

Tools for Thinking, Thinking Tools

The Semantic Engine and the Literary Analysis Toolkit

Aaron Coburn, John L. Cuadrado, Gabriel Schine
Middlebury College

Abstract. The growth in the volume of digital text is outpacing our ability to organize and explore it in meaningful ways. This paper describes a project that develops computer models of large digital text collections in such a way that patterns begin to emerge, reflecting the conceptual relationships among documents. Using a graph-theoretic model of the text, this project has developed a variety of applications for making sense of an otherwise overwhelmingly large amount of data.

Promising results have been found in several domains. Foremost among these are expanded recall search tools: search engines that will return relevant documents that may not even contain the original keyword – all without relying on metadata. In addition, this paper describes techniques that cluster documents based on content similarity and visually displays them. Finally, these tools are used in generating visualizations of literary texts, particularly the patterns of interaction among characters.

It is becoming increasingly difficult for Humanities Scholars to navigate the explosive growth of publicly available text, both online and in electronic form, and current search engines have not been able to keep pace. Furthermore, while personal computers themselves are highly effective for organizing and storing documents, we contend that this ability to fill increasingly large hard drives has not helped scholars think more clearly or to better understand the data that is accessible.

This paper describes a project that models large document collections in such a way that one can find not only documents that contain a particular key word or phrase -- the typical domain of full-text searches -- but also relevant documents that do not contain any instances of the search phrase. This project does not rely on any metadata -- which, while highly effective for categorization, can also be prohibitively time consuming or expensive to create. Instead it uses techniques and algorithms from Statistical Natural Language Processing and Graph Theory to identify the latent patterns and connections among documents, found by analyzing word frequency and distribution statistics. The project is based at Middlebury College and funded by a grant from the Andrew W. Mellon Foundation.

By noticing word co-occurrence patterns across large data collections, a computer can make inferences about semantic relationships among documents. These word frequency statistics can also help organize texts into conceptually meaningful categories. This means that a user can enter a query that returns relevant results that do not necessarily contain any of the original keywords. For example, one might search an historical

database for 'Napoleon' and find documents that contain the word 'Bonaparte', even if 'Napoleon' does not appear.

This example only hints at the possibilities with this sort of tool. Through a combination of Natural Language Processing and graphical visualizations, these tools not only free the researcher to spend more time thinking, but with them, it also begins to appear that the computer itself is engaged in thinking.

Text Modeling

The core of this project lies in a graph-based representation of a text corpus. The graph is bipartite – composed of two types of nodes: document nodes and term nodes. Each document node is connected to term nodes by edges of varying weights. The term nodes are selected from the text in an entirely automated process, selecting meaningful words and phrases that occur neither too often nor too seldom¹. Typically, the term list is made up of nouns and noun phrases, but this depends on the collection and the type of analysis being conducted. Other possibilities are to use primarily verbs or adjectives to show action rather than concept.²

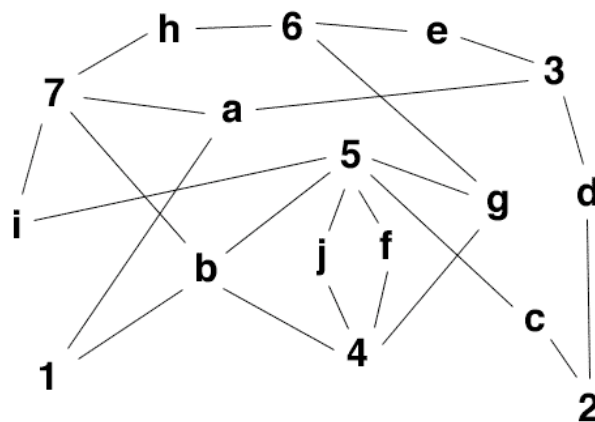


Figure 1: Connected bipartite graph consisting of terms (represented as letters) and documents (represented as numbers)

In the resultant graph, complexity of the structure is mostly hidden from the user, who will tend to interact with smaller subsections of the graph at any one time. Documents will be one step away from terms, two steps away from other documents and four steps away from still more documents. Because edges will be weighted differently, according to how frequently a term occurs in a document and the collection as a whole, one can begin to have the computer calculate measures of similarity or difference between documents³.

This structure, as a mathematical abstraction, can now be used in many different applications and scholarly domains. This project has even had some initial success in non-textual domains such as bioinformatics.⁴ Here, the authors describe two particular applications of this method of text modeling: information retrieval and data visualization.

Expanded Recall Search

The two customary approaches used in the field of Information Retrieval include a structured search: one that relies on metadata, and an unstructured search: a full-text search that uses little to no metadata. Ours is an unstructured approach masquerading as the other.

A fine example of the structured search can be found in the electronic card catalog at any university library. Each published book is assigned subject headings and other metadata (e.g. title, author, date of publication) by expert catalogers from a controlled vocabulary that is constantly being revised. The great advantage of this is that a search for “France – History, 1789 - 1815” will find every item in the catalog that is significantly about that period.

Result Page [Prev](#) [1](#) [2](#) [3](#) [4](#) **5** [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) ... [36](#) [Next](#)

Num	Save	SUBJECTS (49-60 of 428)	Year	Entries 2799 Found
49	<input type="checkbox"/>	France -- History -- 1789-		13
50	<input type="checkbox"/>	France -- History -- 1789-1793.		14
51	<input type="checkbox"/>	France -- History -- 1789-1793 -- Influence.	c1983	1
52	<input type="checkbox"/>	France -- History -- 1789-1793 -- Personal narratives.		2
53		France History 1789-1815 -- See Also the narrower term French Expedition to Ireland, 1796-1797		1
54	<input type="checkbox"/>	France -- History -- 1789-1815.		33
55	<input type="checkbox"/>	France -- History -- 1789-1815 -- Congresses.	c2002	1
56	<input type="checkbox"/>	France -- History -- 1789-1815 -- Historiography.		2
57	<input type="checkbox"/>	France -- History -- 1789-1815 -- Influence.	1970	1
58	<input type="checkbox"/>	France -- History -- 1789-1815 -- Religious aspects -- Pictorial works.	1989	1
59	<input type="checkbox"/>	France -- History -- 1789-1900.		20
60	<input type="checkbox"/>	France -- History -- 1789-1900 -- Art.	c2002	1

Figure 2: Search results for “France – History” using a university library catalog based on Library of Congress subject headings

One of the most familiar instances of an unstructured search is Google, which, rather than functioning as an electronic card catalog, resembles a concordance. It provides a full-text search of the public web, and if the search terms exist together in a document, it will be found. Here, however, the result set can be overwhelmed by false positives and an exclusion of synonymous terms.

A full-text search is, in many cases, indispensable, but it is rare that, with a single key word or phrase, one can find all the relevant documents related to a particular query. Usually the results will be too general or too specific. In this context, our project attempts to index text collections in such a way that the search engine will not only return the documents that contain particular keywords but also make suggestions about other related

documents and similar concepts. And it does this with little to no human intervention (e.g. concept lists, thesauri, metadata).

Using the data model described earlier, a search engine can be constructed that traverses the graph using a method known as spreading activation.⁵ One or more nodes provide the starting point for the “initialization energy”, which spreads outward, dissipating according to the weights assigned to each edge. Starting from an initial term, the greatest amount of energy will spread to those documents containing the original term. The energy continues to spread to additional documents until it reaches a minimum threshold. Relevance is reflected in the amount of energy a node receives in this process. Not only relevant documents, but also related terms are found and can be displayed in the interface to help the user to orient to the semantic context of the result set.

There are several methods for conducting these traversals, from using random numbers to find approximations of relevance to using a comprehensive depth-first search. For large data collections it is significantly faster to use random numbers

Similar documents can be found by starting this process on a document node. Any combination or number of term and document nodes can be used in a search, allowing for greater and greater granularity in the process of navigating the collection.

Document Clustering

Clustering documents into meaningful categories can be helpful while searching a collection, but it can also be used on its own, to allow the user to view the structure of a collection. This is particularly useful when beginning to study an unfamiliar set of data or to organize a large amount of familiar information. By grouping terms and documents into logical clusters, it also provides a useful way to identify the distribution of concepts and the relationships among documents in a collection.

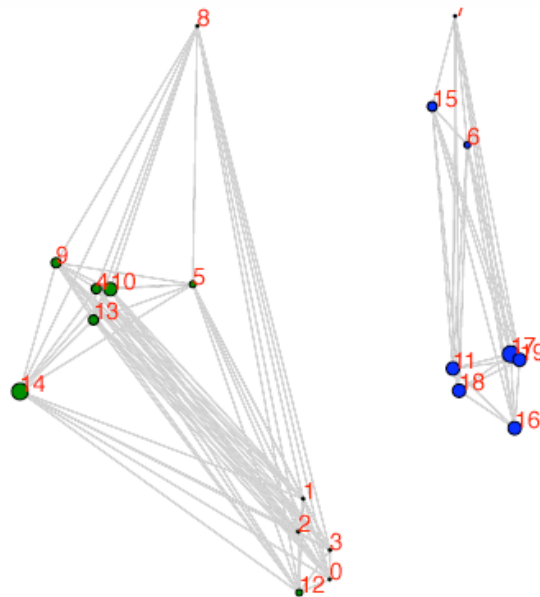


Figure 3: Document nodes divided into two distinct clusters

These clusters are identified by using the data found in the graph structure. Using edge weights and node degrees, a computer creates a matrix describing the relative connectedness between nodes and therefore the degree of similarity. These calculations, when applied to the entire graph structure, will identify patterns and relationships that may be missed while navigating through large datasets. As clusters, these relationships can be displayed visually as in Figure 3.

Limitations

This approach to textual modeling shows promise, particularly with large text collections for which no metadata exists. There are some notable limitations to this, however. Firstly, in small document collections, the computer's results begin more to resemble dubious guesses than logical inferences. Collections with fewer than 2000 documents may not contain enough data to derive meaningful word co-occurrence statistics. As collections grow, the accuracy of these techniques will only improve.

Furthermore, if the document length in the collection varies widely, especially in a small collection, there are occasionally unexpected results. Similarly, if the content of individual documents is very unfocused, it is difficult to identify the principle themes of a text, and therefore its relationship to other documents is less clear. Extremely short documents (i.e. less than three paragraphs) also make it difficult to categorize a text.

Without some sort of supplementary translation software, multi-lingual collections are problematic for obvious reasons. Inconsistent spelling, extensive dialogue and stylistic flourishes also make it difficult to apply these tools to text. Prime examples of this include weblog posts and literature.

Literary Toolkit

These limitations are not insurmountable, but applying these tools to such domains as literature requires some changes in how the graph model works. For instance, a literary text must be approached sequentially, chapter-by-chapter, rather than as an unorganized collection of documents. There are difficulties with metaphors, anaphora, dialect and other aesthetic characteristics; nevertheless, there are interesting visual analyses that can be generated computationally.

The literary analysis toolkit began as a project to create an electronic text reader with some Natural Language Processing tools built into it. It allows a user to read, annotate and search a text. It also extracts statistically significant words and phrases that can help readers who are new to the material begin to orient themselves to the content.⁶

Of greater interest, though, are the visualizations of the texts. Rather than indexing the entire content of a novel, only the interactions among characters can be extracted. Interaction, here, is defined as having characters mentioned by name in the same segment of text – either a paragraph or a window of several adjoining paragraphs.

In Jane Austen's *Emma*, one can quickly navigate the changing constellations of character interaction to find relevant passages. It is also worth noting that seemingly accurate graphs were drawn in an entirely automated process: the ambiguity of Emma's matchmaking is captured, while toward the end of the novel, those characters that marry are shown in three neat pairs.

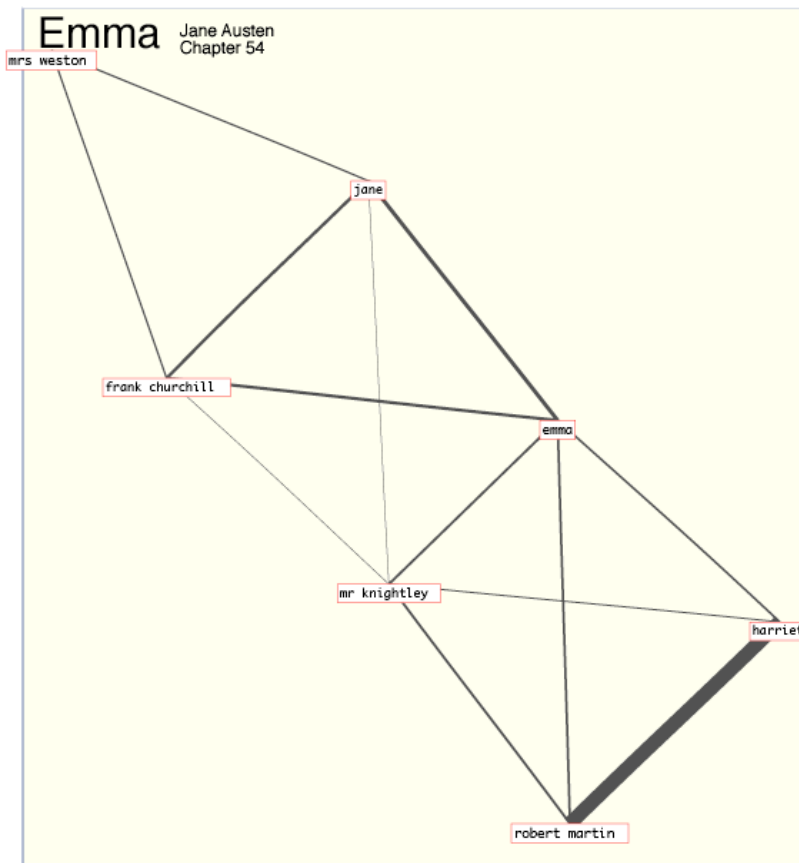


Figure 4: Visualization of chapter 54 in Jane Austen's *Emma*.

Austen's works are, however, short enough that they are easily read and re-read in a short span of time, making such a tool less useful to someone very familiar with the text. Larger texts, though, show some more promise. Samuel Richardson's *Clarissa*, for instance, is one of the longest novels written in English. The first edition was published in 1747 and 1748 in seven volumes, with a revised second edition published the following year. The third edition of the text was published two years later than that, in 1751, in eight volumes, while a fourth edition was published in 1759.

With this novel we generated a series of character visualizations for each letter contained in the first and third editions.⁷ Many of the graphs are identical, some are slightly different and some represent complete revisions. Patterns begin to emerge, providing a birds-eye view of the text, which can allow the reader to more quickly focus attention or analysis on certain aspects of the text.

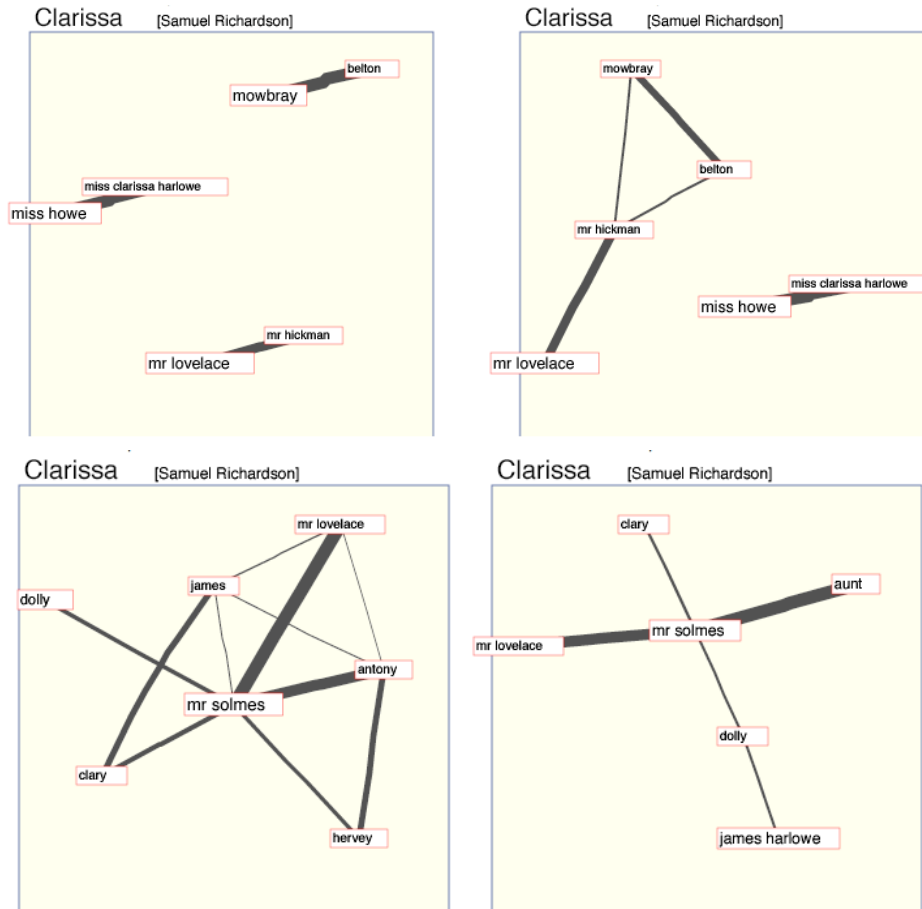


Figure 5: Graphs comparing the first and third edition of *Clarissa*, Volume 2, Letters III (top) and XXXIII and XXXI (bottom).

The Literary Analysis tool has been used successfully with Spanish, Russian and Chinese novels.⁸ As greater quantities of text for a language become available, the accuracy of the tool's ability to extract meaningful data improves.

Conclusion

While computers are able to store greater quantities of data, techniques from Natural Language Processing are allowing scholars not only greater access to the data, but also better tools for navigating the content of the information. As these tools find and display the patterns corresponding to the latent word co-occurrence statistics, their findings begin to resemble a type of initial thinking about a text. This "thinking" may be somewhat rudimentary, but it is allowing researchers and students to engage in even more effective and comprehensive analysis of large amounts of text.

¹ Term selection typically eliminates words and phrases occurring in fewer than three documents and terms that occur in greater than 10% of the collection documents. This range varies according to collection, but it has shown to be an effective way to identify document similarity.

² Nouns and noun phrases are identified first by using a part-of-speech tagger (available at <http://search.cpan.org/author/ACOBURN>), and then with pattern-matching techniques described by Bader, R., Callahan, M., Grim, D., Krause, J. and Pottenger, W., "The role of the HDDI™ collection builder in hierarchical distributed dynamic indexing", *Workshop on Text Mining*, Chicago, 2001, pp. 23-30.

³ No single approach for determining edge weight works best under all circumstances. Typically, however, edge weight is derived from term frequency and inverse document frequency statistics.

⁴ Ceglowski, M. and Cuadrado, J. Slide presentation, O'Reilly Bioinformatics Conference, February 2003. http://conferences.oreillynet.com/cs/bio2003/view/e_sess/3406

⁵ Preese, Scott. *A Spreading Activation Model for Information Retrieval*. PhD. Thesis, University of Illinois, 1981.

⁶ The Literary Analysis Toolkit is available at <http://literary.knowledgesearch.org>.

⁷ The complete list of visualizations are available at <http://literary.knowledgesearch.org/richardson/clarissa>

⁸ Novels that have been used in this tool include *Don Quijote* by Miguel de Cervantes, *The Master and Margarita* by Mikhail Bulgakov, and *The Dream of the Red Chamber* by Cao Xueqin.