Renaissance and Reformation
Renaissance et Réforme

McGann, Jerome, project dir. *Juxta*. Open-source tool and web service

Matthew Evan Davis

Volume 41, Number 4, Fall 2018

URI: https://id.erudit.org/iderudit/1061921ar
DOI: https://doi.org/10.7202/1061921ar

See table of contents

Publisher(s)

Iter Press

ISSN

0034-429X (print)
2293-7374 (digital)

Explore this journal

Cite this review

Davis, M. E. (2018). Review of [McGann, Jerome, project dir. *Juxta*. Open-source tool and web service]. *Renaissance and Reformation / Renaissance et Réforme*, *41*(4), 181–185. https://doi.org/10.7202/1061921ar

The *Lost Plays Database* is exactly what the digital scholar needs to tackle a subject in the twenty-first century: an open, dynamic resource, rich in scholarly content, that throws much new light on a forgotten topic.

PAUL BROWN
De Montfort University

McGann, Jerome, project dir.
***Juxta*. Open-source tool and web service.**
Applied Research in Patacriticism (ARP), Charlottesville, VA: University of Virginia, 2012. Accessed 31 March 2018. juxtasoftware.org/.

At its heart, *Juxta*—in all its iterations and just as the website advertises—is a tool for collation. A user uploads or creates a number of files representing various witnesses of a text. *Juxta* then compares the differences between these representations and displays them for the user in the form of neat, easily understood visualizations. It has had a great deal of success as a tool for both scholarship and pedagogy, and it is incredibly useful as such.

Looking more closely, however, there are some points of concern. The juxtasoftware.org site gives the impression of being abandoned, with a page of recent posts that are simply the number "1" alongside a date. The "Recent Tweets" section has nothing since 2013. The juxtacommons.org companion site *seems* much more up-to-date, but this may just be an artifact of not having any sort of dating, thus avoiding the impression of posts being "old." Based on this perception, I assume users are intended to utilize the *Juxta Commons* site (which I will refer to as "*Juxta*" from here on out) in the future and that the older, standalone tool is provided simply as a courtesy. This is borne out when a user goes to the "download" page on the *Juxta* software site, which indicates that the offline version is a "legacy" piece of software. While I appreciate the desire to avoid having to update two separate codebases (as evidenced by the two separate .git repositories for *Juxta*—github.com/performant-software/juxta-desktop and github.com/performant-software/juxta-service, respectively), the decision to deprecate the offline version of *Juxta* is a shame. A standalone tool can be used both off and online, making it able to be used on the fly in archives or (as

happens with some of my work) in remote locations where a quick transcription cannot be easily compared against a body of existing work.

Architecturally, *Juxta* runs on the "Gothenburg Model" of collation, wherein an uploaded work is "tokenized," or separated into individual words, compared using a collation service, and output as a visualization showing the differences between the variants. *Juxta* shows its adherence to this model in its user interface. Upon opening the site, a user is presented with a screen separated into four areas—three at the top for adding sources, selecting witnesses, and creating comparison sets, respectively, and one large box at the bottom for displaying the various visualizations created by those comparisons. The three boxes for selection and the box for visualization can be hidden by means of a set of up and down arrows on the leftmost portion of the screen. Additionally, a number of different file types—plain text files, html, xml, rtf, pdf, and both Word .doc and .docx files—can be used as sources. This Swiss army knife approach makes *Juxta* a very useful tool for instruction, as users do not have to learn the intricacies of XML in order to produce a usable collation for their purposes. This means classroom focus can be spent more on the actual comparison of texts than on their encoding. One thing I would have liked to see, however, is the ability to mass upload files. The ability to delete files and move them through the process in groups is there, found by clicking on the title of each of the three boxes, so it is a shame that you cannot upload sources in groups as well.

I was also pleased to see that *Juxta* avoids one thing that worries me regarding the Gothenburg Model as usually described: the notion that spelling should be normalized during the ingestation and comparison process in order to create a more streamlined set of tokens. While I understand that this might be desirable for a later printed text with a more standard spelling pattern, the types of medieval and early modern texts I work with often have variations in spelling that are semantically important either to understand dialect or to understand orthographic change. Additionally, the variation in spelling presented by a word like "France" breaks most Levenshtein distance-based, "fuzzy" spelling normalization algorithms. Happily, the *Juxta* implementation of the process does not seem to include a normalization aspect. My test files (several XML files encoded using the TEI's <sourceDoc>-based embedded transcription method from my *Minor Works of John Lydgate* archive) did not show any sign of

attempted spelling normalization and the comparison between them correctly showed spelling variants.

Even more happily (and really the only reason I was willing to share works in progress on a public commons), *Juxta* avoids what I see as a flaw of some implementations of TEI repositories: sometimes in order to use them you must put your files on a public forum on somebody else's server without the option to keep your work private. While this is certainly in keeping with the push by libraries, archives, and granting bodies towards open access and collaboration (ultimately a net benefit to scholarship), earlier career scholars or those in precarious positions may have legitimate reasons to want to hold off on opening their work up to the public at large. I was pleased to see that Performant Software, the primary technical designers of *Juxta*, recognize this. They provide three different methods for adding files—an upload feature, the ability to link to an externally-hosted file, and the ability to create sources directly within the tool itself. All three of these options work quite cleanly and the resulting files can be moved through the process without any sort of issue. That said, I would have liked some indication as to whether the file is ultimately hosted locally or linked to remotely, as I imagine it could be very useful and powerful to be able to host works in progress on another site and have the source automatically update in *Juxta* accordingly upon next login. As it appears, the comparison sets are compared then stored statically. The ability to edit or resample comparison sets would help to implement such a feature without adding too much processing overhead.

The resulting visualizations come in three forms: the first is a heatmap, where the various witnesses are provided in a list on the leftmost portion of the lower box. Depending on which witness is selected as a base text (the default is the topmost witness), the text of the witness is displayed in a larger box to the left, with variations denoted by blue highlighting over the word. The more variation present, the darker the highlighting. Clicking on a highlighted word will present a popup where the differences between witnesses for that section of the work are displayed as well as the option to annotate either the difference between all the witnesses or individual witnesses. Any annotations provided by the base text—for example, through a <note> tag in a TEI-encoded XML document—can be displayed via a "Notes" button in the upper right of the box. It does not appear, however, that new annotations can be added unless there's a difference between witnesses. This means that informational notes regarding

aspects of a work discovered while working with the comparison set—such as, for example, information about the scribal hand of a particular text or about a word chosen by the author—cannot be added directly.

The second visualization form directly compares two witnesses by presenting them side by side in the lower window. Changes between the two texts are highlighted in blue and a bar runs between the text compared on each side. When a word is clicked upon, the correspondence becomes darker, clarifying which comparison is actually meant. Additionally, the corresponding text in the other witness becomes darker as well. This is an incredibly intuitive way to understand the connection between the two witnesses. However, there is not only no way to add notes in this mode, you cannot view the already existing notes on each witness. Thus, the histogram is provided but gives no contextual information to allow a viewer to understand what exactly they are looking at.

The resulting comparisons can be output in a number of different ways. The first is a TEI-encoded XML file, utilizing the parallel segmentation method, that combines all of the witnesses. This could be very useful if you had a plaintext file, word document, or other source that you wanted to convert into an XML file for sharing and display. Alternately, *Juxta* provides an "edition starter": an experimental tool to use their comparison and convert it in a .docx or .html edition, with the variants noted at the bottom of the page and the witnesses marked with sigla of the user's choosing. Confusingly, however, the base text is the final file in the list rather than the one the user was currently working with, which could be a problem. The final option is not so much an output as a means to look at the various witnesses in Susan Schriebman's Versioning Machine. This could be very useful as a quick reference when dealing with a large body of witnesses.

For my particular set of files, however, the result of both the collation process and the various outputs left a little to be desired. While the comparison between witnesses was accurately displayed and I could easily see the differences between them, my encoding method meant that *Juxta* could not accurately divide the text (a poem by the fifteenth-century poet and dramatist John Lydgate) into stanzas. I suspect *Juxta* looks for the <lg> tag in TEI-encoded texts and, lacking that, simply places all the content into a single block. Moreover it often left paratextual information, such as the labels I use to indicate where the stanza can be found in an Early English Text Society edition, attached to the text. This has the potential to create the impression of difference where

none actually exists. Creating text files of a sample sentence, with deliberate differences in both word presentation and layout, indicates that *Juxta* will present the text as the user expects when presented with a text it can parse into its preferred internal format.

There does not seem to be a quick fix to this problem of presentation, however. The desktop version of *Juxta* provides instructions for creating and editing "parsing templates," which provide a set of instructions for the tool to use when dealing with a particular XML file. In the online version such parsing templates either do not exist or are not immediately apparent to a user. Instead, *Juxta Commons* provides the ability to ignore punctuation and capitalization and a number of levels of sensitivity to hyphens. While useful, the coarseness of these choices only underscores the lack of functionality present when compared with the "legacy" version provided for the desktop. That desktop version also allowed for the addition of critical annotations in ways that seem much more limited in the *Juxta Commons* version of the tool. Additionally, documentation of the *Juxta Commons* tool is entirely nonexistent, instead pointing to a set of instructions for API calls to the tool rather than the fairly robust documentation of the desktop version. And finally, while the Versioning Machine integration is quite welcome, the fact that it currently uses Versioning Machine 4.0 rather than 5.0 (which has been out for two years as of 31 March 2018) underscores that no significant development has occurred on either the online or desktop versions of *Juxta* for almost five years. To me, this begs the question of what happens when the version of Java or Ruby that *Juxta* expects to run on is deprecated to the point of no longer being supported. Ideally, the tool will receive some attention and an update, because the online version has quite a lot of potential and is still incredibly useful even in its current state. I would like to see what sorts of possibilities a fully-functional online version of the desktop tool, integrated with the latest version of other tools created over the last five years, might provide.

MATTHEW EVAN DAVIS
McMaster University