An Open, Reproducible Method for Teaching Text Analysis with R

Tassie Gniady
ctgniady@iu.edu
Indiana University, United States of America

Eric Wernert
ewernert@iu.edu
Indiana University, United States of America

Over the past year and a half, the Cyberinfrastructure for Digital Humanities (CyberDH) Group at Indiana University has been developing an open instructional workflow for text analysis that aims to build algorithmic understanding and basic coding skills before scaling up analyses (Gniady et al., 2017). We have chosen to bootstrap in R, a high level and high productivity language, with methods that are open, repeatable, and sustainable. The aim is to provide code templates that can be adapted, remixed, and scaled to fit a wide range of text analysis tasks. This poster presents our approach to teaching computational text analysis and a representative hypothetical case study in which two different users are able to start with the same corpus and adapt code to achieve very different end results in a way not currently possible with black box tools.

This paradigm is fundamentally different from that currently practiced by many in the digital humanities. Black-boxed tools with GUIs that hide computation are very popular for introducing new practitioners of text analysis in the digital humanities to basic algorithms and outputs. In 2012, AntConc was downloaded 120,000 times by users in 80 different countries (Anthony, 2014). Voyant 1.0 had 113 sites linking to it actively in 2012 (Sinclair and Rockwell, 2013) and the week Voyant 2.0 was released the server went down multiple times from excess traffic (@VoyantTools, 2016). However, one of its default corpora is the Shakespearean dramas, with speaker names and stage directions. ((Sinclair and Rockwell, 2016). The inclusion of speaker names skews all algorithms related to frequency counts of characters (e.g. word clouds), which a new user may not even think to take into account. Using AntConc's concordance tool with a Shakespearean corpora including speaker names gives an idea of when a character speaks **and** when a character is mentioned, but this conflation might not jump out at a new user. If anything, we suggest learning about algorithms **first** and then moving up to black-box tools when one has the means to critique them.

Having looked at popular "plug-and-play" tools for corpora visualization, it becomes evident that even simple visualizations can lead to inaccurate results if the user is not thinking through how a corpus is being processed to produce a result. We believe that if the user understands how the algorithm is generating visualizations, they can contribute more meaningfully to critiques of sophisticated algorithms when partnered with programmers or even go on to bootstrap themselves with awareness of their domain's particular caveats. Thus, we advocate teaching humanists the basics of coding to create **conversant programmers** similar to the methodology behind Matthew Jockers' *Text Analysis with R for Students of Literature*, but with a slightly slower ramp up. To this end we have a three-step process of introducing R: web-deployed Shiny apps, highly marked up RNotebooks, and lightly commented RScripts, both in "regular" and higher performance versions. All are available for download on Github (with associated sample data from Shakespeare and Twitter) (CyberDH Team, 2017). We hope that this simpler bootstrapping method that mixes code and explanation, pedagogy and self-driven inquiry, will be of use to those looking to onramp new practitioners who may go on to partner with programmers if needed or to remix available code to look at their own knowledge domain.

Bibliography

Anthony, L. (2016). Antconc 3.4.4. Software. http://www.laurenceanthony.net/software/antconc/.

Gniady, T. Thomas, G. and Kloster, D. (2017). *Text Analysis Github Repository*. https://github.com/cyberdh/Text-Analysis.

Jockers, M. (2014). *Text Analysis with R for Students of Literature*. New York: Springer International Publishing.

Sinclair, S. and Rockwell, G. (2013). "Voyant Notebooks: Literate Programming, Programming Literacy." *Digital*

Humanities 2014: Conference Abstracts. Nebraska-Lincoln: http://dh2013.unl.edu/abstracts/ab-295.html.

Sinclair, S. and Rockwell, G. (2016). *Voyant Tools*. http://voyant-tools.org/.

@VoyantTools. Twitter. 8 April 2016.