

1) Introduction

The major distinction between an information-retrieval search and a corpus query is that a search retrieves documents from a collection based on their relevance to the input pattern, whereas a corpus query retrieves word occurrences, and words themselves. Although subtle, this conceptual distinction, along with its corresponding technical requirements, is one of the major sources of difficulty in building a modern corpus query application.

The other major problem is one of design--web user interfaces have progressed by leaps in bounds in the last ten years, whereas corpus query applications have remained relatively static. The "faceted browsing" paradigm, in particular, has transformed many day-to-day search applications, such as on e-commerce and blog sites, and its advantages for usability have been established in numerous studies as well as real-world deployment.

I will argue that faceted browsing is a good solution to the usability challenge presented by powerful corpus query tools, and describe both the design of a faceted corpus query interface and the architecture of a corpus query web service that can support rapid filtering and aggregation.

2) Context

Faceted search, or faceted browsing, is typically defined as an iterative process, in which users begin with a simple query, and receive initial search results, along with a summary of the results in a tabular or hierarchical form, grouped according to orthogonal metadata "facets": a graphical or tabular summary of the search result that allows for interactive refinement of the query. This iterative process produces the same query, and the same results, as formulating a complex Boolean expression, but with substantial improvements in usability [Hearst, 2002]--there are no "wrong" values, and no empty result sets.

Initial work on this design technique focused on e-commerce sites and small image databases, aiming for collections of no more than a few hundreds of thousands of objects, i.e., orders of magnitude smaller than the scale of non-faceted search engines. Notable projects in the digital humanities and digital library field include the MIT SIMILE/Exhibit Project. [Mazzochi et al., 2005] Project MONK's bibliographic browser [Unsworth et al., 2009], and the NINES Collex interface, which built a faceted interface to an aggregated collection of nearly one million works [Nowviskie, 2005] -- demonstrating the necessity of a client-server architecture for effective scalability.

A corpus query engine, however, faces much steeper technical requirements than a bibliographic search--it has to efficiently search a collection of hundreds of millions to billions of word objects, and typically resists caching due to unpredictable access patterns. The scaling problem is compounded by a problem of complexity--digital texts are commonly purported to be usefully represented by an "ordered hierarchy of content objects", e.g., an XML tree containing many nested objects, each with its own relevant metadata [Renear et al., 1996]. Such structures are unsuited to modeling in common relational database systems, but algorithms to

search them efficiently do exist [Whaling, 2010].

The advantages of modeling these complex text objects are substantial as well, especially where the information-retrieval notions of “relevance” and “ranking” are unsuitable. Tree-oriented query languages, such as XQuery and the Annis Query Language [Zeldes et al., 2009], allow for precise specification of such patterns, whereas systems such as PhiloLogic simplify some of the Boolean logic, and instead display a metadata space with a dozen or more categories, from which ancestor-descendant relationships can be inferred implicitly [Olsen et al., 2004]. In either case, corpus query tools strive to use powerful query formulation to limit the size of the result set to a workable number of occurrences, such that a few hundred or so could be read in a single sitting. This paper describes a faceted corpus query interface that offers a similar level of power and accuracy, but is substantially easier to use than a specialized query language or complex search form.

3) Architecture

A typical concordance-style result display can only fit 10 or 20 search results at once, and a KWIC display is limited to around 50, whereas a tabular summary of a metadata distribution allows for an overview of thousands or tens of thousands of items at a glance. Displaying such frequency distributions in all these forms has been a feature of corpus query tools for decades; the major barrier to faceted interfaces is that full evaluation of a frequency distribution can take 20-30 seconds on a few hundred documents, increasing linearly with corpus size. This is quite a bit longer than a typical user is inclined to remain on a page, waiting for results.

We addressed this problem in the architecture of the server-side components of the corpus query system. First, by separating concordance requests from frequency distribution requests, we ensured that the trivial case of displaying a small number of results remained efficient. Second, we designed the frequency component to support partial evaluation with the same semantics as search paging; e.g., one could request a frequency distribution over only hits 501-1000 of a 5000 hit result set, and with a cluster of servers, one could evaluate many such requests simultaneously, since the query semantics are stateless (see Figure 1). Even in serial evaluation, this technique allows us to immediately display a partial frequency distribution to the user, which rapidly grows in coverage as more data are sampled. Finally, both the frequency and concordance requests use a common parameter set with a straightforward URI representation, which allowed us to generate the URI's for all state transitions on the server, and radically simplifying the implementation of the client; this is a straightforward application of Fielding's principle of “Hypertext as the Engine of Application State” [Fielding, 2000],[Fielding, 2008]. As a result, any HTTP-capable platform should be capable of hosting a similar faceted UI with little novel technical effort.

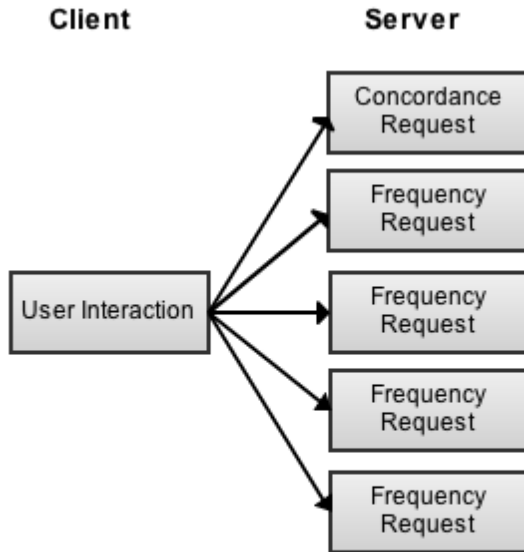


Figure 1: The client decomposes interactive use into a set of partial, stateless server requests.

4) Design

A recent example of the client concordance application can be seen in Figure 2. The current iteration of the facet display uses a simple three-column layout, with the search form on the left, concordance results in the center, and the facet/frequency display on the right. The major design constraint for the client is that the extant metadata categories will vary from collection to collection, and there may be dozens of them—for example, the Perseus collection has 15 different metadata fields on 3 different object types, and each category can have hundreds or thousands of distinct values. Thus, unlike SIMILE/Exhibit, the client can't display all metadata categories for simultaneous browsing. Instead, we augment the search form with a drop-down menu labeled as "group by:" containing all the metadata categories, as well as derived facets such as collocates or lemmas (word stems). This maps the faceted browsing idiom onto the sorting and filtering operations common in corpus query tools, and makes an explicit analogy from the facet display to a "summary" of the query results.

In the example workflow, a user begins with a query on the works of Shakespeare for the word "love", receiving 1752 results and a frequency distribution by title. He or she then selects the play *Othello, the Moor of Venice* and views the 62 results from it, grouped by character. Finally, the user selects the character Iago and retrieves all 26 occurrences of the word "love" spoken by Iago in *Othello*. The result of this interaction is equivalent to a sophisticated query pattern in ANNIS or PhiloLogic, but requires almost no prior knowledge of the contents of the corpora or the semantics of Boolean or hierarchical search.

query:

within word(s)

author:

title:

who:

group by:

running query for love @ loadtest with arg 1. displaying 1 - 50 of 1752 hits.

1. [William Shakespeare, The First Part of King Henry the Fourth Act 1](#) I answered indirectly, as I said;
And I beseech you, let not his report
Come current for an accusation
Betwixt my **love** and your high majesty.
Blunt
The circumstance considered, good my lord,
What e'er Lord Harry Percy then had said
2. [William Shakespeare, The First Part of King Henry the Fourth Act 1](#)
And think we think ourselves unsatisfied,
Till he hath found a time to pay us home:
And see already how he doth begin
To make us strangers to his looks of **love**.
Hotspur
He does, he does: we'll be revenged on him.

title	count
Two Gentlemen of Verona	131
Sonnets	124
As You Like It	102
Romeo and Juliet	99
A Midsummer Night's Dream	82
Much Ado About Nothing	81
Love's Labour's Lost	79
Othello, the Moor of Venice	62
The Tragedy of King Richard the Third	55

Search for "love"
group by play title:
1752 hits.

query:

within word(s)

author:

title:

who:

group by:

running query for love @ loadtest with arg 1. displaying 1 - 50 of 62 hits.

1. [William Shakespeare, Othello, the Moor of Venice Act 1](#) And not by old gradation, where each second
Stood heir to the first. Now, sir, be judge yourself,
Whether I in any just term am affianed
To **love** the Moor.
Roderigo
I would not follow him then.
Iago
O, sir, content you;
2. [William Shakespeare, Othello, the Moor of Venice Act 1](#) It is as sure as you are Roderigo,
Were I the Moor, I would not be Iago:
In following him, I follow but myself;
Heaven is my judge, not I for **love** and duty,
But seeming so, for my peculiar end:

who	count
Iago	26
Othello	17
Desdemona	11
Cassio	5
Bianca	1
Brabantio	1
Emilia	1

Select play "Othello",
group by TEI @who
(speaker) : 62 hits

query:

within word(s)

author:

title:

who:

group by:

running query for love @ loadtest with arg 1. displaying 1 - 26 of 26 hits.

1. [William Shakespeare, Othello, the Moor of Venice Act 1](#) And not by old gradation, where each second
Stood heir to the first. Now, sir, be judge yourself,
Whether I in any just term am affianed
To **love** the Moor.
Roderigo
I would not follow him then.
Iago
O, sir, content you;
2. [William Shakespeare, Othello, the Moor of Venice Act 1](#) It is as sure as you are Roderigo,
Were I the Moor, I would not be Iago:
In following him, I follow but myself;
Heaven is my judge, not I for **love** and duty,
But seeming so, for my peculiar end:

who	count
Iago	26
Othello	17
Desdemona	11
Cassio	5
Bianca	1
Brabantio	1
Emilia	1

Select speaker "Iago":
26 hits

Figure 2: Example - Workflow

The major downside of this design is that it has relatively few capabilities for filtering on multiple

categories simultaneously -- in practice, we find that selecting an individual facet is a sufficient filter for small-to-mid-size databases--but adding check-box-selection for iterative filtering would greatly improve functionality for large corpora. The primary barrier to this has been the difficulty of displaying multiple facets simultaneously, which we expect will require further experimentation with tabs, breadcrumbs, hierarchical facets, and other possible UI features.

5) Conclusion

This paper has described the design of a faceted concordance system in the REST paradigm, and the architecture of a web service framework to support such a system. The combination of concordances and frequency distributions goes far beyond faceted browsing, forming a primitive data source for any number of interfaces--an efficient corpus query engine can provide the framework for histograms, timelines, comparative queries, clustering, and many other applications. The promise of the digital humanities, as a field, lies in the willingness to treat textual objects in their full complexity, without making compromises for the sake of convenience. Having demonstrated the computational tractability of these objects, we can advance application and interface design even as we scale toward truly massive corpora.

References:

Fielding, Roy, Architectural Styles and the Design of Network-based Software Architectures, PhD dissertation, UC Irvine, 2000.

Fielding, Roy, REST API's Must Be Hypertext-Driven, <http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>, modified Monday Oct. 20 2008.

Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., and Yee, P., Finding the Flow in Web Site Search, in Communications of the ACM, 45 (9), September 2002, 42-49.

Mazzocchi, Stefano, Garland, Stephen, and Lee, Ryan, SIMILE: Practical Metadata for the Semantic Web, <http://www.xml.com/pub/a/2005/01/26/simile.html>, modified January 26 2005.

Nowvickie, Bethany, COLLEX: Semantic Collections & Exhibits for the Remixable Web, <http://www.nines.org/about/Nowvickie-Collex.pdf>, modified January 2005.

Olsen, Mark, with Andreev, Leonid, Montoya, Orion, and Voyer, Robert, An Open Source Implementation of PhiloLogic for Large TEI-Lite Document Collections, ALLC/ACH 2004 Conference, Gotenberg, Sweden.

Renear, Ilen, Mylonas, Elli, and Durand, David, Refining our Notion of What Text Really Is: The Problem of Overlapping Hierarchies, in Research in Humanities Computing, edited by N. Ide and S. Hockey, Oxford University Press, 1996.

Unsworth, John and Mueller, Martin, The MONK Project Final Report", <http://www.monkproject.org/MONKProjectFinalReport.pdf>, modified September 2, 2009.

Whaling, Richard, Humanities Computing at Scale: Engineering Digital Collections, presented at the Computation Institute, University of Chicago, 2010, slides available at http://docs.google.com/present/view?id=dhdrzp66_52dsj2zmqq.

Zeldes, Amir, Ritz, Julia, Lüdeling, Anke and Chiarcos, Christian, ANNIS: A Search Tool for Multi-Layer Annotated Corpora, in Proceedings of Corpus Linguistics 2009, July 20-23, Liverpool, UK.