2: Extracting features

text -> dfm

Feature creation (NLP)

Tokenizing

Removing stopwords

Stemming (removing word endings)

Skip/ ngrams

Dictionaries

Feature selection

Document frequency (e.g. words that appear in more than n docs)

Term frequency (e.g. words that appear more than n times)

Purposive selection (specific words you care about)

Deliberate disregard (words you want to get rid of)

And can do, say, words that only appear at least 3 times in at least 5 docs

inaug\_dfm <- dfm(inaug\_corpus)

head(inaug\_dfm)

inaug\_dfm <- dfm(inaug\_corpus, remove=stopwords(‘english’))

Can remove punctuation and so on

Can create dictionaries that map words to other words

Grouping’s a fun one. We could *group* features by president rather than by document, for example.

inaug\_dfm <- dfm(inaug\_corpus, groups=’president’)

Simple.

Clustering. Rather than look at whole DFM, let’s look at a subset after 1950

Inaug\_corpus %>% subset(year>1950) -> recent\_corpus

Recent\_dfm <- dfm(recent\_corpus, group=’president’)

Recent\_distance\_matrix <- dist(as.matrix(recent\_dfm))

Recent\_cluster <- hclust(recent\_dist\_matrix)

Recent\_cluster$labels <- docnames(recent\_dfm)

Plot(recent\_cluster)

Now we’ve got a dendrogram of which presidents use the most similar words!

QTA step 3: analysis

Supervised scaling is an analysis where goal is to differentiate document characteristics

A v common one is to classify where the doc or its authors) falls on the political spectrum

Like ML classification problem, but w/ continuous outcome

Get training (reference) texts

Generate word scores in training texts

Score test texts

Evaluate performance

Wordscores approach developed by Ken Benoit

Demonstration using UK party manifestos to capture when Labour become more centrist than lib dems

Library(quantedaData) #has some nice data sources

Data(ukManifestos)

Uk\_corpus <- corpus\_subset(ukManifestos, Year %in% c(1992, 1997) & Party %in% c(‘Con’, ‘Lab’, ‘LD’))

Docvars(uk\_corpus)

Uk\_dfm <- dfm(uk\_corpus, removePunct=TRUE)

Training\_scores <- c(17.21, 5.35, 8.21, rep(NA, 3)) #came from an expert survey

Ws\_fitted <- textmodel(uk\_dfm, taining\_scores, model=’wordscores’)

Ws\_fitted #word scores that correspond to features – tell you how words contribute to classification as left or right

Pred <- predict(ws\_fitted, newdata=uk\_dfm[4:6,], rescaling=’lbg’) #Rescaling b/c regression to the mean means a lot of words that shouldn’t do end up getting considered indicators of affiliation

Pred >- data.frame(pred@textscores)

Pred$gold <- c(17.18, 9.17, 5.)) #also from the literature

Mean(abs( #skipped thru quickly

## Digression: testing

Greg Wilson at Software Carpentry asks “Do you want your results to be correct or plausible?”

If you want your results to be correct, better test them!

V easy to get high accuracy w/ smthng that won’t work in the real world

Used Travis service to do continuous integration and test code for bugs. Worked v well.

Sometimes it’s R’s fault!

Fcts used to create temp files

Base::tempfile(): usually diff filenames w/in the same session – randomized

Base::tempdir(): always the same dir name win the same session

This resulted in problems when tried to create new temp dirs. And got ones that already had files in them

Readlines() can behave weirdly, fail to understand where a file ends, resolve when change the length of the file slightly by deleting or adding chars *anywhere* in the file. Why? Not yet known!

# Questions

QTA is poli sciterm. Does overlap w/ text mining, but there are some poli sci specific questions and methods it’s good for.

NLP is used more to mean processing a grammatical structure of words. E.g., stemming is an NLP issue; in QTA is a data cleaning step, but is research area in NLP (w/ outstanding problems)

Says text mining is largely a marketing term more than technical term

Sentiment analysis is hard b/c of contexts! Good approaches include expert labeling, and seeing how others *react* to it – do they react to it as if it’s an insult?

That seems vulnerable to a recursion, though. What’s the I-feel-insulted sentiment?

Why develop a new package rather than contribute to an existing one like tm?

Note quanteda and tm are easy to use in collaboration, can exchange a lot of objects back and forth

Some of it is personal preference, some things are simpler in one than the other

Quanteda is Swiss Army knife, lots of diff text models and readability measures; aims to have all the common measures used in an academic discipline. Tm is more “do one thing and do at well,” plus interoperate w/ other things.