quanteda initiative

quanteda

Quantitative Analysis of Textual Data

Stefan Müller (www.muellerstefan.net)

Presentation at Zurich R User Group, 14 October 2019

About me

- Stefan Müller
- PhD in Political Science
- Postdoc at the University of Zurich (since 01/2019)
- Assistant Professor at University College Dublin (from 01/2020)
- My research:
    1. Party competition and campaign strategies
    2. Elections and public opinion
    3. Quantitative text analysis
- Core contributor to the **quanteda** package
- Member of the Quanteda Initiative
- Contact:
    - https://muellerstefan.net
    - https://quanteda.io
    - @ste_mueller

Text is (almost) everywhere

- Open-ended survey questions
- Newspapers
- Videos (speech recognition)
- Online discussions
- Social media
- Party manifestos
- Political speech
- Legal texts and judicial decisions

quanteda: Quantitative Analysis of Textual Data

quanteda: Quantitative Analysis of Textual Data

History

- 7 years of development
- 30 releases, 8,500 commits

Core contributors

Kenneth Benoit, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2018. "quanteda: An R Package for the Quantitative Analysis of Textual Data." *Journal of Open Source Software* 3(30): 774.

Design of the package

- Consistent grammar
- Flexible for power users, simple for beginners
- Analytic transparency and reproducibility
- Compability with other packages
- Emphasize performance: use parallelization and sparse matrices
- Pipelined workflow using **magrittr**'s %>%
- Extensive documentation

Workflow, assumptions, and examples

Workflow, demystified

Raw texts → **Matrix representation** → **Analytics**

Fellow-Citizens of the Senate and of the House of Representatives: Among the vicissitudes incident to life..

Fellow citizens, I am again called upon by the voice of my country to execute the functions of its Chief Magistrate. When the occasion

When it was first perceived, in early times, that no middle course for America remained between unlimited submission to a foreign legislature

- tokenization
- feature selection

column 0: rownames	fellow-citizens	of	the	senate	and
1789-Washington	1	71	116	1	48
1793-Washington	0	11	13	0	2
1797-Adams	3	140	163	1	130
1801-Jefferson	2	104	130	0	81
1805-Jefferson	0	101	143	0	93
1809-Madison	1	69	104	0	43
1813-Madison	1	65	100	0	44
1817-Monroe	5	164	275	0	122
1821-Monroe	1	197	360	0	141
1825-Adams	0	245	304	0	116
1829-Jackson	0	71	92	0	49
1833-Jackson	0	76	101	0	53
1837-VanBuren	0	198	252	0	150
1841-Harrison	11	604	829	5	231
1845-Polk	1	298	397	0	189

Statistics:

- Term frequencies
- Keyness
- Readability
- Lexical diversity
- Similarity, distance

Models

- Supervised ML
- Unsupervised ML
- Scaling
- "Word embeddings"
- Topic models

Plots

Keyness, networks, scaling, word clouds, "x-ray"

Workflow: destroy language and turn it into data

```
library(quanteda)
corp <- corpus(c("A corpus is a set of documents.",
                 "This is the second document in the corpus."))
```

```
tokens(corp)
```

```
## tokens from 2 documents.
## text1 :
## [1] "A"         "corpus"    "is"        "a"         "set"       "of"
## [7] "documents" "."
##
## text2 :
## [1] "This"     "is"       "the"      "second"   "document" "in"
## [7] "the"      "corpus"   "."
```

```
dfm(corp)
```

```
## Document-feature matrix of: 2 documents, 12 features (37.5% sparse).
## 2 x 12 sparse Matrix of class "dfm"
##        features
## docs    a corpus is set of documents . this the second document in
##   text1 2      1  1   1  1         1 1    0   0      0        0  0
##   text2 0      1  1   0  0         0 1    1   2      1        1  1
```

Feature selection

```
# remove punctuation and stopwords and stem terms
toks <- tokens(corp, remove_punct = TRUE) %>%
  tokens_remove(stopwords("en")) %>%
  tokens_wordstem()
toks
```

```
## tokens from 2 documents.
## text1 :
## [1] "corpus"   "set"      "document"
##
## text2 :
## [1] "second"   "document" "corpus"
```

```
# create document-feature matrix
dfm(toks)
```

```
## Document-feature matrix of: 2 documents, 4 features (25.0% sparse).
## 2 x 4 sparse Matrix of class "dfm"
##        features
## docs    corpus set document second
##   text1      1   1        1      0
##   text2      1   0        1      1
```

Bag of words is a (convenient) lie

- Stemming and lemmatization are crude
- Words occur in *phrases* in most languages
  - Example: value added tax, United States of America
  - BUT: Oberweserdampfschifffahrtskapitän

Text analysis is fundamentally qualitative

Corpus of Irish budget speeches

```
summary(data_corpus_irishbudget2010, n = 6)
```

```
## Corpus consisting of 14 documents, showing 6 documents:
##
##                  Text Types Tokens Sentences year debate number   foren
##   Lenihan, Brian (FF)  1953   8641       374 2010 BUDGET     01   Brian
##  Bruton, Richard (FG)  1040   4446       217 2010 BUDGET     02 Richard
##    Burton, Joan (LAB)  1624   6393       307 2010 BUDGET     03    Joan
##   Morgan, Arthur (SF)  1595   7107       343 2010 BUDGET     04  Arthur
##     Cowen, Brian (FF)  1629   6599       250 2010 BUDGET     05   Brian
##      Kenny, Enda (FG)  1148   4232       153 2010 BUDGET     06    Enda
##     name party
##  Lenihan    FF
##   Bruton    FG
##   Burton   LAB
##   Morgan    SF
##    Cowen    FF
##    Kenny    FG
##
## Source: /Users/kbenoit/Dropbox (Personal)/GitHub/quanteda/* on x86_64 by kbenoit
## Created: Wed Jun 28 22:04:18 2017
## Notes:
```

Text analysis is fundamentally qualitative

```
kw <- kwic(data_corpus_irishbudget2010, pattern = "Christmas", window = 7)
nrow(kw)
```

```
## [1] 19
```

```
head(kw, 8)
```

```
##
##  [Bruton, Richard (FG), 699]              to survive and to see out this |
##    [Burton, Joan (LAB), 419]       ask listeners to suggest titles for a |
##    [Burton, Joan (LAB), 428]        single. Fianna Fáil's hit single for |
##   [Burton, Joan (LAB), 1039]        men and women will say goodbye after |
##   [Burton, Joan (LAB), 1701]     roaring trade in single golf clubs this |
##   [Burton, Joan (LAB), 1929] the Simon Community faking its message this |
##   [Burton, Joan (LAB), 3508]         shopping bags. In previous years at |
##   [Morgan, Arthur (SF), 374]                    the€ 204 per week or the |
##
##  Christmas | in the hope of something better in
##  Christmas | hit single. Fianna Fáil's hit single
##  Christmas | will be," I saw NAMA
##  Christmas | because they must take the decision to
##  Christmas | . With a possible election next year
##  Christmas | ? Is the Society of St.
##  Christmas | time people were laden down with shopping
##  Christmas | bonus. Of course, that is
```

Word context is important

```
mwes <- tokens(data_corpus_irishbudget2010) %>%
  tokens_remove(pattern = stopwords("english"), padding = TRUE) %>%
  textstat_collocations(size = 2)

head(mwes, 8)
```

```
##      collocation count count_nested length   lambda        z
## 1 social welfare    70            0      2 8.081143 28.82286
## 2  child benefit    45            0      2 8.320640 24.96713
## 3      next year    37            0      2 6.711856 24.00550
## 4 public service    60            0      2 7.527766 23.23233
## 5       per week    25            0      2 7.111580 21.99013
## 6  public sector    30            0      2 5.143782 21.37840
## 7   labour party    21            0      2 6.992251 19.92961
## 8    green party    20            0      2 6.925392 19.58852
```

quanteda functions for the typical workflow

Step-by-step workflow

1. Reading in texts (`readtext`)
2. Corpus (`corpus`)
3. Tokenization (`tokens`)
4. Document-feature matrix (`dfm`)
5. Textual statistics (`textstat`)
6. Text scaling models (`textmodel`)
7. Textual data visualization (`textplot`)
8. Other textual analysis, such as topic models, word embeddings, deep learning (interoperability with **topicmodels**, **stm**, **text2vec**, **keras**)

Functions for corpus

A **corpus** object contains texts with document-level variables

Function	Description
`corpus()`	construct a corpus
`corpus_reshape()`	recast the document units
`corpus_segment()`	segment text into component elements
`corpus_subset()`	extract a subset of a corpus
`corpus_trim()`	remove sentences based on their token length

Functions for tokens

A **tokens** object contains individual words or symbols as tokens

Function	Description
`tokens()`	Tokenize a set of texts
`tokens_compound()`	Convert token sequences into compound tokens
`tokens_lookup()`	Apply a dictionary to a tokens object
`tokens_select()`, `tokens_remove()`	Select or remove tokens
`tokens_ngrams()`, `tokens_skipgrams()`	Create ngrams and skipgrams
`tokens_tolower()`, `tokens_toupper()`	Convert the case of tokens
`tokens_wordstem()`	Stem the terms in an object

Functions for document-feature matrix

A **dfm** object contains frequencies of words or symbols in a matrix

Function	Description
`dfm()`	Create a document-feature matrix
`dfm_group()`	Recombine a dfm by a grouping variable
`dfm_lookup()`	Apply a dictionary to a dfm
`dfm_select()`, `dfm_remove()`	Select features from a dfm or fcm
`dfm_weight()`	Weight a dfm
`dfm_wordstem()`	Stem the features in a dfm
`fcm()`	Feature co-occurrence matrix

Statistical analytic functions

textstat_*() functions perform statistical analysis of textual data

Function	Description
`textstat_collocations()`	Calculate collocation statistics
`textstat_dist()`, `textstat_simil()`	Distance/similarity computation between documents or features
`textstat_keyness()`	Calculate keyness statistics
`textstat_lexdiv()`	Calculate lexical diversity
`textstat_readability()`	Calculate readability

Machine learning functions

textmodel_*() functions perform machine learning on textual data

Function	Description
`textmodel_ca()`	Correspondence analysis of a dfm
`textmodel_lsa()`	Latent semantic analysis of a dfm
`textmodel_nb()`	Naive Bayes (multinomial, Bernoulli) classifier
`textmodel_wordscores()`	Laver, Benoit and Garry (2003) text scaling
`textmodel_wordfish()`	Slapin and Proksch (2008) scaling model
`tefxtmodel_affinity()`	Perry and Benoit (2017) class affinity scaling
`convert()`	Interface to other packages (**topicmodels**, **stm** etc.)

Note: **quanteda.classifiers** under development

Visualization functions

textplot_*() functions plot textual data

Function	Description
`textplot_scale1d()`	Plot a fitted scaling model
`textplot_wordcloud()`	Plot features as a wordcloud
`textplot_xray()`	Plot the dispersion of key word(s)
`textplot_keyness()`	Plot association of words with target vs. reference set

Accompanying packages

readtext: import text files

- A one-function package that does exactly what it says on the tin
- Available file formats: `txt`, `csv`, `tsv`, `tab`, `json`, `xml`, `pdf`, `docx`, `doc`, `xls`, `xlsx`, `rtf`
- Can important multiple files at one time with
  - a wildcard value (filepath + glob)
  - URL
  - file archives (e.g. `tar`, `tar.gz`, `zip`)

Import text from URL and get most frequent terms

```
library(readtext)

# read PDF file from URL
url <- 'https://theoj.org/joss-papers/joss.00774/10.21105.joss.00774.pdf'
dat <- readtext(url)
```

```
# get 10 most frequent terms
corpus(dat) %>%
  dfm(remove_punct = TRUE, remove_numbers = TRUE) %>%
  dfm_remove(pattern = stopwords("en")) %>%
  topfeatures(n = 10)
```

```
##   quanteda    package   analysis          r       text       data
##         27         27         22         17         15         12
##  functions     benoit    textual processing
##         12         12         11         11
```

spacyr: an R wrapper for SpaCy

- Returns data-frame of POS tagged tokens from text
- Options: POS-tagging, lemmatization, dependency parsing, named-entity extraction
- Using **reticulate** in backend
- Can use numerous language models in spaCy
- Automatically detect spaCy installation from all python executables available in the system

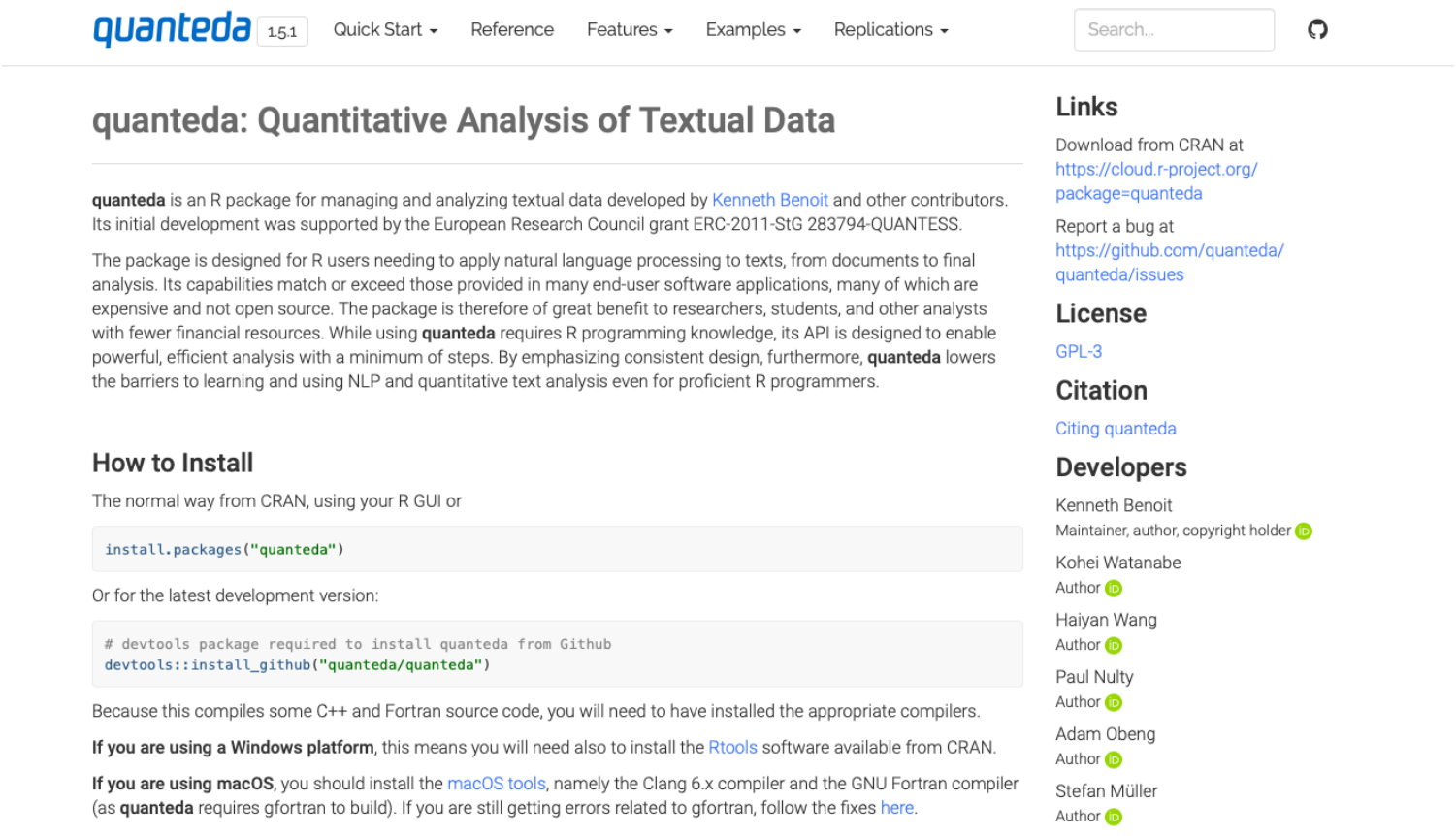
spacyr: workflow

```
library(spacyr)

# initialize spacy
spacy_initialize(model = "en")
txt <- "quanteda is an R package providing a comprehensive workflow and toolk

# parse text
spacy_parse(txt)
```

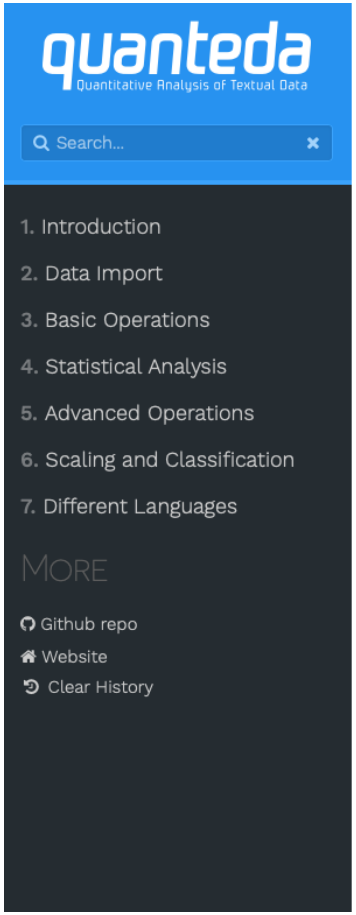
```
##    doc_id sentence_id token_id         token         lemma   pos entity
## 1   text1           1        1      quanteda      quanteda  NOUN
## 2   text1           1        2            is            be  VERB
## 3   text1           1        3            an            an   DET
## 4   text1           1        4             R             r  NOUN
## 5   text1           1        5       package       package  NOUN
## 6   text1           1        6     providing       provide  VERB
## 7   text1           1        7             a             a   DET
## 8   text1           1        8 comprehensive comprehensive   ADJ
## 9   text1           1        9      workflow      workflow  NOUN
## 10  text1           1       10           and           and CCONJ
## 11  text1           1       11       toolkit       toolkit  NOUN
## 12  text1           1       12           for           for   ADP
## 13  text1           1       13       natural       natural   ADJ
## 14  text1           1       14      language      language  NOUN
## 15  text1           1       15    processing    processing  NOUN
## 16  text1           1       16         tasks          task  NOUN
## 17  text1           1       17             .             . PUNCT
```

Additional resources

Documentation: https://quanteda.io



Extensive tutorials: https://tutorials.quanteda.io



QUANTEDA TUTORIALS

By Kohei Watanabe and Stefan Müller

This website contains a step-by-step introduction to quantitative text analysis using **quanteda**. The chapters cover a brief introduction to the statistical programming language R, how to import text data, basic operations of **quanteda**, how to construct a corpus, tokens objects, a document-feature matrix, and how to conduct advanced operations. The final chapter deals with text scaling (e.g., Wordscores, Wordfish, correspondence analysis), document classification using Naive Bayes and topic models.

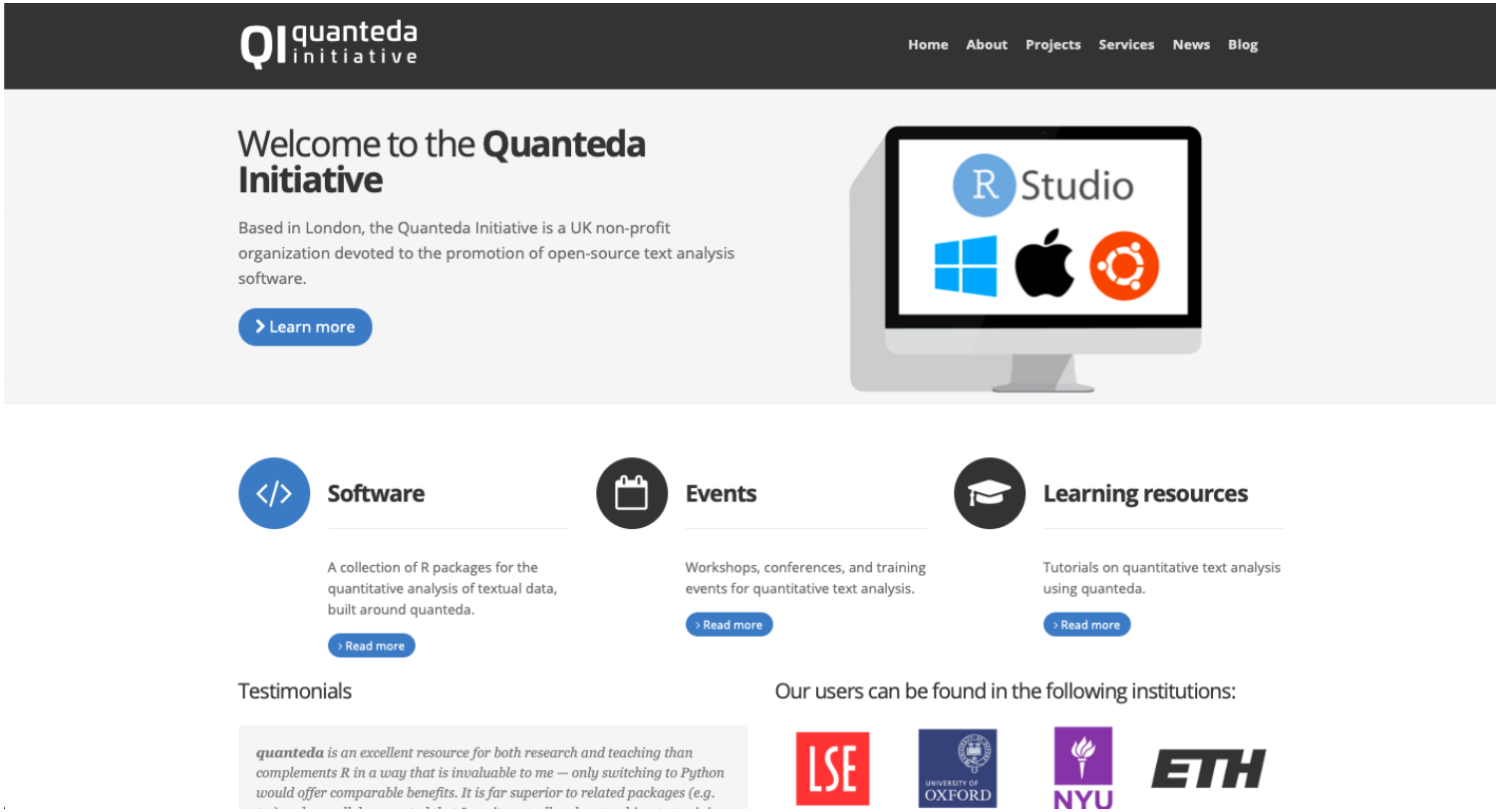
The six chapters consist of over 30 sections. If you click on the name of a chapter on the left-hand side of this page, the sections will pop up. You can also use the "Search" field in the top-left corner to look up the occurrence of certain terms or R functions covered in the tutorials.

This website is created for workshops held by the **quanteda** team and for users who look for a comprehensible step-by-step introduction to text analysis using R. We have also created several additional useful resources, such as vignettes, replications, a cheatsheet and a comparison to other text analysis packages (in terms of functions to get you started.

You can not only see the R commands but execute them yourself if you download the source code of this website from the Github repository. You should unzip the files on your machine and click `quanteda_tutorials.Rproj` to open RStudio. Executable R commands are in the `.Rmarkdown` files under the `content` folder.

Contributions in the form of feedback, comments, code, and bug reports are most welcome. If you have questions on how to use **quanteda**, please post them to the quanteda channel on StackOverflow. If you find a bug, please report it to the quanteda issues. *We prefer these platforms to emails in communicating with our users* because the records will help other users who have similar problems.

Dissemination: https://quanteda.org



How to reward software development?

Kenneth Benoit, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 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.

Official laptop stickers!

quanteda
Quantitative Analysis of Textual Data

Useful links

- Package documentation
- Quanteda tutorials
- Quanteda cheatsheet
- GitHub issues
- Stack Overflow