Text Modeling and Visualization with Network Graphs

Aaron Coburn *(acoburn@middlebury.edu)*

National Institute for Technology and Liberal Education (Middlebury College)

As greater quantities of textual source material become available to Humanities scholars, it becomes ever more challenging to retrieve, explore and manage these text collections (Lyman). Large, heterogeneous collections present further difficulties in attempts to represent the text visually. Metadata, particularly XML, is highly effective in adding structure to these collections and therefore aids in the process of locating and visually rendering the relationship between documents. Nevertheless, the process of creating high-quality metadata is both time-consuming and expensive, and this luxury is not always available.

This presentation will describe a project that is investigating and implementing statistical and graph-theoretic techniques for identifying, classifying and representing the content of large document collections in the absence of metadata. Models of the collection can be displayed to show various types of relationships between documents and their content, including term-based concept maps across a set of texts, document similarity measures or visualizations of the interaction among the characters in a novel. Furthermore, this model allows users to query the collection, in most cases with better recall and precision than a full-text keyword search.

Constructing a Network Graph

The basis of these investigations is in the representation of a document collection as a graph of interconnected nodes. Each node corresponds to either a document or a term appearing in the collection, and nodes are connected by edges according to the frequency of their co-occurrence. The list of term nodes is typically filtered to include only those grammatical constructs, such as nouns and noun phrases, which tend to carry more semantic information about the content of the document. Documents nodes connect only to term nodes and term nodes connect, likewise, only to document nodes. The connecting edges are weighted by several factors, including term frequency and a normalization for document size. Depending on the analysis being conducted, term nodes can represent single words, phrases, character names, locations, stylistic data or any combination thereof. Likewise, document nodes may represent arbitrarily sized text blocks, from sentences to paragraphs to entire book-length works.

The index derived from the text collection is similar to the term-document matrix of a vector model, but it is interpreted differently. For instance, when the collection is being explored with natural language queries, it proceeds with a technique called spreading activation (Preese). Each term in the graph that appears in the search statement is initialized with an amount of activation energy. This value is dispersed along each edge according to the established weight of each connection, and additional nodes are activated with the remaining energy. This process repeats itself along the graph until the initial amount of energy has been completely dispersed. Those nodes with the highest energy levels at the end of this process are considered most relevant to the query, and the results can be sorted accordingly. The document nodes become the corresponding result set while the activated term list can be used as relevance feedback to the user: a guide to the semantic composition of the result set (Search::ContextGraph).

This process will not only find documents containing the terms in the query string, but also relevant documents in which the search terms do not appear. In full-text keyword searches, highly relevant documents that do not contain any of the search terms are necessarily excluded. With a graph-based technique, however, documents with many shared terms are considered more closely related, and therefore, the process of traversing the graph will mark those documents to be relevant even if they do not contain any of the original search terms. In this way, the query *arctic* would likely also return documents that contain words such as *polar*, *north*, *ice* and *tundra*, but which happen not to contain the term *arctic*. Relevance is determined based on the overall measure of connectedness between nodes, rather than whether a term exists in a document.

Visualization

The graph-based model of the text collection is also useful in creating visualizations of document relationships. This requires, first, the creation of a distance matrix representing the degree of similarity between nodes, and second, the scaling of the matrix to two dimensions. The distance matrix is calculated by conducting traversals of the graph networks, either with a breadth-first algorithm or (in order to save processing time) a random walk of the graph. Then, in order to scale the graph structure to display on a screen, the system identifies smaller clusters of nodes, each of which is mapped to a reduced dimensional space. This technique of locally linear embedding tends to preserve the general structure of both local and global clusters of nodes from the original graph (Saul).

When the network of terms and documents are clustered in a reduced dimensional space, they can be rendered in a browser window, showing the relative similarity among nodes. This particular system will output a vector graphic map (SVG), and a user can both view and interact with the graph.

This technique has proved useful in several experiments to display the relationship and interaction among characters in literary texts. First, proper names are extracted with a part-of-speech tagger (Lingua::EN::Tagger); in the absence of metadata, variant forms can be combined during an interactive step. Term nodes, in this case, represent character names, while document nodes consist of either individual paragraphs or a window of adjacent paragraphs: the number of paragraphs in each 'window' typically ranges from one to three. Interaction between characters is based on the frequency in which individuals are named, and the strength of that connection is reflected in both the proximity of the respective terms and the strength of the connecting edge. In the attached example from *Master and Margarita* by Mikhail Bulgakov, the characters with greater amounts of interaction are clustered accordingly (see Figure 1). This type of visualization can also allow users not only to view the relationship among characters in a text, but also to provide clickable access to the very interactions to which the model refers. Resolving anaphora, however, remains a major challenge in assessing a truly accurate measure of character interaction.

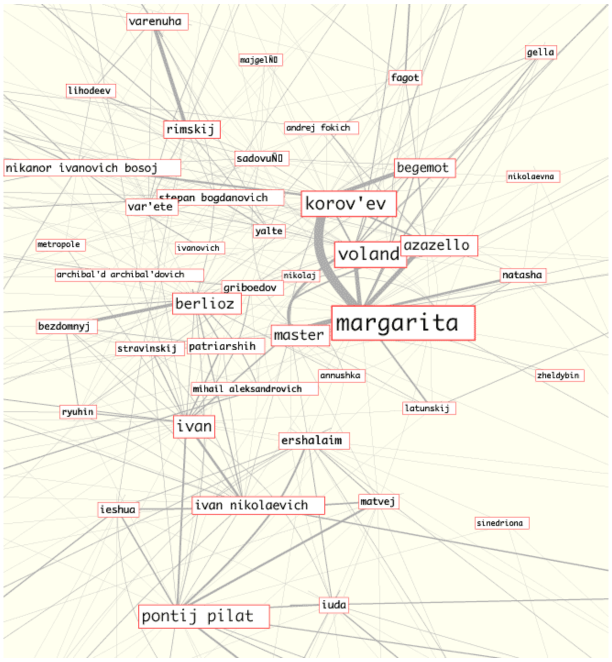

Figure 1

These models of text collections show promise in representing both content and stylistic similarity among texts. Furthermore, the ability to quickly and accurately search a collection of texts shows many advantages over a full-text search. In the absence of metadata, this approach may provide a useful first step in navigating and managing any sufficiently large text collection.

Bibliography

Bulgakov, Mikhail. *Master and Margarita.* n.p.: Penguin Classics, 2001.

Lingua::EN::Tagger (Perl Module). Accessed 2005-04-07. <http://search.cpan.org/~acoburn/Lingua-EN-Tagger-0.11/>

Lyman, Peter, and Hal R. Varian. *How Much Information.* SIMS, University of California, Berkeley. Accessed 2005-04-07. <http://www.sims.berkeley.edu/how-much-info-2003>

Preese, Scott. *A Spreading Activation Model for Information Retrieval.* Ph.D. thesis, University of Illinois, 1981.

Saul, Lawrence, and Sam Roweis. *An Introduction to Locally Linear Embedding.* Accessed 2004-06-30. <http://www.cs.toronto.edu/~roweis/lle/papers/lleintroa4.pdf>

Search::ContextGraph (Perl Module). Accessed 2005-04-07. <http://search.cpan.org/~mceglows/Search-ContextGraph-0.15/>