The Journal of Open Source Software

Fast, Consistent Tokenization of Natural Language Text

Lincoln A. Mullen[1], **Kenneth Benoit**[2], **Os Keyes**[3], **Dmitry Selivanov**[4], **and Jeffrey Arnold**[5]

1 Department of History and Art History, George Mason University **2** Department of Methodology, London School of Economics and Political Science **3** Department of Human Centered Design & Engineering, University of Washington **4** Open Data Science **5** Department of Political Science, University of Washington

DOI: 10.21105/joss.00654

Software

- Review
- Repository
- Archive

Submitted: 27 March 2018
Published: 28 March 2018

Computational text analysis usually proceeds according to a series of well-defined steps. After importing texts, the usual next step is to turn the human-readable text into machine-readable tokens. Tokens are defined as segments of a text identified as meaningful units for the purpose of analyzing the text. They may consist of individual words or of larger or smaller segments, such as word sequences, word subsequences, paragraphs, sentences, or lines (Manning, Raghavan, and Schütze 2008, 22). Tokenization is the process of splitting the text into these smaller pieces, and it often involves preprocessing the text to remove punctuation and transform all tokens into lowercase (Welbers, Van Atteveldt, and Benoit 2017, 250–51). Decisions made during tokenization have a significant effect on subsequent analysis (Denny and Spirling forthcoming; D. Guthrie et al. 2006). Especially for large corpora, tokenization can be computationally expensive, and tokenization is highly language dependent. Efficiency and correctness are therefore paramount concerns for tokenization.

The **tokenizers** package for R provides fast, consistent tokenization for natural language text (L. Mullen 2018, R Core Team (2017)). (The package is available on GitHub and archived on Zenodo.) Each of the tokenizers expects a consistent input and returns a consistent output, so that the tokenizers can be used interchangeably with one another or relied on in other packages. To ensure the correctness of output, the package depends on the stringi package, which implements Unicode support for R (Gagolewski 2018). To ensure the speed of tokenization, key components such as the n-gram and skip n-gram tokenizers are written using the Rcpp package (Eddelbuettel 2013, Eddelbuettel and Balamuta (2017)). The tokenizers package is part of the rOpenSci project.

The most important tokenizers in the current version of the package can be grouped as follows:

- tokenizers for characters and shingled characters
- tokenizers for words and word stems, as well as for Penn Treebank tokens
- tokenizers n-grams and skip n-grams
- tokenizers for tweets, which preserve formatting of usernames and hashtags

In addition the package provides functions for splitting longer documents into sentences and paragraphs, or for splitting a long text into smaller chunks each with the same number of words. This allows users to treat parts of very long texts as documents in their own right. The package also provides functions for counting words, characters, and sentences.

The tokenizers in this package can be used on their own, or they can be wrapped by higher-level R packages. For instance, the tokenizers package is a dependency for the tidytext (Silge and Robinson 2016), text2vec (Selivanov and Wang 2018), and textreuse (L. Mullen 2016) packages. More broadly, the output of the tokenization functions follows

the guidelines set by the text-interchange format defined at an rOpenSci Text Workshop in 2017 (Text Workshop 2017). Other packages which buy into the text-interchange format can thus use the tokenizers package interchangeably.

The tokenizers package has research applications in any discipline which uses computational text analysis. The package was originally created for
historical research into the use of the Bible in American newspapers (L. A. Mullen forthcoming) and into the borrowing of legal codes of civil procedure in the nineteenth-century United States (Funk and Mullen 2018, Funk and Mullen (2016)). The `tokenizers` package underlies the tidytext package (Silge and Robinson 2017), and via that package tokenizers has been used in disciplines such as political science (Sanger and Warin, n.d.), social science (Warin, Le Duc, and Sanger, n.d.), communication studies (Xu and Guo 2018), English (Ballier and Lissón 2017), and the digital humanities more generally.

References

Ballier, Nicolas, and Paula Lissón. 2017. "R-Based Strategies for DH in English Linguistics: A Case Study." In *Teaching NLP for Digital Humanities (Teach4DH) Co-Located with GSCL 2017.*, 1918:1–10. Proceedings of the Workshop on Teaching Nlp for Digital Humanities (Teach4dh) Co-Located with Gscl 2017. Berlin, Germany: Peggy Bockwinkel. https://hal.archives-ouvertes.fr/hal-01587126.

Denny, Matthew J., and Arthur Spirling. forthcoming. "Text Preprocessing for Unsupervised Learning: Why It Matters, When It Misleads, and What to Do About It." *Political Analysis*, 49.

Eddelbuettel, Dirk. 2013. *Seamless R and C++ Integration with Rcpp*. New York: Springer. https://doi.org/10.1007/978-1-4614-6868-4.

Eddelbuettel, Dirk, and James Joseph Balamuta. 2017. "Extending extitR with extitC++: A Brief Introduction to extitRcpp." *PeerJ Preprints* 5 (August):e3188v1. https://doi.org/10.7287/peerj.preprints.3188v1.

Funk, Kellen, and Lincoln A. Mullen. 2016. "A Servile Copy: Text Reuse and Medium Data in American Civil Procedure." *Rechtsgeschichte [Legal History]*, no. 24:341–43. https://doi.org/10.12946/rg24/341-343.

———. 2018. "The Spine of American Law: Digital Text Analysis and U.S. Legal Practice." *American Historical Review* 123 (1):132–64. https://doi.org/10.1093/ahr/123.1.132.

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

Guthrie, David, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. 2006. "A Closer Look at Skip-Gram Modelling." In *Proceedings of the 5th International Conference on Language Resources and Evaluation*. http://www.lrec-conf.org/proceedings/lrec2006/pdf/357_pdf.pdf.

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

Mullen, Lincoln. 2016. *Textreuse: Detect Text Reuse and Document Similarity*. https://github.com/ropensci/textreuse.

———. 2018. *Tokenizers: Fast, Consistent Tokenization of Natural Language Text*. http://lincolnmullen.com/software/tokenizers/.

Mullen, Lincoln A. forthcoming. *America's Public Bible: Biblical Quotations in U.S. Newspapers*. Stanford University Press. http://americaspublicbible.org.

R Core Team. 2017. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Sanger, William, and Thierry Warin. n.d. "The 2015 Canadian Election on Twitter: A Tidy Algorithmic Analysis."

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

Silge, Julia, and David Robinson. 2016. "Tidytext: Text Mining and Analysis Using Tidy Data Principles in R." *JOSS* 1 (3). The Open Journal. https://doi.org/10.21105/joss.00037.

———. 2017. *Text Mining with R: A Tidy Approach*. O'Reilly. http://tidytextmining.com/.

Text Workshop. 2017. *Tif: Text Interchange Format*. https://github.com/ropensci/tif.

Warin, Thierry, Romain Le Duc, and William Sanger. n.d. "Mapping Innovations in Artificial Intelligence Through Patents: A Social Data Science Perspective."

Welbers, Kasper, Wouter Van Atteveldt, and Kenneth Benoit. 2017. "Text Analysis in R." *Communication Methods and Measures* 11 (4):245–65. https://doi.org/10.1080/19312458.2017.1387238.

Xu, Zhan, and Hao Guo. 2018. "Using Text Mining to Compare Online Pro- and Anti-Vaccine Headlines: Word Usage, Sentiments, and Online Popularity." *Communication Studies* 69 (1):103–22. https://doi.org/10.1080/10510974.2017.1414068.