Relevance Feedback at the INEX 2004 Workshop

Carolyn Crouch
Department of Computer Science
University of Minnesota Duluth
ccrouch@d.umn.edu

In 2004, the INitiative for the Evaluation of XML Retrieval (INEX), in its third year of investigations into various aspects of theoretical and applied structured retrieval, added a Relevance Feedback (RF) Track. The purpose of this Track is to explore issues related to the use of relevance feedback in a structured environment. Because the interim between the receipt of relevance assessments by Workshop participants and the due date of their papers was relatively short, the amount of time available for experiments was limited. However, three groups reported early experiments: Mass and Mandelbrod (Relevance Feedback for XML Retrieval); Mihajlovic, Ramirez, de Vries, Hiemstra, and Blok (TIJAH at INEX 2004 Modeling Phrases and Relevance Feedback); and Crouch, Majahan and Bellamkonda (Flexible Retrieval Based on the Vector Space Model).

The most extensive investigations are reported by Mass and Mandelbrod, who add relevance feedback to their basic retrieval algorithm. Working within a traditional information retrieval environment based on the Vector Space Model, the authors describe their approach to structured retrieval based on component ranking. This approach first determines the "most informative" component type in the collection (articles, sections, and paragraphs for INEX) and then creates a separate index for each type. The query is run against each index and a set of components retrieved. The scores in the retrieved sets are normalized to enable comparison across indices and then scaled by a factor related to the score of the containing article in the article index. Results are merged to form a rank-ordered list of components associated with the query.

Relevance feedback proceeds as follows. For each query, the top N components in the rank-ordered list are examined for relevance or non-relevance as per their assessments. In contrast to traditional retrieval, where the assessed components are all of the same type, the components here are of differing granularities. But the authors claim that such a component can be used in the RF process regardless of granularity. They illustrate by using two different approaches to the construction of the feedback query, namely, Rocchio's algorithm and Lexical Affinity. For each index, the feedback query is constructed and run against the index, the resultant scores are normalized and scaled as above, and the results are merged in rank order. Early results show small improvements attributable to relevance feedback over the base case.

A very different view is taken by Mihajlovic, *et. al.*, who extend their database approach to include what they refer to as structural relevance feedback. They contend that knowledge of component relevance provides "implicit structural hints" which may be used to improve performance. Their implementation is based first on "extracting the structural relevance" of the top-ranked elements and then restructuring the query and tuning the system based on RF information. They argue that if a component is assessed as relevant for a given topic, the journal it is contained in is apt to contain similar information, so journal name is used to model structural relevance. The XML tag name is also utilized in this process, with the objective of raising in the ranked results list those types of elements known to be relevant for the topic. Size of an element is also used, based on the idea that elements which are similar in length to relevant elements are more likely to be those sought. Based on the structural information and assessments associated with the relevant elements, the query is rewritten and evaluated. Early results indicate that further experimentation is required to test parameter settings and evaluate the current interpretation and use of the relevance assessments.

Crouch, *et. al.*, implement relevance feedback in a conventional information retrieval environment based on the Vector Space Model. This paper focuses on their approach to flexible retrieval, which allows the system to retrieve at the element level. The paragraph is selected as the basic indexing unit, and the collection is indexed on paragraphs. The topics are run against this index, and a rank-ordered list of paragraphs having a non-zero correlation with the query is retrieved. Every such paragraph represents a terminal node in a tree which represents the document that paragraph is contained in. The flexible retrieval module is then called to produce a corresponding set of elements (upper level nodes in the tree such as sub-sections, sections, and body) with associated values based on e- and s-values propagated from below. The elements from all such trees are sorted on value and the top N are reported for this query.

A simple experiment in relevance feedback is performed as follows. The top 20 paragraphs retrieved from an initial search are examined for relevance. (Only highly exhaustive paragraphs are used in this case.) A feedback query is constructed based on Rocchio's algorithm. The result of the feedback iteration is another list of rank-ordered paragraphs. Flexible retrieval is performed on this set to produce the associated elements. Again, small increases in average recall-precision were produced.

As these papers indicate, much discussion and experimentation is required in future months to see how relevance feedback methods can best be utilized in a structured environment. In INEX '04, basic investigations began.

For those who are interested, these papers will be available in the *INEX 2004 Workshop Proceedings* as a part of the Lecture Notes in Computer Science series by Springer-Verlag. Information about INEX is available on the website at *http://inex.is.informatik.uni-duisburg.de/*