Renaissance and Reformation
Renaissance et Réforme

brat Rapid Annotation Tool. Web-based annotation and visualization tool

Jon Saklofske

Volume 42, numéro 2, printemps 2019

URI : https://id.erudit.org/iderudit/1065133ar
DOI : https://doi.org/10.7202/1065133ar

Aller au sommaire du numéro

Éditeur(s)

Iter Press

ISSN

0034-429X (imprimé)
2293-7374 (numérique)

Découvrir la revue

Citer ce compte rendu

Saklofske, J. (2019). Compte rendu de [brat Rapid Annotation Tool. Web-based annotation and visualization tool]. *Renaissance and Reformation / Renaissance et Réforme*, *42*(2), 180–189. https://doi.org/10.7202/1065133ar

Pyysalo, Sampo, project lead.
***brat Rapid Annotation Tool.* Web-based annotation and visualization tool.**
Natural Language Processing Laboratory (NLPLab), non-affilated virtual lab, 2010. Accessed 25 October 2018. brat.nlplab.org/index.html.

Introduction

brat combines a web-based text annotation tool with a visualization environment that allows users to add customizable, structured markup to text documents via mouse clicks and menu selections. Instead of showing the underlying code structure, this markup is then graphically represented as highlights, floating tags, and visual links between parts of the original text. In other words, *brat* offers a way of encoding a document via a WYSIWYG (what you see is what you get) graphical user interface rather than within a traditional XML (eXtensible Markup Language) editor, expanding the uses and visualization potential of tagging beyond simply preparing a document for text analysis or machine readability. Annotation categories, types, and constraints can be fully customized to incorporate various markup schemas and standards or to create a unique schema. This flexibility means that the tool can be adapted for use in any project that requires an existing text document to be annotated. *brat* supports the following annotation types:

- **text span** (categorical annotations for entities)
- **relation** (connecting entities via simple edges)
- **n-ary** (linking annotations to specific roles)
- **normalization** (linking internal annotation to external weblinks)
- **freeform notes** (while designed primarily for structured, computer-readable markup, *brat* also allows users to annotate documents with non-standard comments)

Project scope and intended audience

brat was initially created as a tool for use with science-related data, biomedical documentation, and biocuration, but its customization options enable applications well beyond this original intention. This review focuses on how

brat can be applied to early modern texts. Annotation examples on its website suggest a broad potential range of use, including (but not limited to):

- normalization (within a single document, multiple ontologies can be annotated and normalized to the appropriate ontology identifier);
- chunking (dividing text into non-overlapping segments that are typically further assigned labels);
- dependency parsing (syntactic analysis);
- meta-knowledge (identifying how factual statements should be interpreted, according to their textual context).

brat can also be used as a simple visualization tool, as a means of visualizing annotation output from other tools. However, its current build does not feature ready-made tagging configurations for digital humanities (DH)-related schemas or conversion options for humanities-related markup tools. This means that while *brat* can be customized for compatibility with such DH-related standards, users who wish to employ *brat* towards these ends will have to do this setup work themselves. Despite the need to create DH-related annotation configurations, *brat* remains potentially useful for DH projects, including early modern textual scholarship work, in which documents or collections (which may feature multiple languages) require markup or annotation and would benefit from visual representations of such markup on the text itself. At its most basic level, *brat* enables users to manage and export annotations and markup for text-based documents.

Usefulness in scholarly work

brat's usefulness stems from its innovative combination of a markup tool with a visualization environment. As mentioned above, its configurability in relation to entity definitions, relation types, event definitions, and attribute definitions makes it a flexible tool that has many potential uses. Users can add manual inline annotations that are then presented in a user-friendly graphical format. It requires plain text as input data, and can export data in *brat*'s standoff format (separate .txt. and .ann files), or export visualizations in .svg, .png, or .pdf formats. *brat*'s standoff (.txt + .ann) output file format ensures that the original

text file is never directly altered (only the .ann file is modified as annotations are added), and the discrete standoff markup can be queried separately for the purposes of higher-level analysis. However, *brat*'s unique standoff format does create some problems with converting output from *brat* to more common file formats (such as XML) for use in other markup tools. Users have had to independently write their own conversion programs for this process (stackoverflow.com/questions/26705608/how-to-convert-annotation-ann-file-to-xml), which is an unfortunate additional step needed to ensure DH-related file compatibility with work initiated in the *brat* environment.

Other useful features for scholarly work include the integrated ability to engage in annotation comparison for comparing multiple sets of annotations for the same documents, almost like a Juxta tool (juxtasoftware.org) for comparing markup versions. As well, given that *brat* is a web-based tool, every annotation on a document hosted on the *brat* server receives a globally unique address for linking to, which might prove useful for sharing specific parts of the markup visualization on a particular document with broader scholarly communities. Furthering the potential to share *brat*-related work with others, annotated documents can also be embedded in web-documents as read only displays. *brat* also supports collaborative, real-time annotation of the same document by multiple users with instant in-browser updates to reflect changes made to annotations, and has built-in visualizations for annotation validation, showing errors and incomplete links via simple graphical highlights on the document itself.

Potential avenues for research made possible by the project

brat can be used as a basic markup tool to easily add markup to a corpus or dataset via Graphical User Interface (GUI) without the need to type tags. For example, the .txt file versions of Shakespeare's writings from *Folger Digital Texts* (folgerdigitaltexts.org) can be easily imported into the tool and annotated using standard or customized tag sets. It was also effortless to type the text of Sonnet 26 into the online version of the tool for markup (See Figure 1).

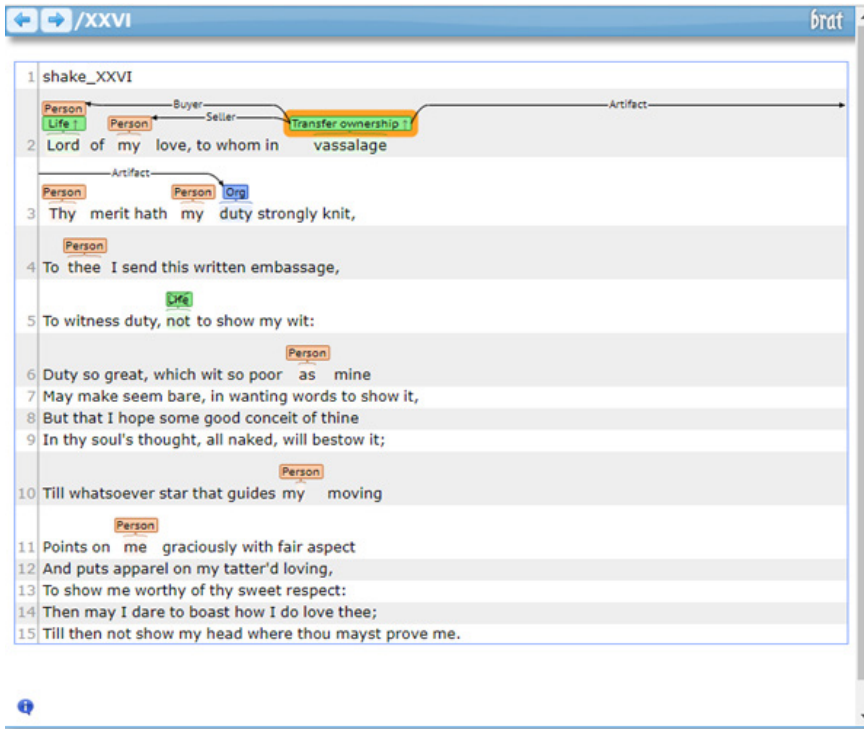

Figure 1: Annotating entities and relationships in Shakespeare's Sonnet 26 using the *brat* tool online. Note that the markup options (some of which are noticeably unsuitable to accurately describe the sonnet and its relationships) are selected and generated from the default menu of tags available to the online user. Additional or more accurate tags and relations would need to be customized by the user after locally installing *brat* on their own server.

While marking up a document with structured tags is menu-driven (following the creation of a configuration file), this process is still a manual one and can be quite laborious (see Figure 2).

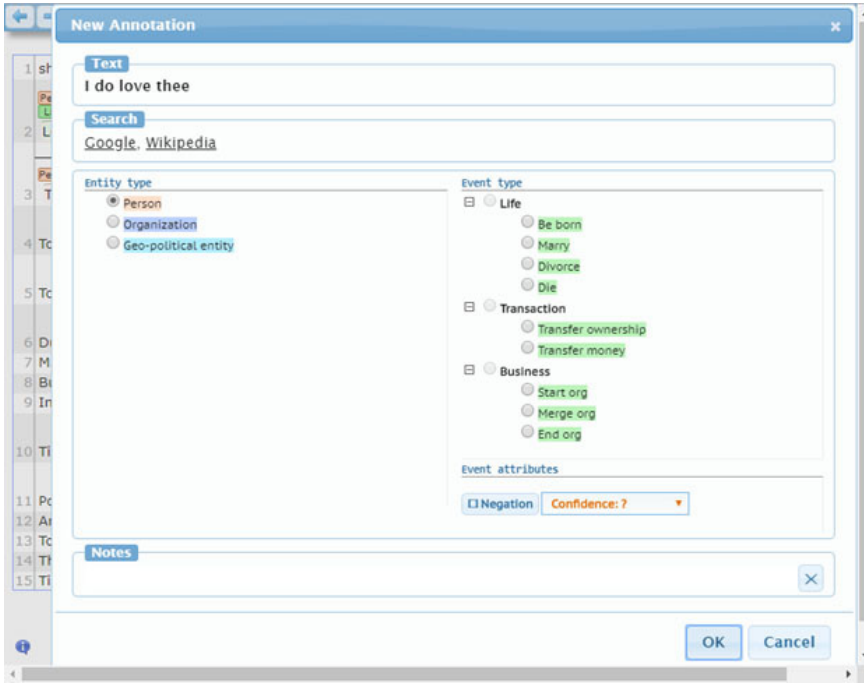

Figure 2: *brat*'s menu-driven tagging process. While this screenshot confirms *brat*'s ability to support multiple annotation types (including annotate text span, relation, n-ary, normalization, and freeform notes), the default schema offered by the online demo is quite limited and not inherently configured for use with early modern texts or humanities-related interests.

Importantly, though, *brat* features linked data annotation capabilities: it assigns a unique identifier for all annotations and includes data normalization features (brat.nlplab.org/normalization.html) that work to harmonize and integrate data from different sources into standard terminologies. *brat*'s dependable autosave feature and inherent ability to allow multiple simultaneous users to engage in the collaborative markup of documents and collections via a web-based interface suggest that it can be used as a virtual research space for geographically distributed research teams, or as a local or remote teaching tool.

Interface design and usability

I found *brat* to be easy to use and explore. Its interface is simple, uncluttered, and intuitive, and features many customization options for visual appearance and functionality. Adding annotations to a document is enabled via easy click and drag controls. *brat*'s visualizations are based on Scalable Vector Graphics (SVG), and can be rendered and exported in high quality at any scale.

There are some significant limitations, however. In regards to basic installation, while reviewing the tool I made use of the online installation demo (weaver.nlplab.org/~brat/demo/latest/#/), but from November 2018 to June 2019, it was returning errors that prevented any user from uploading or annotating new texts. Beyond such unreliability and the limitations of the schema available through the online demo, additional requirements present further difficulties and barriers for many humanities scholars. For instance, to test and configure the full potential of *brat* for this review would have required me to install it on an Apache server (which I do not currently have access to) or to install a standalone server on my local machine. As a Windows user, to install a local instance on my machine would require the enabling of a virtual machine, the installation of a Linux OS, and the mounting of *brat* on a local server within that virtual machine. My ultimate reluctance to do this in the service of this review resulted from the tool's usage limitations relating to early modern humanities scholarship that I encountered when the demo installation was functional. To import a text document into the online *brat* environment involves typing or pasting it into a dialogue box, or uploading a .tar archive, which is fairly limited. As mentioned above, there are no DH-related conversion tools or schemas included with the basic *brat* install, meaning that DH users would have to create such tools to accommodate various document formats from other markup environments into *brat*. Another major limitation of the tool is that while it is good at defining single tags and basic local relations, it is less useful for dense relations or relations across multiple lines within a single document. For example, when attempting to use *brat* in the most basic way by connecting Prospero and Miranda "Person" markup entities with a basic "family" relation between lines of the play in an imported text file version of *The Tempest* from *Folger Digital Texts*, the result was neither helpful nor visually appealing (see Figure 3).

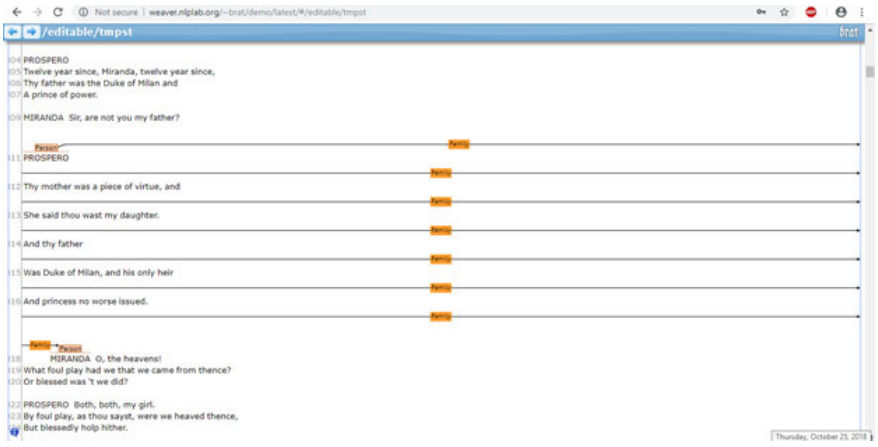

Figure 3: Attempting to link two entities labelled "person" with a "family" relation across multiple lines from *The Tempest* in *brat* produces an awkward and unhelpful visualization.

Depending on the intended uses that a researcher has for the *brat* tool in a DH and early modern context (especially in relation to non-prose source material), these limitations might be enough to recommend looking for other, more robust annotation tools. In this reviewer's opinion, the challenges facing a researcher who wants to use, adapt, and customize this tool and its functions for early modern humanities research interests outweigh the potential benefits of such a time investment.

Project's documentation

Very detailed *brat* server installation instructions are provided on the project's website (brat.nlplab.org/installation.html). Excellent user documentation is also provided, including a user manual (brat.nlplab.org/manual.html), troubleshooting page, and useful, interactive, in-tool tutorials available via the website's sample *brat* install environment. However, as mentioned above and given the unreliability of the existing, web-based demo installation, researchers should install their own instance of *brat*. The error state that prevented the online demo installation from functioning for eight months suggests that there are some persistent bugs that can prevent the tool from functioning, and the

fact that *brat* has not been updated since 2012 implies that continued developer support might be waning.

Technical elements

brat is a web-based, interactive annotation environment with a JavaScript frontend. Editing and viewing documents work best using Chrome and Safari browsers. While simple viewing of *brat* documents works well using Firefox and MS Edge browsers, editing documents in the *brat* interface via these browsers becomes buggy and unreliable.

brat files can be downloaded from the homepage or the GitHub repository (github.com/nlplab/brat) and need to be installed on an existing server (Apache recommended). *brat* is a Python program run as a common gateway interface on an existing web server. While it can be installed as an experimental standalone server on a local machine, such a feature is largely untested and the developer strongly recommends avoiding this and serving *brat* via a full web server when using sensitive data.

The *brat* server and client feature full Unicode support—supporting over one hundred different scripts and enabling annotation of texts in any language. As mentioned above, the *brat* standoff format stores text and annotations separately, meaning that the original text database is never altered by the annotation operations performed through the tool. While this is useful to maintain the integrity of the original data, it presents some problems with exporting files that are compatible with other tools.

Interoperability

Although *brat*'s file output is not inherently interoperable with many standard XML annotation tools, it can be converted. While this incompatibility might present a problem for researchers who want to share *brat*-created annotated documents more broadly, *brat* makes up for this shortcoming by also serving as a visualization environment that can share annotated documents by embedding them in existing web pages, or exporting scalable vector graphics for use in presentations. In this way, *brat* might be more useful as a tool for sharing elegant visualizations of annotation projects that have been marked up in other programs. However, conversion scripts also need to be written to enable such

importation, as the included conversion options in the existing *brat* package (github.com/nlplab/brat/tree/master/tools) are geared towards tools used in specific science research fields. Humanities researchers would have to write a Python script to convert their existing database files into two files, separating the .txt file consisting of original data from the .ann file which stores the markup annotations. One such example can be found at github.com/Edillower/XML2ANN. Given the density of markup in something like the TEI or XML versions of Shakepeare's plays from *Folger Digital Texts*, such a conversion script would not be easy to construct in every case. As well, if one did manage to execute such a conversion process, a text-based configuration file would also have to be written—following the format suggested in the *brat* documentation (brat.nlplab.org/configuration.html)—to document all of the TEI markup categories used in the original file (to ensure that they are replicated in the *brat* GUI). Given the difficulty in obtaining a useful visualization while linking annotated entities in *The Tempest*, mentioned above, I imagine that converting existing projects with markup for use in *brat* would produce additional frustrations with the way that such annotations are ultimately displayed in the tool. In short, using this tool with plain text files or files with existing markup would require a significant time investment upfront to produce something useful, and even then the tool's visualization limitations regarding texts that feature line breaks might make this an undesirable choice.

Metadata practices and standards

brat is fully configurable in relation to entity definitions, relation types, event definitions, and attribute definitions, and so can be customized to allow users to annotate text using a variety of metadata standards. However, these customization files will need to be created by the researcher who uses *brat*, as the existing configurations included with the installation package are science-related.

Licensing, copyright, and reuse

The code for the *brat* program is licensed under the open source MIT license (opensource.org/licenses/mit-license.php) and documentation under the Creative Commons Attribution License (CC-BY) (creativecommons.org/

licenses/by/2.0/). The *brat* authors feature examples of the tool's usage with freely available datasets, furthering their visible commitment to promoting the use of open access tools for use with open access data.

JON SAKLOFSKE
Acadia University

Wall, John N., project dir.
***Virtual Paul's Cross Project: A Digital Recreation of John Donne's Gunpowder Day Sermon.* Other.**
Raleigh: North Carolina State University. Accessed 27 April 2019. vpcp.chass.ncsu.edu/.

The study of Renaissance drama has in recent years fully embraced the idea that a play is text plus performance. Prominent series of editions of Shakespeare's plays, for example, focus on the play in the context of its performance history. *The Virtual Paul's Cross Project* (*VPCP*) addresses a gap in other domains of Renaissance studies where performance was equally important—the arts of discourse and, specifically with respect to oratory and the fifth office of rhetoric, delivery. The *VPCP* might therefore be best appreciated as an edition of a text in performance, where the supporting materials serve to provide context for understanding how a sermon might have been delivered and received within the circumstances of its original performance. The text and occasion featured here is Donne's sermon intended for delivery at Paul's Cross (in the churchyard of St. Paul's Cathedral) but in the event relocated inside the church owing to inclement weather. The sermon was preached on 5 November 1622, "being the Anniversary celebration of our Deliverance from the Powder Treason," as subtitled in the print edition of the sermon.[1] The core of the *VPCP* edition is not the written text (although transcriptions of both extant witnesses are provided), but rather two models that provide insight into the performance of the early modern sermon: in the first instance, a visual rendering of the space in which a Paul's Cross sermon (although in the end not this one) was delivered, and therefore some of the environmental considerations that would have affected

1. John Donne, *Fifty Sermons* (London, 1649), 398–411 sigs. [Llv]-Mm3.