Quantitative Content Analysis: Lecture 8

Matthias Haber

05 April 2017

Today's outline

- Constructing a document-feature matrix
- Preprocessing
- Regular Expressions

Basic Principles

- Corpus texts are text repositories.
  - Should not have their texts modified as part of preparation or analysis
  - Subsetting or redefining documents is allowable
- A corpus should be capable of holding additional objects that will be associated with the corpus, such as dictionaries, stopword, and phrase lists, etc.
- A document-feature matrix (dfm) is a sparse matrix that is always documents in rows by features in columns
- Encoding of texts should be done in the corpus, and recorded as meta-data in the corpus
  - This encoding should be `UTF-8` by default (problem for Windows machines)

Quanteda

quanteda is an R package for managing and analyzing text, created by Kenneth Benoit, Kohei Watanabe, Paul Nulty, Adam Obeng, Haiyan Wang, Ben Lauderdale, and Will Lowe. You can install **quanteda** from inside RStudio, from the Tools. . . Install Packages menu, or simply using

```
install.packages("quanteda")
```

You can also install the developers version directly from Github

```
# the devtools package is required
devtools::install_github("kbenoit/quanteda")
```

Note that on Windows platforms, it is also recommended that you install the RTools suite, and for OS X, that you install XCode from the App Store.

Explore quanteda

```
# Load the package
library(quanteda)
```

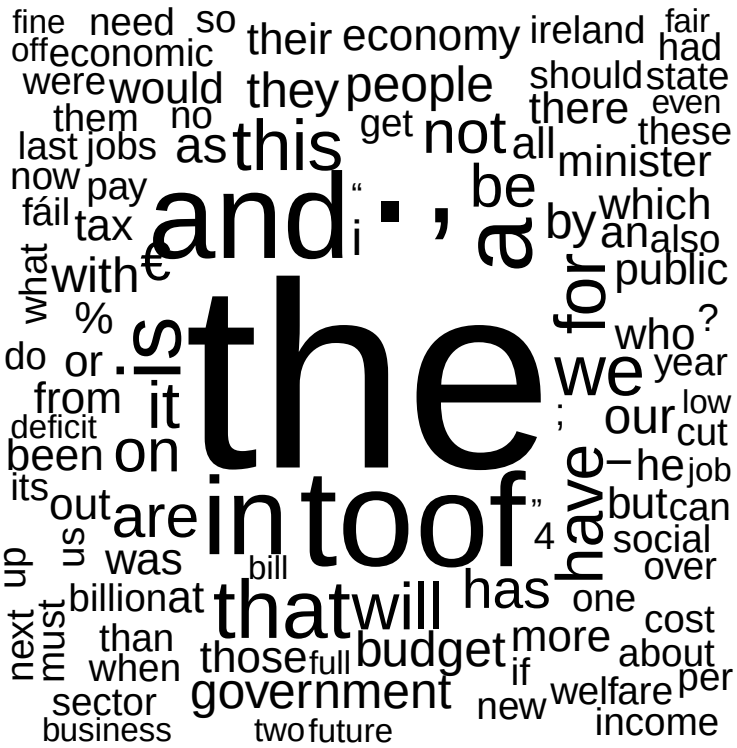
```
# Summarize some texts in the Irish 2010 budget speech corpus
summary(data_corpus_irishbudget2010)
```

```
# Create a document-feature matrix from this corpus
ibDfm <- quanteda::dfm(data_corpus_irishbudget2010,
                       verbose = F)
```

```
# Look at the top occuring features
quanteda::topfeatures(ibDfm)
##  the    .   to    ,   of  and   in    a   is that
## 3600 2371 1639 1548 1537 1360 1233 1013  868  804
```

Explore quanteda (II)

```
# Make a word cloud
quanteda::textplot_wordcloud(ibDfm, min.freq=25,
                             random.order=F)
```



Text analysis workflow

The goal is to simplify text and reduce dimensionality of the dfm created from it. In a nutshell, we want to filter relevant information and discard irrelevant information.

1. Creating the corpus
   - reading files
   - creating a corpus
   - addingdocument variables and metadata
2. Defining and delimiting documents
   - defining what are “documents” and what are “sentences”

Text analysis workflow (II)

3. Defining and delimiting textual features, using:
   - indentify instances of defined features ("tokens") and extract them as vectors
   - usually these will consist of terms, but may also consist of:
     - `ngrams` and `skipgrams`, sequences of adjacent or nearby tokens
     - multi-word expressions, through `phrasetotoken`
   - in this step we also apply rules that will keep or ignore elements, such as
     - punctuation
     - numbers, including or currency-prefixed digits
     - URLs
     - Twitter tags
     - inter-token separators

Text analysis workflow (III)

4. Further feature selection

- Once defined and extracted from the texts (the tokenization step), features may be:
  - removed or kept through use of predefined lists or patterns
  - collapsed by:
    - stemming
    - converting to lower case