Approaching Humanities Questions Using Slow Visual Search Interfaces

Adam J. Bradley, Victor Sawal, Christopher Collins

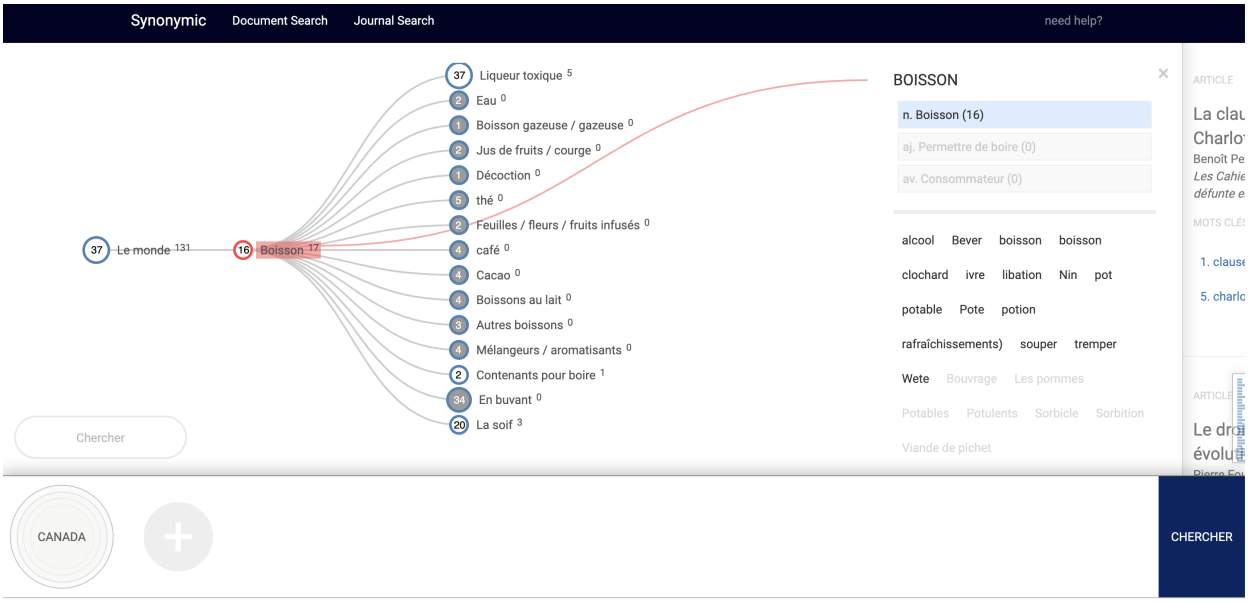

Fig. 1. The visual hierarchy search screen. Users can navigate the hierarchy of a large thesaurus to build evolving search queries.

Abstract—In this paper we discuss a visual search system that was designed and implemented for humanities scholars to ask questions of large document corpora. The system allows searches to be specified through traditional search, as well as through providing reference documents or visually exploring a semantic ontology for terms of interest. We offer a discussion of using visualization to try to answer humanities questions and take an honest look at the difficulties of using a methodology of one discipline to solve the problems of another.

Index Terms—visual search, visualization, digital humanities, visualization pipeline

1 Introduction

For digital humanists, the pursuit of visualization to help solve domain problems is an obvious one. It is a perfect entry point into technology for an interdisciplinary domain, such as DH, because it is a mixture of computing science, art, and design. There is something familiar about the pursuit of 'artistic science' that appeals to those who have spent a large portion of their life studying art and its implications. For humanities disciplines less focused on design, they study people, and visualization offers a new view on the data of those people. So, as good humanists who are trained in visualization we approach problems in the ways that we are taught: we define a problem, we then abstract that problem, organize our data, create a mapping, and design a visualization. Simple right? But what continually happens in this process is that the needs and wants of the humanities when it comes to system design are slightly different than scientists. Filters and views often feel like square pegs in round holes, and the time it takes to develop a visualization and implement it is often too extensive to keep up with how fast humanists iterate on their questions. In response to this we were challenged by our stakeholders, a group of humanities and social science researchers, to produce an interface that allows for asking quickly evolving questions of large document corpora. In our initial interviews the greatest challenge for researchers was not knowing the keywords to use to answer the questions they had of large document corpora; visualization was an obvious solution to this challenge.

We began by taking the usual steps of setting up stakeholder interviews, discussing their needs, gathering requirements, and starting to brainstorm about possible visualization designs. What quickly became apparent was that all of our designs, even the interactive visualizations, were too static to allow for the types of iterative questions that were being proposed. So what we tried to do was design from a humanist's perspective. We started asking questions of the process in the way that a philosopher would challenge the premise of a logical system. What this led us to was an idea that we thought interestingly confronted an existing paradigm; we sought to *complicate* the idea of the search bar as a way to deepen the engagement with the data, but ultimately found in practise that our approach did not completely solve the problem we set out to tackle. We feel that as visualization and digital humanities evolve in parallel, it is important to describe projects that attempted to answer DH questions and came up short. What we present is a fully

- *Adam. J. Bradley is with Ontario Tech University. E-mail: adam.bradley@uoit.ca.*
- *Victor Sawal is with Ontario Tech University. E-mail: victor.sawal@uoit.ca.*
- *Christopher Collins is with Ontario Tech University Research. E-mail: christopher.collins@uoit.ca.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

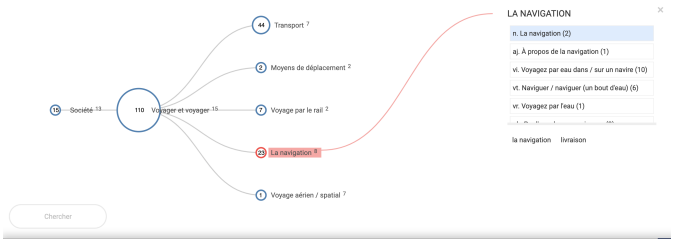

Fig. 2. The visual search interface. Users can navigate the hierarchy of a large thesaurus to build search queries with minimal knowledge of the search space. Numbers provide previews of document matches.

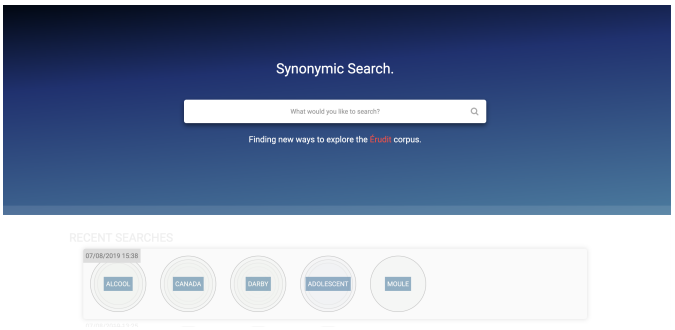

Fig. 3. The search box with a set of recent search terms. This list is kept for the previous ten searches to allow for quick iteration.

working system. At its most basic level it works well, but when trying to solve the problem of how to structure and ask humanities questions algorithmically, this project showed us we have a long way to go.

The single line search query has become so completely ubiquitous and integrated into our everyday lives that you would have to be mad to challenge its use. However, its very nature presumes the user has a well-formed query in mind. The question we had was: "If we integrate humanities data into the search itself can we iterate on the process?" We imagined a way to build queries using visual search elements that allowed for flexibility in refining the expression of information needs. There have been attempts at visual search interfaces in the past. From visual query systems [1,3], to semantic search [4], to database and XML search applications [5–8]. There has also been work done on visualizing systems [2, 10]. Even though the input modalities are different, perhaps the most similar in theory to our approach is the work on sketching database queries [9,12], but none of these solutions satisfied the need to be able to quickly manipulate the question itself, and most importantly, what if you don't know exactly what you are looking for? We built our visual search system as a way to explore these questions. The system consists of a standard input window where you can type in single line queries and search documents in the corpus using an Apache Solr backend. But to address the problem of not having a well-defined query string (a problem outlined to us by our stakeholders) we implemented a method of building queries from multiple sources and interactions.

2 System Design

The system is built to help scholars navigate the Érudit corpus, which is a database of scholarly articles and cultural journals totaling 259,267 documents. The corpus holds documents in both French and English.

The prototype is designed to work as a feedback loop. The current paradigm for search is the minimalist model found on most search engines that simply have an empty query bar that can be filled by a user with search terms and for the advanced user a minimal set of query commands such as 'AND', 'OR', and 'NOT'. To try and address our stakeholder's concerns of being able to *think through problems* using search and ask humanistic questions of large document corpora, it became clear from our initial interviews and our ongoing design discussions a different workflow was needed. The proposed flow allows users to build up queries using different input modalities and to track back to edit those queries until a useful question has been asked of the corpus. We accomplish this by using three different input methods (document upload, interactive visualization, and search query bar). Because of the available feedback loop, all parts of the system needed to be present for the user to interact with in any direction they choose. Figure 2 shows the visual interface. In actuality the screen can be shifted between the two main states by simply clicking on the area that is 'peeking' into the current active screen. This allows the user to traverse through the different stages of inputs and also allows them to use the each part of the system in whatever order they choose.

2.1 Query Builder

The query builder function allows any of the input modalities to contribute to the final search query. The user can at any time add search terms to the current query from any of the inputs, remove search terms, and save queries for later. As the user builds these queries the results screen is updated in real time and is always just a click away. This is our attempt to allow the user to define which connotations of words they want within their search.

2.2 Input

There are multiple ways to build search queries within the system. The first is the recognizable search query bar. The system can be used as a straight search engine where a user inputs a set of search terms and receives outputs based on those queries. But, we also allow two other input modalities that can be used in any order to build up a search query, Queries become evolving questions being asked of the corpus. The modalities are uploading documents and an interactive visualization that attempts to build metaphoric connections to the growing query string.

2.2.1 Uploading Documents

The first method of input was built to directly address one of our stakeholders' concerns. In our initial interviews with humanities researchers, one of the scholars expressed that they were having trouble finding similar documents to ones they already knew were relevant. We built the paper upload bar to allow scholars to upload documents that they knew they were interested in to find if there were related documents within the corpus (see Figure 4). After experimenting with several methods for modelling documents we concluded that term frequency inverse document frequency (TF-IDF) was the most appropriate. We modelled the entire corpus and used the TF-IDF vector of each document in the larger corpus in order to match incoming documents. When a user uploads a document, we pre-process the text to remove stop words, lemmatize, and then apply the TF-IDF algorithm. We then populate our query builder with the top three terms from pre-processing and we generate search results based on the best matches to the uploaded document (see Figure 3). Document upload can be done at any point in the query process. It could be done first to begin a search session, but could also be done last to add terms to an existing query.

2.2.2 Query Bar Keywords

The current search query can also be added to using simple keywords within the search query bar. Based on our discussions with stakeholders it became evident that an input method that users were familiar with was a necessary part of the system. It was expressed to us that in their work flows they sometimes simply know exactly what they are searching for and should have a way to input those search terms. When a query is put into the search bar and the 'search' button is clicked, the system both populates those terms into the query builder, but also provides results from the corpus. This allows the system to be used as a normal search engine, but also allows the user to enter into the feedback loop at any point. If after scanning the results page the user wants to update their query, they can use any or all of the input methods to do this.

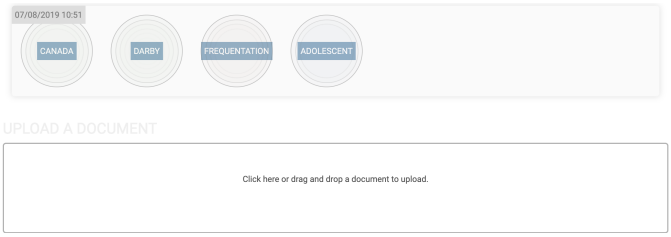

Fig. 4. Uploading documents to generate an interactive visualization. The current search query (circles on top) will be expanded by the document modeller when a user drags and drops a document into the upload bar (bottom).

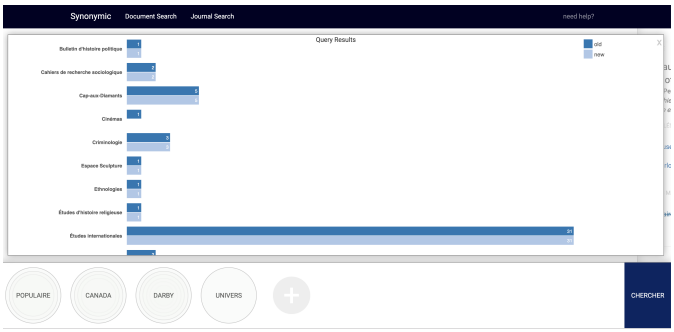

Fig. 5. The journal match screen updates as new searches are run, to show the user how the evolving queries are having an effect on output.

2.2.3 Interactive visualization

The problem that was set out to us by our stakeholders was "how do I search if I do not know the exact search terms that I want to use?" As inspiration for this design we took a well known phenomena from library science which is that library goers often find related books to the ones they are searching for by simply going into the books stacks with direction from an initial search. This phenomena has previously been explored in the visualization community [11]. This is a function of how libraries are organized, books about Shakespeare tend to be located on similar shelves and near one another. We have extended this idea to our query builder.

Our stakeholders told us that in their work they are often 'close' to knowing what they are searching for but cannot quite find the right search terms. To address this we used the Oxford Historical Thesaurus, which is a kind of ontology that sets up a hierarchy of relationships between specific senses of English words and is the largest thesaurus in the world. Most importantly, it organizes its hierarchy using semantic relationships, meaning we can allow the user to build search queries and interact with a visualization with minimal knowledge of a subject area. Also, this part of the interaction allows the users not only to explore the nature of their search terms, it also allows them to decide which versions of words to keep and which to discard.

2.2.4 The Oxford Historical Thesaurus

The Oxford Historical Thesaurus (OHT) covers all the words in the second edition of The Oxford English Dictionary, organized by semantics. The OHT assigns dates and types of usage to words from the Old English period to the present. It contains 800,000 meanings for 600,000 words, within 230,000 categories. Because our stakeholder's corpus contains documents in French and English, we used Google Translate to convert all of the hierarchical relationships within the OHT into French as well. The OHT is intended to be a record of the English language, but for our purposes we were leveraging it as an ontology that showed semantic relationships. We reasoned that problems of translation and any semantic differences between the two languages were more than made up for by the size of the data set.

The way the OHT organizes its hierarchy is by semantic concept, meaning that 'snow' is situated at a level above 'snowman'. When we translated the OHT into French, 'neige' still fell in the same place in the hierarchy above 'bonhomme de neige', and so did every other edge case we could think of to test. Accepting that the inherent problems of translation were going to be present we reasoned that this was a design trade off that was acceptable based on that the available ontologies in French, such as WordNet[1], were not comparable in scope to the OHT. When the user first comes to the system the visualization is set at the highest level of the hierarchy which has three choices 'The World', 'Society', and 'The Mind.' As an input modality the user could simply start searching through the interactive visualization and adding terms to the query builder. If they choose to start with a document upload or a search term, we then populate the visual search interface with data from the hierarchy that matches the query in the query builder most closely, as derived from the TF-IDF modelling of the uploaded document. We rank the terms in the query builder based on relevance and situate the user within the OHT at a point closest to their query.

In this way if one enters a search query looking for documents that are related to medieval weaponry, the visualization will zoom in to the nearest match in the hierarchy to that query. This allows the user to then enter into the feedback loop and 'look around' that query within the OHT for semantically related search terms. The current view will be populated with those terms that occur on the same level of the hierarchy as the given search term and the user is able to navigate to the parent and child nodes of those matches to explore words that are closely related to their generated search query. Each grouping in the hierarchy has a set of terms associated with it and we highlight which of those terms will produce matches within the existing corpus. The user can navigate anywhere they want around their search terms and add new terms from the OHT to the query builder in both English and French.

This solution directly addresses our stakeholders question about not knowing which search terms to look up. With the multiple inputs to our system we allow users to model a document they already know, add manual search terms to the query builder, and also explore an interactive visualization for semantically related concepts all while updating search results in real-time. The order of interaction is fluid and unconstrained. The visualization allows users to slow down and think through the data in a way that humanists do all the time. It is a method of analysis that supports serendipity and discovery, but can be completely omitted when the user simply needs quick results. The ability to enter and exit the process is paramount to its success. Users should not be *forced* to slow down, they should have the option to only engage and explore if it is needed. Our design of the three input modalities allows for these various workflows.

2.3 Results Screen

The results screen provides much of the information that one would expect from a document search (see Figure 6). Typical result fields such as paper title, journal, date, and authors are all included. We also provide the search query used to retrieve the results that has been built up in the query builder. This is an important part of the feedback loop. Because these queries can be built in any order, we found that when we tested the interface with our stakeholders that they were getting 'lost' in the process. The addition of the exact search terms which generate each result alleviated some of these concerns. The journal match screen keeps track of how the results across journals have changed with each updated query (Figure 5). It is important to note that there are trade-offs when implementing a system such as this. One of those trade-offs is that because the user can explore the corpus with any of the inputs at any time, there is a tendency to forget where you have been. We have addressed this by ordering the search terms in the query builder to act as a visual history of how that query has been built and also by allowing users to save their queries. But, it is important to note that part of the purpose of designing this process was to try to induce this kind of friction into the engagement. We leverage the ability to slow down the user to provide space to think while experiencing the system,

[1] http://globalwordnet.org/resources/wordnets-in-the-world/

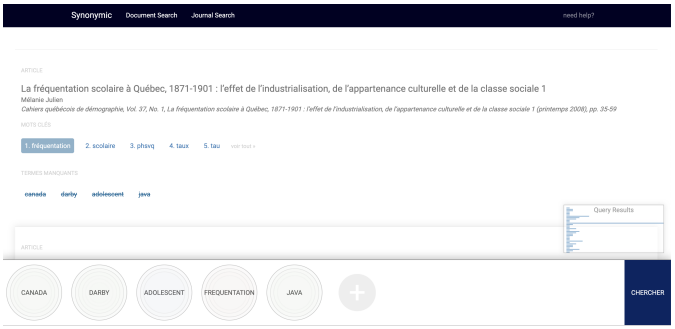

Fig. 6. The search results screen shows document matches, the current search query, and previous search results. Each document result shows which terms are present in the document.

but sacrifice some of the efficiency that we usually pursue in HCI and information visualization design. When asked, our stakeholders felt that for their particular problem this is an acceptable trade-off.

2.4 Iterative Design

Our initial interviews were conducted with five humanities scholars from three different universities who were all working together under the umbrella of a large interdisciplinary grant. Our team was tasked with building tools that could help humanities and social science scholars better navigate a large scholarly document repository in a way that better suited the work flows of humanities scholars. We spent an hour with each scholar and asked them questions regarding their existing work flows, their current projects, how they integrated technology into their work and daily life, and what, if any, problems they were currently having with technology. It became clear quite quickly that all five scholars wanted a way that they could ask questions of a corpora in a humanistic way. This is what lead us to the idea of visual search and what motivated our design for a system that has multiple input modalities that are interchangeable in ways that allow scholars to have different kinds of ideas and build queries from them. Our system integrates previous knowledge of the scholars, document matching, and exploration in a way that addresses this initial problem. We have developed an interactive search engine input that allows for the type of contemplation and discovery that these scholars engage in every day.

3 Stakeholder Final Evaluation

When we presented our final prototype, our stakeholders proposed to us that we implement the system in their commercial front end as a web-based search tool. We took this as a successful confirmation that our idea of a multi-modal search engine resonated with our primary stakeholders. Plans are underway to incorporate our model into their platform and has reiterated for us a need for different types of interaction that satisfy the work flows of specific domain user groups. We have attempted to challenge the overwhelming paradigm that faster is better and our stakeholders have confirmed that this kind of interaction design is useful for their work.

4 Discussion

In this project we set out to design a system that could address a specific search problem in the humanities using data visualization. While our stakeholders were happy with the results, we are actually skeptical about the real-world viability of our approach. In practise, the idea of multi-modal search feels disrupting in a way that will most likely have people resort back to input search queries. While slowing down the process and allowing for new ways to ask questions was our goal, we did not anticipate that at the end of this process, having accomplished what we set out to do, the process may actually be too slow. In future work, we plan to study this process and try to better understand the acceptable limits of the approach. For now, the project has satisfied the requirements of our stakeholders and feedback will be gathered from users after implementation and roll out. While it is out of the ordinary to perhaps take a skeptical stance of your own work, we thought that it is important while pursuing interdisciplinary problems, to allow for possible failure while exploring new design spaces. This pushed us to think about the accepted search paradigms and try to challenge how the work flow was accomplished for search. For hybrid teams working on interdisciplinary problems, it becomes increasingly important to take chances with design. This allows for the relationship between the humanities and the visualization community to grow. We have offered a glimpse into our evolving design process and have tried to describe a system that began with good intentions, and satisfied our stakeholders, but for us fell short of our own expectations. That does not mean the attempt was not valuable. We encourage all researchers working at these intersections to take similar chances; to try and break convention as true innovations often come from failed attempts at change.

5 Conclusion

As a design paradigm visual search has been dealt with in the past, but could also be explored in many new ways. We have demonstrated a single system that addresses a specific problem for humanities and social science scholars, but the approach could be used in different domains with varying and interesting results. Our future work will look to expand this design concept into other domains with stakeholders undergoing different kinds of work. We consider this project a proof of concept, but further implementation and evaluation are needed to understand the scope of the process including in the domain of information visualization, where visual analytics is well established.

References

[1] M. Angelaccio, T. Catarci, and G. Santucci. Qbd*: a graphical query language with recursion. *IEEE Trans. on Software Engineering*, 16(10):1150–1163, Oct. 1990. doi: 10.1109/32.60295

[2] D. Calcinelli and M. Mainguenaud. Cigales: A visual query language for a geographical information system: the user interface. *J. Vis. Lang. Comput.*, 5:113–132, June 1994. doi: 10.1006/jvlc.1994.1006

[3] T. Catarci, G. Santucci, and M. Angelaccio. Fundamental graphical primitives for visual query languages. *Inf. Syst.*, 18(2):75–98, Mar. 1993. doi: 10.1016/0306-4379(93)90006-M

[4] S.-F. Cheng, W. Chen, and H. Sundaram. Semantic visual templates: Linking visual features to semantics. In *Proc. Int. Conf. on Image Processing*, vol. 3, pp. 531–535, Oct 1998. doi: 10.1109/ICIP.1998.727321

[5] M. Derthick, J. Kolojejchick, and S. F. Roth. An interactive visual query environment for exploring data. In *Proc. ACM Symp. on User Interface Software and Technology*, pp. 189–198. ACM, 1997. doi: 10.1145/263407.263545

[6] S. Dogru, V. Rajan, K. Rieck, J. R. Slagle, B. Tjan, and Y. Wang. A graphical data flow language for retrieval, analysis, and visualization of a scientific database. *J. Vis. Lang. Comput.*, 7:247–265, Sept. 1996. doi: 10.1006/jvlc.1996.0014

[7] M. Erwig. Xing: A visual XML query language. *J. Vis. Lang. Comput.*, 14:5–45, July 2002. doi: 10.1016/S1045-926X(02)00074-5

[8] S. Fernandes, U. Schiel, and T. Catarci. Visual query operators for temporal databases. In *Proc. 4th Int. Workshop on Temporal Representation and Reasoning*, pp. 46–53, May 1997. doi: 10.1109/TIME.1997.600781

[9] D. J. L. Lee, J. D. Lee, T. Siddiqui, J. Kim, K. Karahalios, and A. G. Parameswaran. You can't always sketch what you want: Understanding sensemaking in visual query systems. *arXiv: Databases*, (1710.00763), 2017.

[10] S. P. Reiss. A visual query language for software visualization. In *Proc. IEEE Symp. on Human Centric Computing Languages and Environments*, pp. 80–82, 2002.

[11] A. Thudt, U. Hinrichs, and S. Carpendale. The bohemian bookshelf: Supporting serendipitous book discoveries through information visualization. *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, 2012:1461–1470, May 2012. doi: 10.1145/2207676.2208607

[12] M. Wattenberg. Sketching a graph to query a time-series database. In *CHI Extended Abstracts on Human Factors in Computing Systems*, pp. 381–382. ACM, 2001. doi: 10.1145/634067.634292