The Journal of Open Source Software

quanteda: An R package for the quantitative analysis of textual data

Kenneth Benoit[1], **Kohei Watanabe**[1], **Haiyan Wang**[2], **Paul Nulty**[3], **Adam Obeng**[1], **Stefan Müller**[4], and **Akitaka Matsuo**[1]

1 Department of Methodology, London School of Economics and Political Science **2** De Beers Inc. **3** Centre for Research in Arts, Social Science and Humanities, University of Cambridge **4** Department of Political Science, Trinity College Dublin

DOI: 10.21105/joss.00760

Software

- Review
- Repository
- Archive

Submitted: 30 May 2018
Published: 04 June 2018

Summary

quanteda is an R package providing a comprehensive workflow and toolkit for natural language processing tasks such as corpus management, tokenization, analysis, and visualization. It has extensive functions for applying dictionary analysis, exploring texts using keywords-in-context, computing document and feature similarities, and discovering multi-word expressions through collocation scoring. Based on entirely sparse operations, it provides highly efficient methods for compiling document-feature matrices and for manipulating these or using them in further quantitative analysis. Using C++ and multithreading extensively, **quanteda** is also considerably faster and more efficient than other R and Python packages in processing large textual data.

Corpus management

quanteda makes it easy to manage texts in the form of a "corpus", which is defined as a collection of texts that includes document-level variables specific to each text, as well as meta-data for documents and for the collection as a whole. With the package, users can easily segment texts by words, paragraphs, sentences, or even user-supplied delimiters and tags, group them into larger documents by document-level variables, or subset them based on logical conditions or combinations of document-level variables.

Natural language processing

quanteda is principally designed to allow users a fast and convenient method to construct a document-feature matrix from a corpus with an ability to perform the most common natural language processing tasks such as tokenizing, stemming, forming n-grams, selecting and weighting features. With these functions, users can easily remove stop words and stem words in numerous languages, select words in a dictionary, and convert frequency counts into weights, for instance using *tf-idf* scoring.

Using the ICU library in the **stringi** package (Gagolewski 2018) for text processing, **quanteda** correctly handles Unicode character sets for regular expression matching and detecting word boundaries for tokenization. Once texts are tokenized, **quanteda** serializes tokens into integers to increase processing speed while reducing memory usage. Many of the text processing functions are parallelized using the Intel TBB library via the **RcppParallel** package (Allaire et al. 2018).

Models and textual statistics

quanteda is especially suited to research because it was designed from the outset for the social scientific analysis of textual data. Its “textmodel” functions provide native, highly efficient implementations of several text analytic scaling methods, such as Wordscores (Laver, Benoit, and Garry 2003), Wordfish (Slapin and Proksch 2008), class affinity scaling (Perry and Benoit 2017), and correspondence analysis (Greenacre 1984). More general textmodel functions include efficient implementations of a multinomial Naive Bayes classifier designed specifically for textual data (Manning, Raghavan, and Schütze 2008) and latent semantic analysis (Deerwester et al. 1990). **quanteda** also works flexibly and efficiently with dictionaries, and is distributed with the 2015 version of the Lexicoder Sentiment Dictionary (Young and Soroka 2012).

In addition to models, the package provides a variety of text statistics, such as frequency analysis, “keyness”, lexical diversity, readability, and similarity and distance of documents or features. These make use of the sparseness document-feature matrices – often over 90% sparse – and parallelism for efficient, fast computation. **quanteda** also provides methods for statistically scoring collocations, useful in identifying multi-word expressions.

Text visualization

The package provides extensive methods for visualizing textual analyses, via its family of “textplot” functions. These are typically designed to take another package object as an input, to produce a specific form of plot. For instance, from a feature co-occurrence matrix, or `fcm`, we can directly plot a network using `textplot_network()`:

```r
library("quanteda")

# construct the feature co-occurrence matrix
examplefcm <-
    tokens(data_corpus_irishbudget2010, remove_punct = TRUE) %>%
    tokens_tolower() %>%
    tokens_remove(stopwords("english"), padding = FALSE) %>%
    fcm(context = "window", window = 5, tri = FALSE)

# choose 30 most frequency features
topfeats <- names(topfeatures(examplefcm, 30))

# select the top 30 features only, plot the network
set.seed(100)
textplot_network(fcm_select(examplefcm, topfeats), min_freq = 0.8)
```

Package design

quanteda has been carefully designed with several key aims in mind.

Consistency. **quanteda** functions and objects are named systematically such that `corpus()`, `tokens()` and `dfm()` construct those object types, and that `corpus_*()`, `tokens_*()` and `dfm_*()` return a modified version of these objects. Naming consistency applies also to the extensive built-in data objects in the package, whose names always start with `data_*` followed by object types. This not only gives the users a clear overview

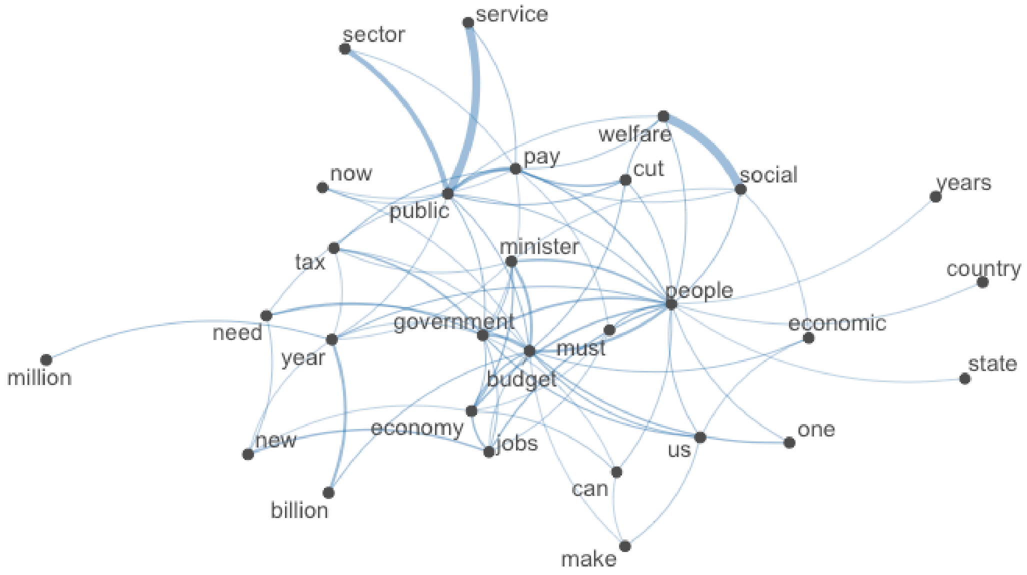

Figure 1: Feature co-occurrence network plot example.

of the package, but also makes the package more reliable for other packages that depend on it.

Accessibility. **quanteda** contains extensive manual pages structured around the naming rules. Furthermore, there are references, package vignettes, examples, and tutorials on the website at http://docs.quanteda.io. These materials help beginner users to understand how to use these functions for basic operations, and expert users to combine the functions for advanced text processing and analysis.

Performance. Built around on sparse data structures, **quanteda** can process large textual data that are difficult for other R packages (such as computing distances or scoring collocations). Its high performance further enhanced by data serialization and parallel computation implemented in C++, permitting large and fast text analysis even on computers with relatively limited resources (such as laptop computers).

Transparency and reproducibility. **quanteda** is designed to facilitate rigorous, transparent, and reproducible scientific analysis of text. Being open-source software, its source code can be scrutinized and corrected by other experts. Its functions are designed to encourage a reproducible workflow by linking successive processing tasks in a clear, readable manner.

Compatibility with other packages. For analysis not provided by built-in functions, users can move **quanteda** objects seamlessly to other packages, such as the **stm** package for structural topic models (Roberts, Stewart, and Tingley 2018) or word embedding packages like **text2vec** (Selivanov and Wang 2018). **quanteda** also works well with companion packages such as **spacyr**, an R wrapper to the spaCy (Honnibal and Montani 2017), and **readtext**, a package for importing text files into R.

Funding and Support

Created at the London School of Economics in 2013 with funding from the European Research Council funding (ERC-2011-StG 283794-QUANTESS), **quanteda** is now supported by the Quanteda Initiative, a non-profit organization founded in 2018 to provide ongoing support for the "quanteda ecosystem" of open-source text analysis software.

References

Allaire, JJ, Romain Francois, Kevin Ushey, Gregory Vandenbrouck, Marcus Geelnard, and Intel. 2018. *RcppParallel: Parallel Programming Tools for 'Rcpp'*. https://CRAN.R-project.org/package=RcppParallel.

Deerwester, Scott, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. "Indexing by Latent Semantic Analysis." *Journal of the American Society for Information Science* 41 (6):391.

Gagolewski, Marek. 2018. *R Package Stringi: Character String Processing Facilities*. http://www.gagolewski.com/software/stringi/.

Greenacre, Michael J. 1984. *Theory and Applications of Correspondence Analysis*. London: Academic Press.

Honnibal, Matthew, and Ines Montani. 2017. "SpaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing."

Laver, Michael, Kenneth Benoit, and John Garry. 2003. "Estimating the Policy Positions of Political Actors Using Words as Data." *American Political Science Review* 97 (2):311–31.

Manning, C. D., P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.

Perry, P. O., and K. Benoit. 2017. "Scaling Text with the Class Affinity Model." *ArXiv E-Prints*, October. https://arxiv.org/abs/1710.08963.

Roberts, Margaret E., Brandon M. Stewart, and Dustin Tingley. 2018. *Stm: R Package for Structural Topic Models*. http://www.structuraltopicmodel.com.

Selivanov, Dmitriy, and Qing Wang. 2018. *Text2vec: Modern Text Mining Framework for R*. https://CRAN.R-project.org/package=text2vec.

Slapin, Jonathan B., and Sven-Oliver Proksch. 2008. "A Scaling Model for Estimating Time-Series Party Positions from Texts." *American Journal of Political Science* 52 (3):705–22.

Young, Lori, and Stuart Soroka. 2012. "Affective News: The Automated Coding of Sentiment in Political Texts." *Political Communication* 29 (2):205–31.