Improving Interactive Information Retrieval Effectiveness with 3-D Graphics

Russell C. Swan
Computer Science Department
University of Massachusetts
Amherst, MA 01003
swan@cs.umass.edu

James Allan
Computer Science Department
University of Massachusetts
Amherst, MA 01003
allan@cs.umass.edu

ABSTRACT

Recall-oriented information retrieval requires locating as many documents as possible that are relevant to a query. Traditional information retrieval systems present results only in a ranked list. Inspired by the cluster hypothesis, we present a novel 3-D visualization tool which aids recall-oriented retrieval. The visualization portrays clustered documents and concepts using a modified spring embedder. A small user study suggests this approach has merit.

Keywords

Information Retrieval, Visualization, 3D Interfaces

INTRODUCTION

Most users of Information Retrieval (IR) systems are searching a collection of documents (or web pages) for a small number of relevant texts. It is not important to them whether there are ten, 100, or even 1000 related documents, provided the retrieval system allows one or two to be found quickly. For this purpose, an interface is sufficient if it accepts a query and presents the user with a list of relevant texts--preferably ranked in order of their likelihood of being relevant.

We are interested, however, in a style of interaction called *recall-oriented retrieval*. In this case, a user hopes to find all relevant material (or at least most of it). Using the typical interface, the user poses a query and selects relevant texts from the resulting list. Then he or she tries another query and reads the new texts. The cycle continues until no new documents appear that seem worthwhile. This technique is common in the legal domain, where prior cases serve as precedence and failing to find precedence could destroy a case. Other examples include scanning medical case reports for related symptoms, obtaining material for compiling a summary or briefing on a topic, or doing an exhaustive search for the work by a particular research group.

We believe that typical IR interfaces (i.e., query and ranked list) can be greatly improved for this task by providing the user with alternate views of the relationships between retrieved documents and of the relationships between their contained concepts. This work describes an exciting new visualization and user interface developed to test our hypotheses. The next section describes the Cluster Hypothesis, an idea which motivates the visualization. It is followed by a detailed description of the interface. Next we present the results of an informal user study which suggests that our technique works for the purposes intended--and intriguingly, that experienced users will be both better equipped to understand and less inclined to use the visualization! Finally, we present our conclusions and plans for future work.

CLUSTER HYPOTHESIS

The Cluster Hypothesis states that "closely associated documents tend to be relevant to the same requests".[13] It has been shown that the Cluster Hypothesis is not strictly true on a corpus wide basis, but it does hold true on a set of retrieved documents.[2,6] That is, although similar documents within a collection may not be relevant to the same query, documents that are similar and retrieved in response to the same query are likely to share relevance. This implies that having judged some members of a set of retrieved documents provides some evidence to the user of the likely value of examining in more detail some of the unjudged documents., i.e., those that are “near” the already judged texts.

Clustering was originally proposed and studied partly as a method to reduce processing time when computing time was an expensive resource. Corpora were clustered, then indexed, and a query was compared against cluster representatives to find the best candidate cluster. After the best cluster was found either the entire cluster was returned or the query was compared to documents within that cluster and the best matches from that were returned. Although that method is time efficient it degrades on recall compared to non clustered methods. ("Recall" is the proportion of known relevant documents actually retrieved by a system.) However, a lot of the cluster research is still applicable. Information spaces tend to be clumpy, and returned documents tend to naturally aggregate into distinct groups.[6,13] Fast clustering algorithms have also been applied to collections as an aid to browsing.[6,10]

Given that the Cluster Hypothesis holds for sets of retrieved documents, we investigated an IR system that displayed the nearness of documents to each other for a collection of retrieved documents. If such a system could be designed in a way that the clusters were obvious and the nearness and degree of membership in a cluster could be easily and quickly grasped by the user, then the user's performance on retrieval tasks should be improved.

In this study, we present visualizations of document clusters and of concept clusters. The former are created by converting each document into a vector in t-dimensional space, where t is the total number of unique indexing units in the document collection. The 1987 Wall Street Journal subcollection from the TREC collection [5] contains 46448 documents (130 Mb) and has 76822 unique terms. Vectors are then normalized so that they have unit length. The similarity between two documents is calculated by taking the inner product of their vectors which corresponds to finding the cosine of the angle between the vectors in t-space: identical vectors have a similarity of 1.0, and entirely orthogonal vectors have a similarity of zero. (This form of similarity computation is common in vector space models of information retrieval.[12])

For cluster visualization, we want a distance between objects that is inversely related to the similarity. We calculated the distances by determining the sine of the angle between the vectors. At this point, we have 20-100 documents with known distances between each of them. Since it requires up to n-1 dimensions to display n objects and their relationships accurately, we chose to use spring embedding to present the documents and their relationships.

Understanding distances displayed in this way then becomes a perceptual rather than an intellectual process. There has been much recent work on graph drawing and graph placement from both a computational and aesthetic viewpoint.[4,8] One of the techniques developed has been that of the spring embedder. Since the 3 dimensions these objects will be placed along are sufficiently collapsed from the 10^5 dimensions the objects reside in the 3 physical dimensions will not correspond to anything meaningful, so absolute placement will not matter. Since the idea is to present a general sense of nearness, and the eye is good at discerning large quantities of rough information quickly but is lacking in precision, the errors introduced by a spring embedder should not matter for our purposes. Spring embedders have been developed for embedding objects in both 2-space and 3-space. Some results indicate that embedding complex objects in 2 space reduces dimensionality beyond the point of comprehension.[4] New problems arise when portraying 3-D objects on a 2-D medium -- most notably that some people have difficulty concseptualizing the visualization -- but we felt the benefits outweighed the problems. When rendering 3D objects it is often difficult to determine placement without motion (to provide motion parallax) or a stereo viewer. We addressed this problem by initially displaying the visualization rotating, and by supplying the user with tools for turning and moving the graphic.

There have been several prior systems that displayed clusters visually. [3,7] This system differs from them in decoupling the axes from any special meaning. Prior systems would take 2 or 3 (or occasionally 4 or 5) strong concepts, associate them with a direction in space, and allow the clusters to form along these axes. This could indicate the strength of a concept in a document or collection of documents. Visualizing collections of documents in this way has the advantage of presenting the user with a spatial dimension with a known meaning. However, the clusters will vary considerably depending on which terms or concepts are chosen for the axes. This technique is also very sensitive to vocabulary mismatch. For example, when retrieving a collection of documents about *neural nets* and *connectionist systems*, if two documents are very similar in content and in the words they contain, but one uses the term *neural nets* but not *connectionism*, and the other uses the term *connectionism* exclusively, selecting *neural*

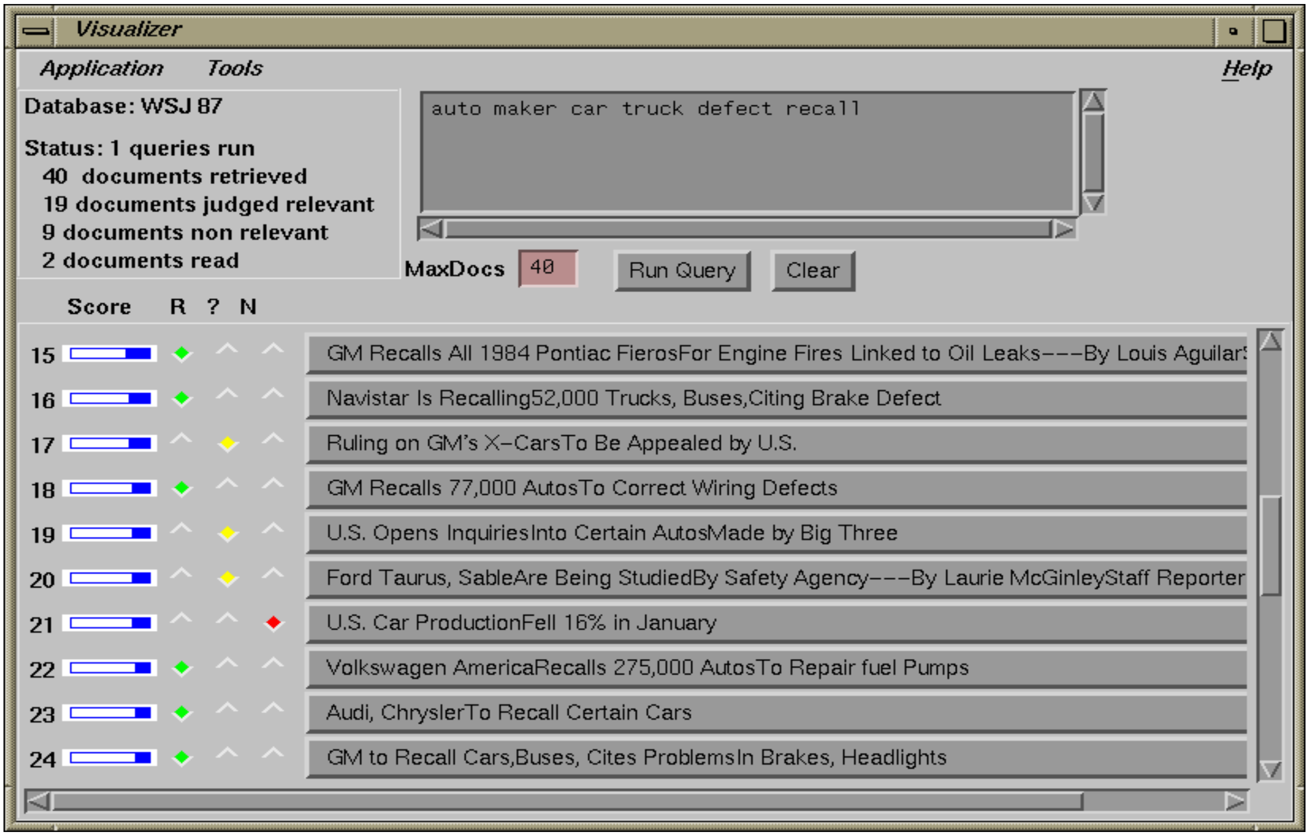

Figure 1 Main Window

nets as the discriminating term would fail to show any proximity between the documents. We believe it is preferable to have the system automatically determine the relationships and let the user browse.

SYSTEM

The retrieval and visualization system consists of five interconnecting windows. Two of those comprise the traditional IR system interface: a query window which also presents a ranked list of retrieved documents, and a viewer for displaying the full text of a selected document. The document map window presented the 3-D visualization resulting from clustering all documents retrieved by the traditional system. The final two windows present the noun phrases which occurred in the retrieved documents, once as a text list, and again as a clustered representation. The following describes the windows in more detail.

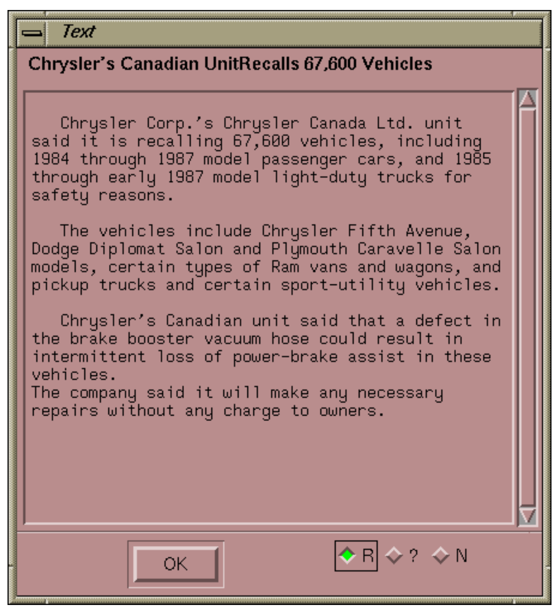

Figure 2 The Text Viewer

Standard System

The system has a main widow, with a query entry area, some status information and controls, and an area for displaying a ranked list. Ranked lists have utility and are a standard and useful tool for an IR system.

A text viewer window for viewing the text of a document. Ultimately the information resides in the text of the document and the user of the system will need to read the text of documents.

The main window and the text viewer window are shown in Figures 1 and 2 respectively.

Document Map

The Document Map was based on the spring embedder discussed above. Some modifications were made to the basic spring embedder to make it more useful here. The basic algorithm of Fruchterman and Reingold[4] was used with modifications from Kamada and Kawai[8] to allow for variable spring lengths. In this model all objects exert a repulsive force on all other objects so the graph spreads as much as allowable. We used the grid variant method of Fruchterman and Reingold[4] whereby repulsive forces ceased after a specified distance; otherwise, low connection graphs have every node forced along the outer edges of the viewing volume. Objects connected by a spring exert an attractive force that is a monotonically decreasing function of spring length. A user definable threshold (slider) was added to the spring embedder, and objects closer to each other than the threshold had their springs added and drawn, objects further away had no springs (and no attractive forces). This led to the objects jumping away from clusters and severe movements when links appeared and disappeared.

A modification was made where objects closer than the threshold had their links appear at full force. Objects further than the threshold had no links drawn, and had their spring forces calculated, then added in severely attenuated. This led to more stability in the graph while still providing separation between objects and clusters. With this partial force arrangement objects usually stay in the same area of space throughout threshold changes, and objects that are within the same cluster at higher threshold values appear near each other. Further, related clusters also appear near each other, and without too much training the user can begin to see how clusters aggregate into larger clusters. If the threshold is set high enough that only one or two clusters appear the the user can discern objects connected closely enough that they will still cluster at low thresholds. Icons for documents were chosen as prisms roughly in the same size ratio as a book.

The Document Map is shown in Figure 3.

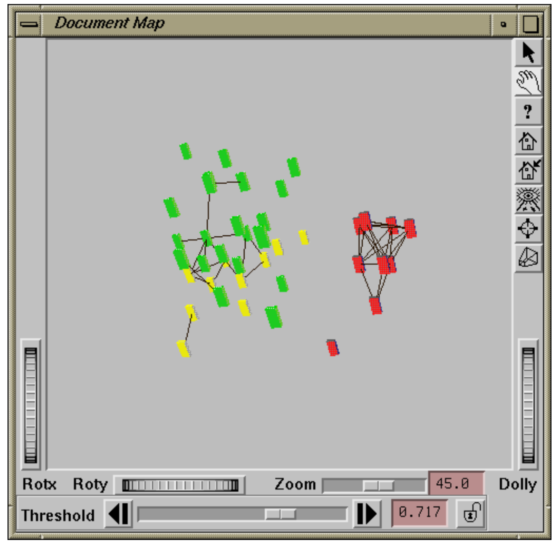

Figure 3 Document Map

Concept map

Standard IR techniques such as relevance feedback [9] have shown that a substantial amount of information is captured by examining terms contained in relevant documents and their relationships. For example, a term which occurs in many relevant documents but in few non-relevant documents is likely to be valuable for retrieval.

We created a concept map visualization in an effort to capture this information visually.

Candidate concepts were chosen by parsing the query and finding terms and phrases that had been indexed. Related terms were found by extracting all terms from the corpus with a high weighted Expected Mutual Information Measure (EMIM) relative to the concepts from the query. The top 40 terms (sorted by weighted EMIM) were selected, and the inter-concept similarities were calculated by taking dot products on their context vectors (representing document co-occurence). Concepts from the query were given a cylinder icon (because it looks like a buoy) and related concepts were given a sphere icon. A text list of the concepts sorted by weighted EMIM value was also presented. The Noun Concept Map is shown in Figure 4.

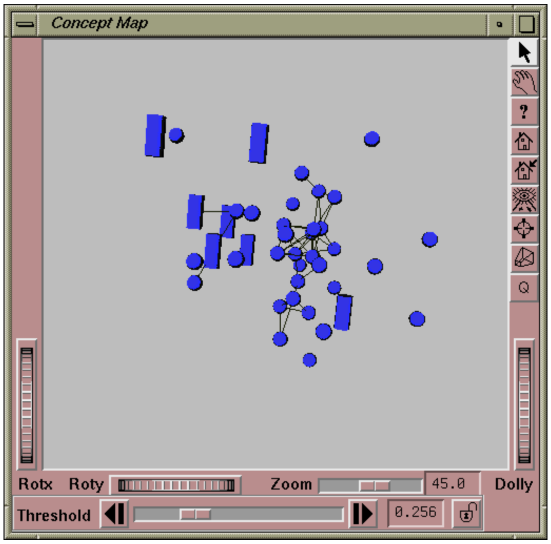

Figure 4 The Concept Map

Relationships

A window with an unordered collection of icons is not that informative. Putting labels on the icons in the window would quickly clutter the display beyond the point of legibility. Since document titles are already available in the list of the main window we linked the list area of the main window with the Document Map so that selecting a document icon in the window scrolls the list to make the surrogate visible and highlights it. Double clicking on a document icon caused the text viewer window to open for that document.

Since the main activity here consists of distinguishing between relevant, non relevant, and unjudged documents we color coded the buttons for selecting the status (buttons available on both the text viewer and the ranked list) and the document icons, with instantaneous updates between displays.

The Noun Concept Map window was linked with the Concept List window so selecting a concept in either window would cause it highlight in both.

The Document Map window was connected to the Concept Map and Concept List windows so that when the user selected a document, concepts that appeared in it would be highlighted in both Concept windows. If the user selected multiple documents concepts that appeared in all selected documents would be highlighted.

The Concept Map window had an “Add to Query” button that would take all selected concepts and add them onto the currently entered query in the main window.

Implementation

We built the system using INQUERY, a probabilistic Bayesian inference network[1]. The system was built and run on a Silicon Graphics Indigo 2 Extreme in ViewKit (A C++ encapsulation of Motif) and OpenInventor (a C++ encapsulation and implementation of OpenGL, the SGI graphics language) and obtained most of its IR functionality by calls to the INQUERY API. Supporting files were obtained by preprocessing the documents into vector files (1 hour processing time, 74 MB storage) and by precalculating term and noun phrase cooccurrence and weighted EMIM scores and storing them in local files (12 hours processing time, 253 MB storage). [14]

EVALUATION

The system as built supports a wide variety of interaction styles and has more functionality than a standard ranked list retrieval system. As a result it is difficult to measure the overall effectiveness of the system. Different users may find different styles of using it, and there are enough available features that the learning curve will have to be accounted for. The authors noticed in their initial experimentation that for certain queries if between 10 and 20 of the top ranked documents were judged, when the Document Map was opened there was a clear separation in the judged documents with most of the relevant documents falling into a well defined cluster and most of the non relevant documents falling into a separate cluster. Analysis of the unjudged documents in the cluster containing the relevant documents showed a far higher proportion of relevant documents than the retrieved set as a whole. While this effect was not consistent for all queries, or even all formulations of the same query, we were quite encouraged by this effect.

Experiment

To measure of the effectiveness we chose a recall oriented task from the TREC collections based upon the TREC-4 Interactive Task.[5]. The corpus used was the Wall Street Journal 1987 subset of the NIST Tipster collection, consisting of 46448 articles and occupying 130 MB. Four users each ran 4 queries with the system, 2 without the visualization tools available, two with the visualization tools. The four queries are given in Table 1. The orders of the queries were randomized so each query was run twice without visualizations as a baseline, and twice with visualizations. The users were told how many relevant documents the TREC evaluators had found in the corpus and were given 15 minutes to find and mark as many relevant documents as possible. The tasks were scored with +3 points for every relevant document correctly marked and -1 points for each non relevant document marked as relevant.

Query: 90

Topic: Data on Proven Reserves of Oil & Natural Gas Producers

Description: Document will provide totals or specific data on changes to the proven reservefigures for any oil or natural gas producer. 46 relevant

Query: 162

Topic: Automobile Recalls

Description: Document will provide data that expands on the type and purpose ofrecalls on vehicles manufactured by American or Foreign automobilecompanies. 52 relevant

Query: 189

Topic: Real Motives for Murder

Description: Document must identify a murderer's motive for killing a person or persons in a true case. 42 relevant

Query: 195

Topic: Stock Market Perturbations Attributable to Computer Initiated Trading

Description: Document will discuss positive ornegative instances of stock market shifts primarily owing to computer assisted trading initiatives. 46 relevant.

Table 1 TREC Queries used in study

Query	90	162	189	195
User 1	18+, 9-	44+, 23-	11+, 4-	12+, 9-
	45 NV	109 V	29 V	27 NV
User 2	13+, 3-	16+, 2-	3+, 0-	1+, 3-
	36 NV	46 V	9 V	0 NV
User 3	10+, 1-	22+, 4-	3+, 3-	2+, 2-
	29 V	62 NV	6 NV	4 V
User 4	25+, 26-	36+, 12-	7+, 7-	1+, 19-
	49 V	96 NV	14 NV	-16 V

Table 2 Study results

Empirical results

Table 2 presents the actual scoring for each run. In 3 of the 4 queries the highest score was generated using the visualization tool. In 3 of the 4 queries the lowest score was generated using the visualization tool. The sample size is too small and the data is too preliminary to generate any meaningful statistics. User 1 had the highest score on 3 of 4 queries, and the second highest score on the 4th. User 4 had the highest score on 1 query, the 2nd highest on 2, and the worst on one. Differences between users overwhelmed the differences between tools.

Subjective results

The users were asked to comment on their impressions of the systems. All 4 users were experienced IR system users, all of whom worked in the IR lab and all of whom had extensive experience with INQUERY. Some sample comments were "I know how to use an IR system. You start at the top and work down". Several commented that they could see how this tool might be valuable, but they had to remind themselves to use it. They had never realized before how much they relied on the notion of a ranked list, and how ingrained the idea of working through a list was. All participants said they enjoyed the study and felt they could have done better if they had had more experience with the tool, and all expressed a desire to have it ported to their systems so they could experiment with it more. Several said they would like to repeat the study after they had had the tool for a while. All felt that the document map was the most useful part of the visualization, and most found that the concept map was not very useful. The concept map was a partial implementation of an association thesaurus, which is an active area of research in IR. Users who had had experiences with association thesauri were disappointed at the weakness of the implementation.

FUTURE WORK

Scalability is always an issue with any prototype system. This system should scale fairly well. The basic IR engine is INQUERY which handles very large corpora very efficiently. Graphics engines can routinely display scenes of 1000 objects without being overwhelmed. The spring embedder algorithm used is $n^2 \log n$, but there have been reports of faster spring embedders.

Vector preprocessing was employed because the current release of INQUERY (3.1) does not support efficient construction of document vectors on the fly. We may make changes to the engine to support this.

Term cooccurrence data is basically the construction of a corpora based association thesaurus. This is a significant problem involving significant resources. If the presence of this view does not add to the functionality of the system this will be dropped, because of both the preprocessing time and the memory requirements.

Specific UI issues involve how many icons can be displayed on a 3D viewer before comprehensibility is lost. Clusters that are heavily linked could be replaced with special icons to enhance readability.

This system holds a lot of promise as an improved method of interaction with a text based IR system in a recall oriented session. The tools seem fairly well integrated, and orthogonal to several other tools developed for IR [ref. Xerox stuff] so the collections could be combined in a more user customizable toolkit. A larger user study is planned to see if improvements can be obtained in recall-oriented queries with trained users, or if information can be gathered on what kinds of queries offer improvements to this kind of visualization.

This system is still fairly novel, and the interpretation of the cluster shapes may have a lot in common with glyphs in being a high bandwidth form of communication. Like glyphs the interpretation is not intuitive but is a language that must be learned. Unlike glyphs the cluster shapes are a data artifact and are not human designed so whether there are real data being expressed, and how to interpret it, will have to wait for experience to decide. There are enough novel aspects of this system that watching users interact with the system may teach us new ways of using the system.

Future activities include ports to other systems and possibly to a Web site to allow more access to users.

ACKNOWLEDGEMENTS

We would like to thank Bruce Croft for suggesting spring embedders. We would also like to thank Morris Hirsch, Donald Byrd, Anton Leouski and Bill Berry for their participation in the user studies and their helpful comments.

This material is based on work supported by the National Science Foundation, Library of Congress, and Department of Commerce, under cooperative agreement number EEC-9209623. It is also based on work supported in part by NRaD contract N66001-94-D-6054.

REFERENCES

1. Callan, J., Croft, W. B. and Harding, S. "The INQUERY Text Retrieval System" in *DEXA-3: Third International Conference on Database and Expert Systems Applications* (1992)

2. Croft, W. B. *Organizing and Searching Large Document Collections*. PhD Thesis University of Cambridge, 1979.

3. Dubin, D "Document Analysis for Visualization" *in Proceedings of SIGIR '95* (Seattle, Washington, July 1995) ACM Press, 199-204

4. Fruchterman, T.M.J. and Reingold, E.M. "Graph Drawing by Force-directed Placement", from *Software -Practice and Experience* Vol. 21 (11) 1129-1164 (November 1991)

5. Harman, D Editor, *The Fourth Text REtrieval Conference* (TREC-4), 1996 Forthcoming.

6. Hearst, M. A. and Pedersen, J. O. "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results" *in Proceedings of SIGIR '96* (Zurich, Switzerland, August 1996) ACM Press 4-11

7. Hemmje, M. Kunkel, C. Willett, A. "LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval" in *Proceedings of SIGIR '94* (Dublin, Ireland, July 1994) ACM Press, 199-204

8. Kamada, T. and Kawai, S. "An Algorithm for Drawing General Undirected Graphs" *Information Processing Letters* 31 (1989) 7-15

9. Koenemann, J. and Belkin, N. "A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness" in*Proceedings of CHI '96* (Vancouver, British Columbia, April 1996) ACM Press, 205-212

10. Pirolli, P. et al "Scatter/Gather Browsing Communicates the Topic Structure of a Very Large Text Collection" in *Proceedings of CHI '96* (Vancouver, British Columbia, April 1996) ACM Press, 213-220

11. Rao, R., et al. "Rich Interaction in the Digital Library", *Communications of the ACM*, April 1995 Vol. 38 No. 4 29-39

12. Salton, G. *Automatic Text Processing*, Addison-Wesley 1988

13. van Rijsbergen, C. J. *Information Retrieval*, Butterworths, 1975

14. Xu, J. amd Croft, B. "Query Expansion Using Local and Global Document Analysis" *Proceedings of SIGIR '96* (Zurich, Switzerland, August 1996) ACM Press 4-11