Spectacles: Novel Software for Speculative Analysis in Active Archives

Peter Downs (downs@mit.edu)

Kurt Fendt (fendt@mit.edu) Regina Barzilay (regina@csail.mit.edu)

Abstract

Rather than poring through dusty stacks of books and papers, researchers can take advantage of new software tools such as full text search and shared annotations to better find answers to specific questions. Existing digital annotation and media exploration tools do not take advantage of modern technology tools to help researchers understand individual documents within the larger context of an archive. To address this need we introduce Spectacles, a web-based software tool that uses natural language processing techniques and open source technology to help researchers perform speculative analysis by creating a digital archive from any repository of digital texts. By analyzing both source texts and researcher-created annotations, rich-media notes about those source texts, Spectacles creates a uniquely useful *active archive*, a site for research that can make useful recommendations to help direct researchers to questions and answers they had not previously considered.

I. Introduction

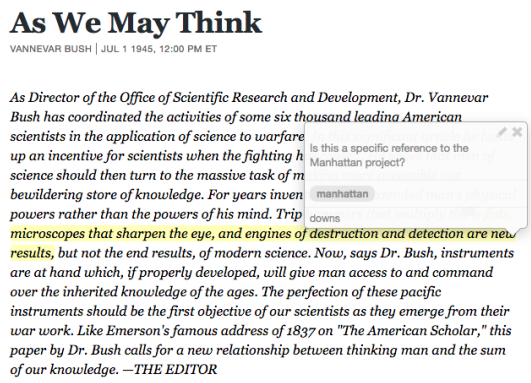
Humanities researchers rely on *archives*, collections of texts and related media, to produce new research. As digital tools have become more able to augment humanistic research, these archives have begun to move online; they have become *digital archives*. These digital archives allow researchers to use new techniques not available in offline archives: search, meta-data analysis, and hyperlinking of related media, are all new methods of interacting with texts that can assist researchers. By allowing researchers to contribute to the archive in the form of *annotations*, rich-media comments linked to specific sections of a text within the archive, these digital archives can be said to become *active archives*: archives that change over time, and allow visitors to actively contribute to their contents. Additionally, more recent natural language processing techniques such as topic modeling and entity recognition can help researchers discover hidden meaning in individual texts and new connections between them. All of these techniques

allow researchers to interact with archives in ways that were never possible before with offline materials. But many of these techniques are best suited to helping researchers answer specific questions; they are most useful when attempting to narrow the scope of analysis. Little work has been done to help researchers broaden their research scope, which might be done by providing broader context to the texts under analysis, suggesting new texts that had not been previously considered, or showing other questions being asked of the same texts by researchers from different academic disciplines and with different research goals.

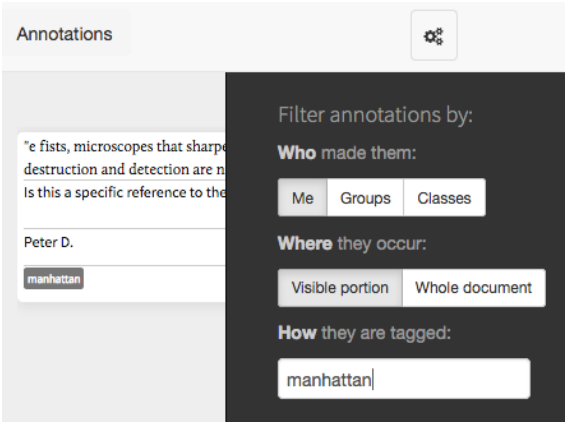
As our archives of texts become digital, it is crucial for software to enable the creation of digital archives, virtual sites of research that can aid researchers in their two primary tasks: to help answer specific questions that a researcher may have that they think can be answered by the texts in the repository, and to provide a larger context to the texts by helping researchers discover new questions to ask of the texts. As Carol Palmer writes in "Companion to Digital Humanities" [3], this *speculative analysis* has traditionally been supported by the offline archive, a place for research that encourages these sorts of interactions by merit of requiring its researchers to collocate themselves in the same physical space. As Johanna Drucker explains in the same source, "The computational processes that serve speculative inquiry must be dynamic and constitutive in their operation, not merely procedural and mechanistic." [4] To this end, we have created Spectacles, a system for creating digital archives that uses modern computational techniques to support speculative analysis.

II. Related Work

Existing software aids for research of digital texts have primarily focused on single texts in isolation, and do not contextualize texts within a broader archive. Annotation Studio [6], a suite of software tools developed by MITs Hyperstudio, allows researchers, students, and the public to create annotations — free-form comments containing images, hyperlinks, and text — linked to sections of a specific digital text. These annotations can be made visible to other users of the software, and appear collocated with the original text in such a way that every user can be said to be engaging with a "hypertext", the dynamic text comprised of the source material and its comments, as per Theodore Nelson [5]. Annotation Studio can be said to create a particular kind of active archive: a site of research allowing active contribution to, and construction of, the hypertexts it presents to its users [2]. It effectively creates active archives around single texts, enabling students and researchers to share ideas, questions, and answers — to collaboratively



(a) Annotations are shown inline with the original text.



(b) Annotations are also shown in a sidebar, and can be filtered by their tags.

Fig. 1: Reading a text and annotations in Annotation Studio. [6]

contextualize the text. But because of this focus on single texts in isolation, it is not well-suited to speculative research in archives, which are comprised of many texts.

This limitation reflects Annotation Studio's history. While it can be used by researchers, it is focused primarily on classroom use by students, collaboratively annotating a single text. Annotation Studio allows viewing annotations created by other users, but the software does not include filtering tools that researchers could use to show only annotations relevant to a specific research question. Annotations can only be found by filtering on their tags, or user-created textual labels, rather than by searching for the text within them. And because it is designed to help students analyze a single text, it does not make any attempt to automatically recommend texts and annotations to students from an archive. Archives of many texts require filtering, searching, and recommendation tools to filter signal from noise, capabilities traditionally provided by offline archives. All of these limitations are serious impediments to effective, collaborative contextualization of texts within a digital archive.

This is not a limitation of the medium: other digital humanities projects have proved the ability for computers to provide meaningful assistance in these tasks. The Tesserae tool created by Coffee, Koenig, Poornima, et. al. [7], assists researchers in discovering intertextual references among four different archives of texts: Greek, Latin, English, and Greek-Latin.

By analyzing every text in each archive and applying Latent Semantic Indexing [8], potential literary allusions between texts can be discovered by the computer and suggested to

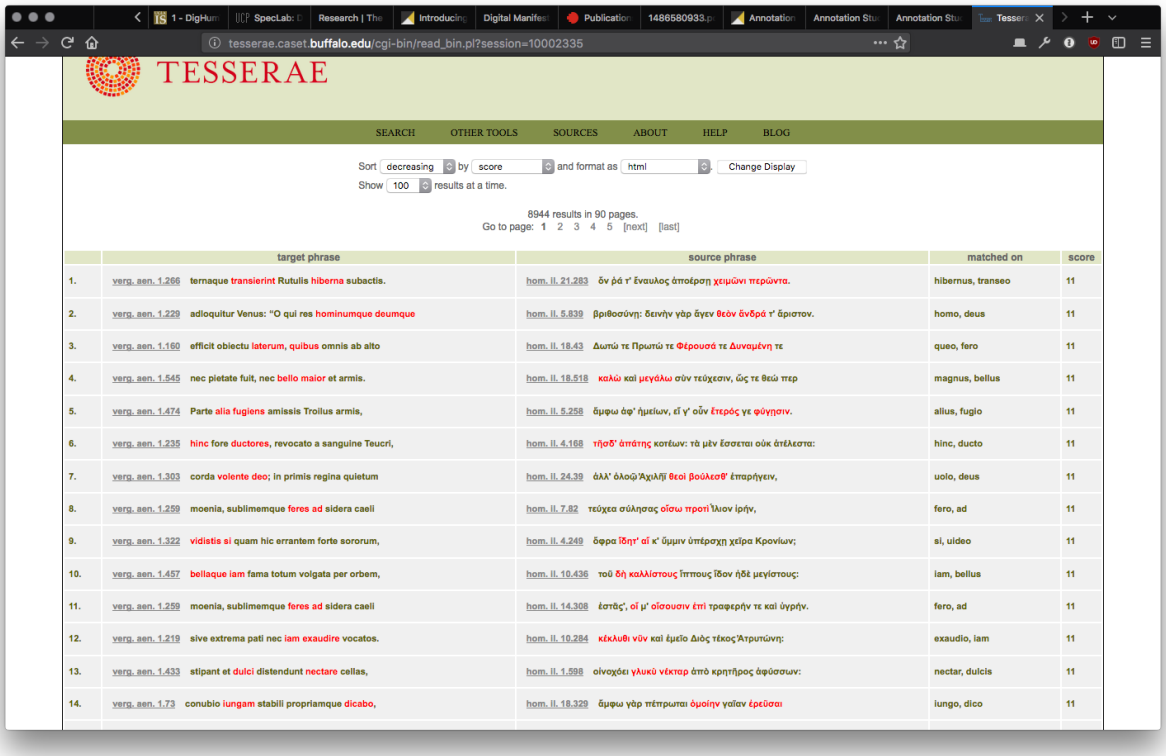


Fig. 2: Tesserae shows two texts side by side, and highlights similar phrases in red. Computational techniques determined which phrases were similar without any input from the researcher. [7]

researchers for further inquiry. Figure 2 shows the Tesserae interface for comparing two texts. While this technique did not perfectly "rediscover" all intertextual allusions previously noted by human commentators in the source texts, it found many of them, and led researchers to confirm new, previously un-noticed references, demonstrating the power of these techniques in guiding human research. This approach of applying natural language processing to a collection of texts can also be used to aid researchers in the context of an active archive: recommending passages, annotations, and entire texts similar to those the researcher has already found useful. But the Tesserae project is focused on a specific set of texts chosen by the project creators, and its software not be easily repurposed for use by researchers working with other digital archives. In addition, it lacks the collaborative contextualization capabilities found in Annotation Studio that allow researchers to situate their research notes within the archive itself.

And while the Tesserae project shows the power of natural language processing for

humanities research, but falls short of enabling active archives, the Nota Bene project [9], similarly to Annotation Studio, fails to use computational techniques to augment human research. Like Annotation Studio, it focuses on classroom use, and encourages collaborative question asking and answering to create a hypertext with which students engage. It also focuses on student analysis of single texts, rather than texts within archives. Because of this, it suffers the same flaws as Annotation Studio: it does not support speculative analysis in the context of an archive. Further, it is limited to only annotating texts in the PDF format, a restriction which makes it less useful as a tool for enabling active archives out of existing repositories of digital texts — PDFs are not well suited to hyperlinking and can be difficult to interact with outside of a desktop computer browser.

III. Results

With this history in mind, we have constructed Spectacles, a software tool that allows digital archives to present their texts in a way that specifically supports speculative analysis – exploratory research without a specific question in mind. Spectacles provides a specific digital space where many researchers can work on the same texts in a digital archive, just as a physical archive provides a physical site for research. Our key innovation is providing powerful tools to augment human research that takes advantage of the digital nature of the texts. Spectacles allows researchers to augment the texts in an archive with in-line *annotations*, rich-media comments linked to specific sections of text from a digital document in an archive. This enables collaborative contextualization: researchers can create their own annotations, and see the annotations created by other researchers, allowing for a better understanding of the texts that have been annotated. With *fuzzy search* tools, researchers can discover texts and annotations based on keyword queries, but without having to worry about exact spellings or capitalization of search terms. And Spectacles uses natural language processing to give *recommendations* of other texts and annotations that are similar to those that a researcher has already found to be relevant to the questions they are working with. These capabilities work together to support speculative analysis in active archives.

A. The Archive View

Once Spectacles has been used to construct an archive, researchers may interact with the texts and these tools through their web browser. The launching point is the Archive view, which shows all of the documents in the archive.

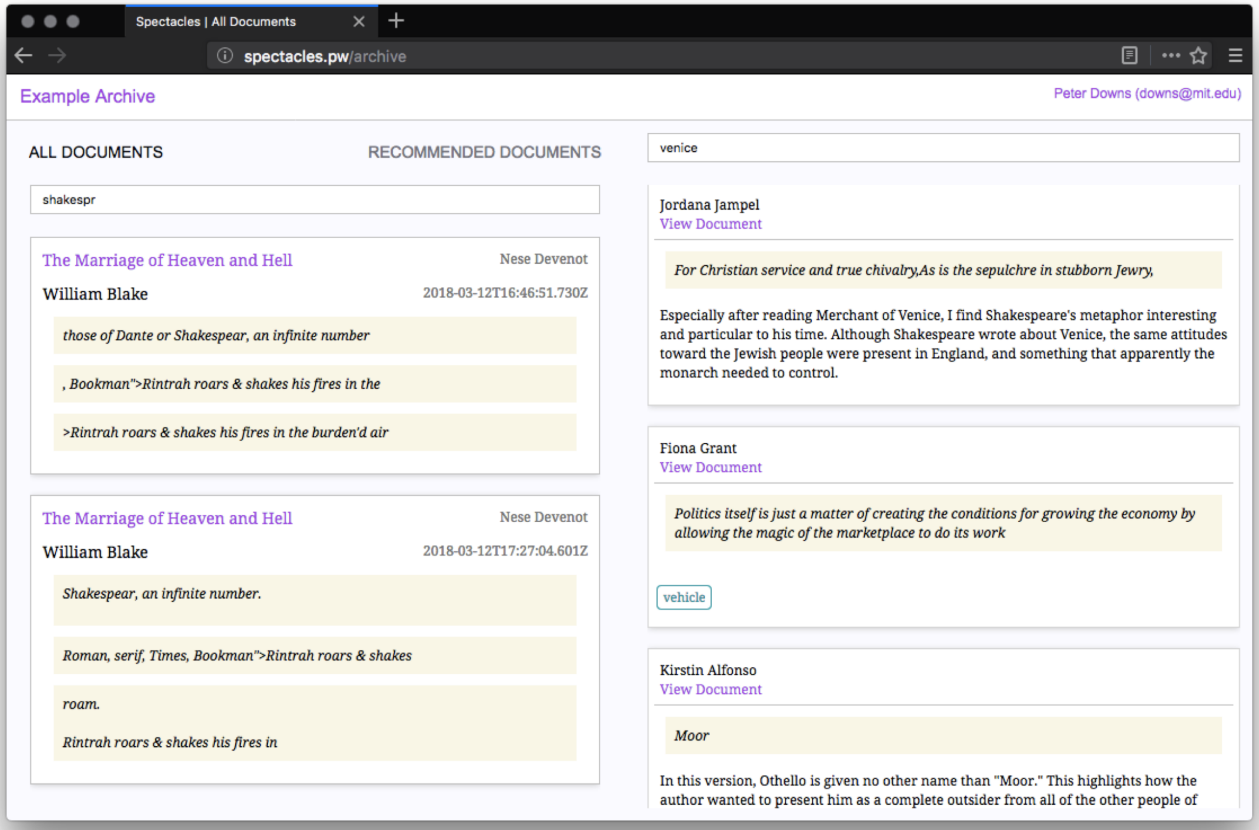


Fig. 3: The archive view allows searching over all of the texts in the archive, as well as all of the annotations. In the left pane, we see the fuzzy search showing documents that match the misspelled query “shakespr”. In the right pane, we see all annotations related to the search query “venice”.

From here, the researcher may perform searches over both the annotations and documents within the archive. The Spectacles design language makes a distinction between these two types of texts, although they may be both viewed as comprising a single metatext. Documents are viewed in the left pane of the screen, while annotations are viewed in the right side. In representations of both documents and annotations the manila background and italicized text is used to visually separate text that comes from documents within the archive from text added by researchers as comments in their annotations.

B. The Document View

Once the researcher has determined a text about which they'd like to learn more, they may navigate to it directly by clicking on a hyperlink. This takes them from the Archive view to the Document view.

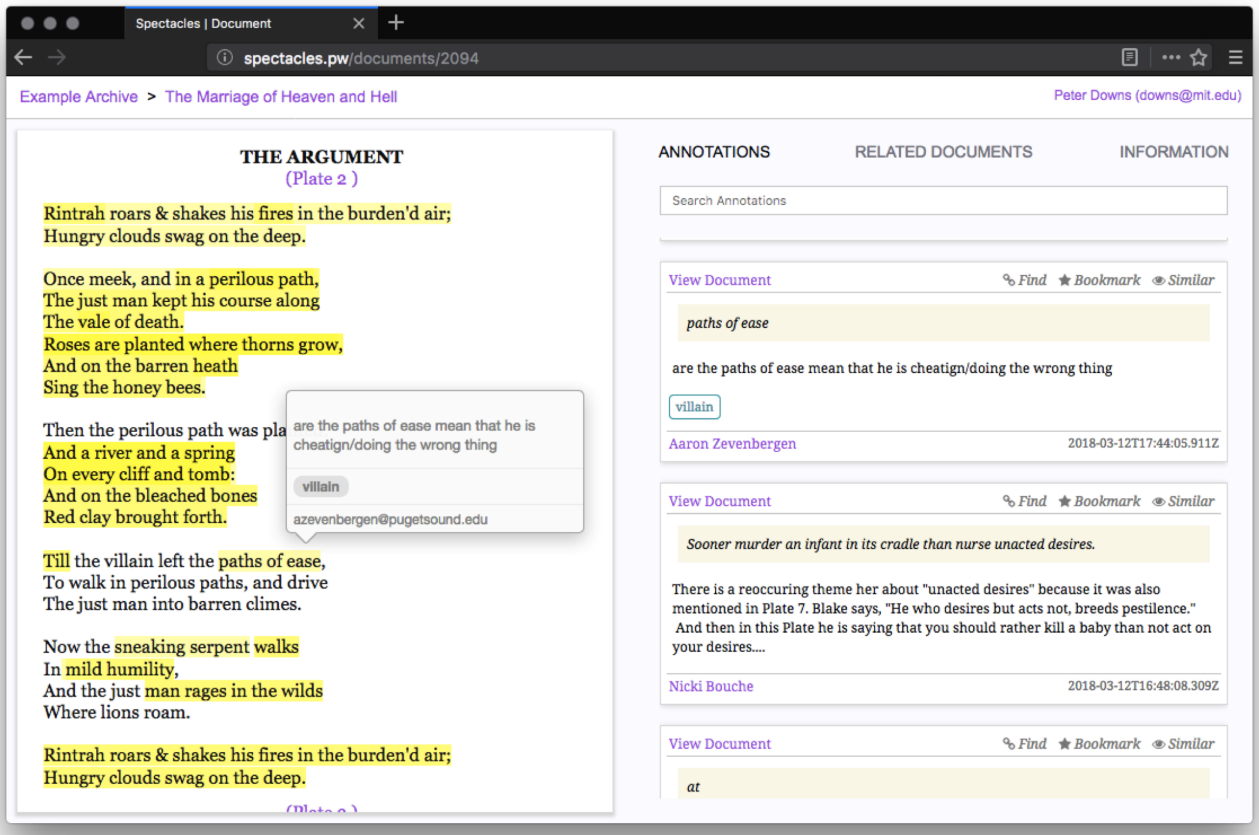


Fig. 4: The document view allows reading of the specific text in question in the left-hand pane. Annotations are shown as pop-ups over highlighted portions of the text, as well as in the right-hand pane.

This view focuses on a specific text and its annotations. The full text is presented for reading in the left-hand pane. The presence of an annotation is denoted with a yellow highlight, which a researcher can mouse over to read the annotation. By selecting any section of the text the researcher may create a new rich-text annotation, incorporating different font-sizes, hyperlinks, images, and even embedded videos. Showing annotations in-line with the source text strengthens

the experience of reading and contributing to the *meta-text* comprised of the source text and its annotations. This is how Spectacles can be said to create an active archive: the digital archive is comprised of meta-texts which change as the researchers add, delete, or update their annotations. When reading a text, one sees all of the annotations created by other researchers on the same text, helping to contextualize the text within other research traditions.

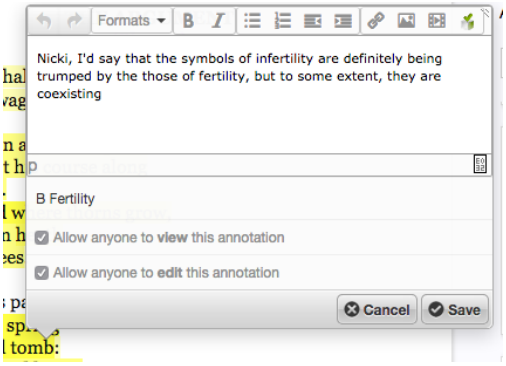
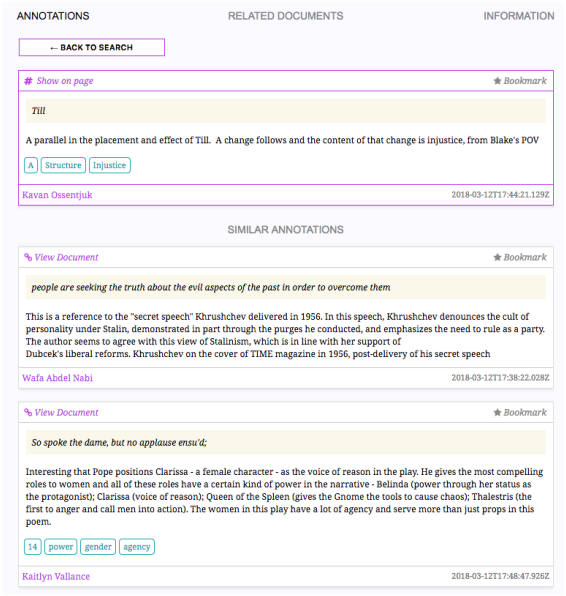


Fig. 5: The rich-text annotation editor allows researchers to comment on sections of a text. [14]

While the left-hand pane is dedicated to the meta-text, the right-hand pane performs three functions. The first focuses on the annotations of the original source text: researchers may perform the same fuzzy searches over the annotations within this text as they could in the archive view. By presenting annotations both inline with the source text and as a separate collection on the right, researchers can shift their attention between the source text and its annotations, and search through annotations without losing their place in the source text.

While viewing annotations, the researcher may be interested in finding other, similar annotations throughout the entire archive. To do so, they click the “similar” button in the upper right-hand side of any annotation, which uses natural-language processing and a high-dimensional nearest-neighbor database to efficiently find semantically-similar annotations and texts. This helps to contextualize any individual text or annotation within the archive as a whole, and is a mechanism through which a researcher may jump from viewing one text to another. Likewise, the second function of the right-hand pane is to find other, similar, texts, based on the text on display rather than on a specific annotation, as seen in 6.

Together, the annotation and recommendation capabilities that Spectacles provides offer a unique contextualization of individual texts within a larger archive and within larger research traditions. By allowing researchers to jump to related documents and annotations, Spectacles



(a) Annotations related to one that the researcher has found relevant, found from across the whole archive and not just the same document as the original annotation.

(b) Other documents related to the text that the researcher is currently reading.

Fig. 6: Researchers can ask to be recommended other annotations and texts similar to those they have found interesting.

emphasizes that every text in the archive is related to the other texts. Spectacles is unique in providing both of these contextualizations: Annotation Studio and Nota Bene only allow for annotations, and Tesserae is only capable of finding similarities between two documents explicitly chosen for comparison by a researcher. By allowing both kinds of contextualization, Spectacles allows researchers to fluidly read through the archive as a single, larger, meta-text comprised of both the source texts and the researcher-provided annotations.

IV. Implementation

Spectacles recommends annotations and texts to each researcher to investigate based on the annotations the researcher creates, as well as the texts and annotations that they mark as relevant to their interests.

Spectacles is open source software that any researcher can use to create active archives of any texts that they may see fit, rather than being released only as the interface to a specific

archive or specific archives. We see this as a firm requirement of any digital research tool, and essential to the aim of the project: to empower humanities researchers to perform speculative analysis by creating an active archive from any existing collection of digital texts. It takes the form of a web application, divided into two parts: a backend responsible for text storage and analysis, and a frontend user interface.

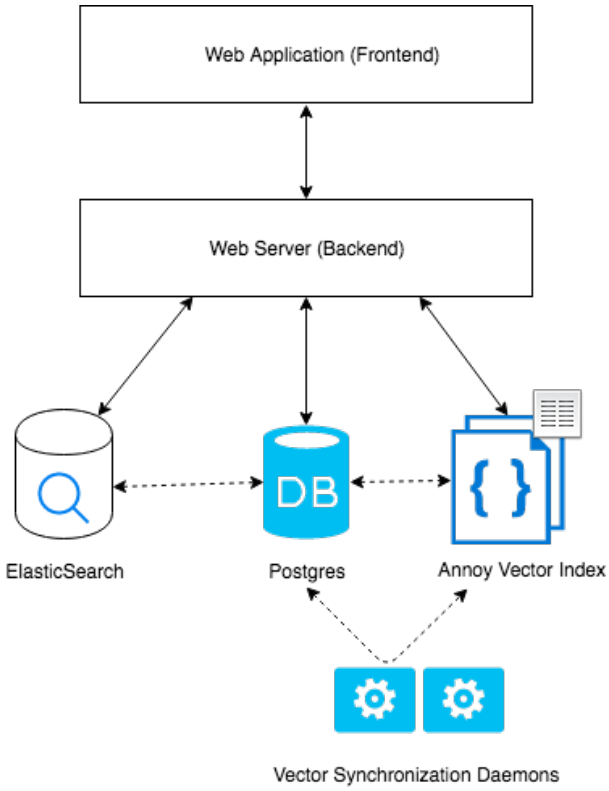


Fig. 7: Spectacles broken down into software subsystems.

A. *Backend*

The backend is written in Python, a popular open source programming language, and uses the open source Django framework [10]. Python and Django were chosen because many existing natural language processing libraries are written in the Python programming language, and Django has existing plugins for synchronizing data to the different types of databases used in Spectacles. The Django framework is used to run a web server, which is responsible for serving the frontend to the web browsers of the researchers interacting with the archive, and to

synchronize data between the many different datastores used by Spectacles provide fuzzy search and document recommendation tools.

The primary data store is in the open source Postgres [12] relational database, due to its strong performance and the support for it that Django provides. This datastore is queried by the web server, and stores every document and annotation in the archive. It implements Application Program Interfaces (APIs) for creating, reading, updating, and deleting annotations. Existing digital texts can be loaded into the system with scripts included in our code, which is freely available online under an open source license. The Postgres database acts as a source of ground truth; the other data stores, which will be described in the subsequent sections, use the Postgres database as a source and transform its data in ways that support specific application functionality.

1) Fuzzy Search: The fuzzy-searching capabilities offered by Spectacles are implemented by making queries to a secondary datastore, Elasticsearch [15], which is dedicated to performing this kind of search on text documents. Spectacles synchronizes the texts and annotations from the Postgres database to Elasticsearch every time they are created, updated, or deleted, so that both databases contain all of the same items. Although Postgres has some fuzzy-search abilities built in, the Elasticsearch query engine is much more powerful, and does a better job of matching related documents to a misspelled query. For this reason, all searches in the Spectacles application are implemented as queries to the Elasticsearch database and not the Postgres database, although the data returned to the frontend is often a mix of data from both sources.

2) Recommendations: As described above, Spectacles recommends texts and annotations to researchers based on how similar they are to other text and annotations that the researcher has found useful. To do this, we model every text and annotation as a 384-dimensional vector in a "semantic space" using the open-source Spacy [11] software package. This semantic space is a vector space defined such that the vector representations of texts that are similar in content are nearby to each other. Given this representation, finding related texts can be done by finding texts whose vector representations are the closest in this space to the vector representation of the text that the researcher finds relevant.

In order to perform this query efficiently, we use the open source Annoy datastore [17], which is designed for performing this type of nearest neighbor query in high-dimensional spaces. Every time a text or annotation is created, updated, or deleted, we synchronize its

vector representation with this datastore. When the web server receives a request to find texts or annotations similar to a specific candidate, it queries Annoy, and returns the results. Although there are other datastores that can perform high-dimensional nearest neighbor search, Annoy was chosen because of its simple interface and ease of use, as well as its highly-competitive performance [16].

While Annoy is highly performant for queries on a fixed data set, Spectacles is designed for use with active archives, whose contents – both documents and annotations – can be updated or added to or deleted over time. In general, high-dimensional nearest-neigbor vector stores do not support on-line updating, and are unsuitable for this usage. But unlike other data stores, Annoy supports memory-efficient background recalculation of the vectors used to answer its nearest neighbor queries. As documents and annotations are updated in Spectacles, daemon processes re-build the Annoy indexes in the background. On a fixed interval (10 seconds), the database service that allows the Spectacles web backend to query the Annoy vector database refreshes its index to the one most recently created by these daemons. In this way, even the newest annotations and documents can be shown as relevant recommendations no later than 10 seconds after being created.

B. Frontend

The frontend is the code that researchers use to interact with the archive, as shown in the Methods section above, and is responsible for drawing the user interface and sending queries to the backend web server. It is implemented in React, a popular open-source library for creating user interfaces in Javascript, HTML, and CSS. We chose React due to our familiarity with it as well as its popularity among the frontend engineering community, with the hope that this will allow other researchers to experiment with the Spectacles frontend and contribute back to our open source repository. React makes it easy to separate the frontend interface into different components, which we then combine to create the final interface. We made this choice because t allows for consistency in the design language of the interface as well as making it easier to rapidly iterate on its implementation.

The annotation editor was implemented by forking and modifying an existing open source project, Annotator.js [18]. This library was chosen as it was the basis of the annotation interface in Annotation Studio, the precursor to this project, and because it is the most full-featured web annotation library of the few that we evaluated. With another existing library [14],

we were able to easily implement the rich-text annotation editing interface. Our fork includes only modest modifications, largely to support our novel interface for displaying annotations in multiple parts of the document view at the same time.

V. Conclusion

Spectacles shows that it is possible to support humanities researchers in their speculative research tasks by applying existing tools and techniques from the world of software engineering. Given this, future successes might include collaborations with existing archive groups and institutes to bring their collections online using Spectacles so that researchers can take advantage of its unique capabilities. This would also allow us to further iterate on its user interface and capabilities in response to feedback from researchers actively using the software.

We consider Spectacles a success: it meets the goal of allowing anyone to create an active archive for speculative analysis from an existing collection of digital texts. It is free and open-source software, distributed under the MIT license, which means that anyone can use it for any purpose, as well as modify it to meet other purposes that the authors did not originally anticipate. Unlike previous work in the field, it is focused on supporting speculative analysis in active archives, and does so through a novel combination of tools: fuzzy-search, annotation, and recommendation. Together these techniques greatly surpass existing software tools and meet the needs of researchers working with large archives of texts.

References

[1] Drucker, Johanna. SpecLab: Digital Aesthetics And Projects In Speculative Computing. Chicago : University of Chicago Press, 2009. Print.

[2] http://hyperstudio.mit.edu/blog/introducing-the-active-archives-initiative-making-stories-within-the-archive/

[3] Palmer, Carol L. “Thematic Research Collections.” A Companion to Digital Humanities, ed. Susan Schreibman, Ray Siemens, John Unsworth. Oxford: Blackwell, 2004. http://www.digitalhumanities.org/companion/

[4] Drucker, Johanna. “Speculative Computing: Aesthetic Provocations in Humanities Computing.” A Companion to Digital Humanities, ed. Susan Schreibman, Ray Siemens, John Unsworth. Oxford: Blackwell, 2004. http://www.digitalhumanities.org/companion/

[5] T. H. Nelson. 1965. Complex information processing: a file structure for the complex, the changing and the indeterminate. In Proceedings of the 1965 20th national conference (ACM ’65), Lewis Winner (Ed.). ACM, New York, NY, USA, 84-100. DOI=http://dx.doi.org/10.1145/800197.806036

[6] http://hyperstudio.mit.edu/projects/annotation-studio/

[7] Neil Coffee, Jean-Pierre Koenig, Shakthi Poornima, Christopher W. Forstall, Roelant Ossewaarde, Sarah L. Jacobson; The Tesserae Project: intertextual analysis of Latin poetry, Literary and Linguistic Computing, Volume 28, Issue 2, 1 June 2013, Pages 221228, https://doi.org/10.1093/llc/fqs033

[8] Deerwester, Dumais, Furnas, Lanouauer, and Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science, 41 (1990), pp. 391-407.

[9] Sacha Zyto, David Karger, Mark Ackerman, and Sanjoy Mahajan. 2012. Successful classroom deployment of a social document annotation system. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 1883-1892. DOI: http://dx.doi.org/10.1145/2207676.2208326

[10] https://www.djangoproject.com/

[11] https://spacy.io/

[12] https://www.postgresql.org/

[13] Laermans, Rudi and Gielen, Pascal. "The archive of the digital an-archive." Image & Narrative: Online Magazine of the Visual Narrative. Issue 17: The Digital Archive., ed. Jan Baetens, Pascal Gielen, Rudi Laermans. 2007.

[14] https://github.com/danielcebrian/richText-annotator

[15] https://www.elastic.co/

[16] https://rare-technologies.com/performance-shootout-of-nearest-neighbours-querying/

[17] https://github.com/spotify/annoy

[18] http://annotatorjs.org/