Studying the Usability of Relevant Proximity Ranking

Anton Leuski
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003 USA
leuski@cs.umass.edu

Motivation

A ranked list returned by an information retrieval system lists the documents in the order they are expected to match the user's query: the first document is most likely to be the most relevant, the second is the next one most likely to be helpful, and so on. It is expected that the user will follow these recommendations, starting at the top of the list and following it down, reading documents one by one. It is a well-known and widely accepted method for presenting the retrieved information and helping the user to find relevant documents. Ideally, the user will see all the relevant documents before any non-relevant ones, though quite often the relevant documents appear to be scattered all over the ranked list.

Automatic clustering techniques are considered to be very successful in grouping similar objects. It is also believed [8, p.45] that a good clustering of the retrieved documents will bring together the documents relevant to the user's query. Numerous visualization approaches for clustering were developed in recent years. They range from text-centered presentations [5] to 2- and 3-dimensional graphical presentation that require high-powered workstations [2].

We are interested in combining the ranked list with a clustering visualization in hope that by leveraging the individual strengths of each approach we can increase the retrieval effectiveness – i.e., help the user find the relevant documents more quickly than she would with the ranked list alone. We expect that the clustering will group similar documents together and the ranked list will point to the relevant group of documents.

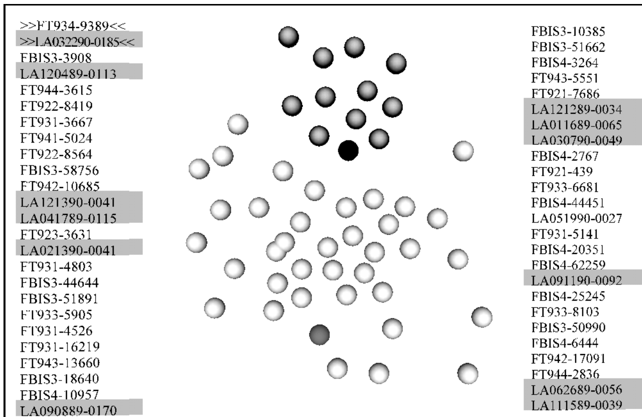

Figure 1: The combination of a ranked list and spring-embedding visualization for a group of fifty retrieved documents. The ranked list is presented on the two panels starting at the top of the left panel and continue down and then to the top of the right panel. The 2-dimensional visualization is in the middle. The first document in the ranked list is non-relevant and the corresponding gray sphere is at the bottom of the visualization. The second document is relevant. It is highlighted with a gray background in the ranked list and its sphere is the black one almost in the center and slightly to the top of the visualization. The rest of the relevant documents are highlighted with a gray in the ranked list and presented with dark gray spheres in the visualization. There are 12 relevant documents total.

System

We have designed a system that combines the ranked list with a 2- or 3-dimensional clustering visualization approach. The ranked list consists of fifty top ranked documents ordered as returned by INQUERY [1]. For the clustering we use a spring-embedding approach from earlier work [7], similar to that found in BEAD [2]. It is a force-directed-placement graph drawing algorithm that generates an approximate solution to a graph layout when the distances between connected nodes are given as constraints [3]. We use inter-document similarities as constraints. The spring-embedding handles both the clustering task and the task of visualizing the clusters.

Figure 1 shows an example of our system. There are fifty documents presented in the ranked list and a 2-dimensional visualization. The document representations in the list and in the visualization are tightly linked: a click on a sphere highlights both the sphere and the corresponding document id in the list and vice versa. (The document ids can be replaced by titles, but ids are shown here to save space.) This figure clearly illustrates the advantages of the clustering visualization over the ranked list: although the relevant documents are widely scattered in the ranked list, the same documents are tightly grouped together in the visualization.

Automatic Use of Clustering

The spring-embedding attempts to map the similarity between documents onto the Euclidean distances in the picture.

Ideally the more similar the documents are, the closer they are displayed in the picture. In another study we show [6] that we can use this spatial "closeness" to generate a significantly better ranking of the documents than the original ranked list. We consider a scenario when the top ranked relevant document is known to the system (e.g, a user starts from the top of the ranked list and follows it until she finds one relevant document). Then we re-rank the rest of the documents based on the spatial proximity in the visualization. We show that the average precision of this *relevant proximity ranking* is higher than the average precision of the ranked list by 17% on the TREC-5 and TREC-6 ad-hoc task [4]. It also exceeds the average precision of the ranking created by running an automatic relevance feedback method, modifying the original query, and re-ranking the documents.

If the user is willing to provide the system with relevance judgments as she examines the documents, the system creates the new ranking by ordering the documents based on their distance from the center of mass of all the found relevant documents. Thus, the ranking is adjusted each time the user discovers a new relevant document. The average precision for this "interactive" ranking exceeds the average precision of the ranked list by 23%.

Research Questions

Our system uses simple proximity clues in the visualization to generate the improved ranking of documents automatically. We are interested in whether people are able to recognize and interpret the same proximity clues as effectively as the system does. If they select documents in a less effective order than the automatically generated ranking, we must incorporate the ranking generation mechanism into the system as a "document selection wizard" that "suggests" the best document to the user.

The visualization properties used to generate the new rankings are very simplistic and rely only on the distances between the individual document representations. We are looking at the different ways people forage for information in this type of visualization. What other type of information, besides proximity, do people receive from the visualization? Do they take into account the shape of the picture, the existence of clumps and gaps? Do they generally follow one direction in the visualization and change it only when unsuccessful?

Based upon the answers to these questions, we can modify the spring-embedding algorithm to take into account such clues and produce more effective presentations of clustering.

User Study

To explore these questions we have designed a user study. We randomly selected two dozen topics from TREC-5 and TREC-6 [4]. The title field of each topic was used as a query for INQUERY. The top ranked fifty documents were then spring-embedded in 2- and 3-dimensions. Each embedding became an information foraging problem for the users to solve: at the beginning the spheres representing the documents are colored in white. The users are told that (1) spheres are actually of two colors: red and green; (2) the true color of a sphere is shown by clicking on it; (3) the spheres of similar color tend to appear close together. The users are asked to find all the green ones as quickly as possible. At the beginning of each problem at least one green sphere is shown – i.e., the sphere corresponding to the highest ranking document. Also all non-relevant document that appear above that document in the ranked list are shown in red.

The system is implemented in Java with elements of JavaScript and VRML. The complete study together with all accompanying questionnaires can be found on-line [9].

Acknowledgments

I am deeply grateful to James Allan for his support and contributions to the project.

This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

This material is based on work supported in part by Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235.

References

[1] James Allan, Jamie Callan, W. Bruce Croft, Lisa Ballesteros, Donald Byrd, Russel Swan, and Jinxi Xu. Inquery does battle with TREC-6. In *Sixth Text REtrieval Conference (TREC-6)*, pages 169–206, 1998.

[2] Matthew Chalmers and Paul Chitson. Bead: Explorations in information visualization. In *Proceedings of ACM SIGIR*, pages 330–337, June 1992.

[3] Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software–Practice and Experience*, 21(11):1129–1164, 1991.

[4] Donna Harman and Ellen Voorhees, editors. *The Sixth Text REtrieval Conference (TREC-6)*. NIST, 1998.

[5] Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of ACM SIGIR*, pages 76–84, August 1996.

[6] Anton Leuski. Combining ranked list and clustering: the best of both worlds. Technical Report IR-172, Department of Computer Science, University of Massachusetts, Amherst, 1999.

[7] Anton Leuski and James Allan. Evaluating a visual navigation system for a digital library. *International Journal on Digital Libraries*, 1999. Forthcoming.

[8] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, London, 1979. Second edition.

[9] http://www-ciir.cs.umass.edu/~leouski/SE99.html.