Evaluating the Accuracy of Query Refinement

Bienvenido Vélez-Rivera and Ron Weiss*
{bvelez,rweiss}@lcs.mit.edu
MIT Laboratory for Computer Science

Query refinement, initially introduced by the Content Routing System (CRS)[2], assists a user in formulating well focused queries to information retrieval systems by suggesting additional query terms. Practical query refinement requires fast and accurate algorithms and data structures, and therefore an essential step in its adoption is the discovery of reliable measures of accuracy.

This paper introduces the first metric for comparing query refinement algorithms using standard collections, together with the first set of empirical measurements of query refinement known to the authors. Two approaches for query refinement that rely on collocation of terms in documents coupled with lossy compression techniques are considered. The accuracy metric allowed the discovery of improvements to algorithms previously implemented in content routing prototypes. The experiments also provide evidence that the proposed query refinement algorithms are effective on short queries.

As shown in figure 1, the query refinement paradigm assumes that the user starts with a simple general query that would match many relevant documents. Query refinement incrementally drives the process of focusing the general query by iteratively offering the user an automatically generated set of terms that can be chosen to make the query more specific. As an example, in response to the query "scheme", the HyPursuit [3] CRS prototype that indexes documents from several MIT web sites suggests terms including: "applications", "compiler", "interpreter", "abelson" and "closures".

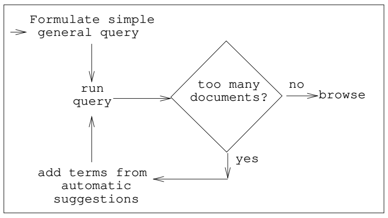

Figure 1: The query refinement paradigm.

The query refinement algorithms evaluated here are based on two different approaches that rely on collocation of terms: *cum-tf* and *min-df*. *Cum-tf*, first computes the set of documents that contain one or more terms from the user's query. It then suggests to the user the terms with the highest cumulative frequency over the above set of documents. The second approach, *min-df*, computes a matrix whose rows and columns are the terms that appear in the document corpus. The matrix entry for terms t_i and t_j stores two values. The first value is the number of documents containing both t_i and t_j. The second value is the sum over all documents in which both t_i and t_j appear of the minimum term frequency of t_i and t_j in each document. Given a query, *min-df* merges all the rows of terms that appear in the query and suggests the terms with the highest matrix entry values.

In the content routing system, scalable query refinement is supported through a process of controlled information loss. Term suggestions based

*The authors are advised by Prof. David K. Gifford and supported by ARPA contract DABT63-95-C-0005. Address: Room 405, 545 Technology Square, Cambridge MA 02139. Web: http://www.psrg.lcs.mit.edu/

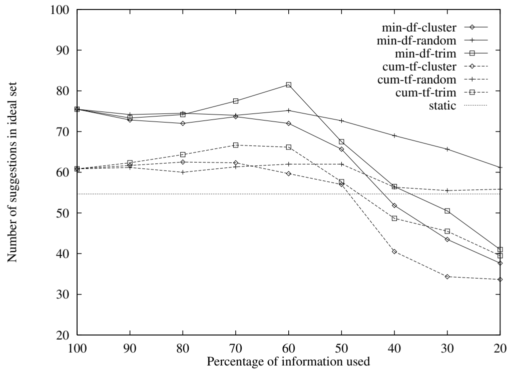

Figure 2: Average Query Refinement accuracy: short queries

on a summary of the document corpus approximate the suggestions that could be obtained from the original corpus. This experiment couples three different information loss mechanisms with both *cum-tf* and *min-df*. *Cluster* groups and summarizes documents that are similar in content, *random* stores a randomly selected subset of the original document corpus, and *trim* drops terms from each document but maintains information on a per-document basis.

The experimental design introduced here measures the accuracy of each algorithm as the ratio of the number of *"ideal"* terms suggested by the algorithm to the *total* number of term suggestions. With the aid of standard collections (consisting of documents, queries and relevance judgements) this experiment defines the "ideal" set of suggestions for a given query to be the union of all terms in documents relevant to the query. Figure 2 illustrates accuracy measurements for the six query refinement algorithms performed on the MED standard collection [1] (1033 documents, 30 queries). The figure also indicates the accuracy of a naive "static" algorithm that always suggests the same set of the 100 highest weighted terms in the collection for any given query.

The figure illustrates that the accuracy of all algorithms remains fairly stable up to the 60% information loss point, and may actually improve as in the case of the *trim* mechanism. It also shows that the *min-df* algorithm outperforms *cum-tf*, used by previous prototypes, in all information loss mechanisms. A surprising result is that *random* information loss performs quite well. Another observation from this experiment (not shown by figure 2) is that all algorithms considered in this study appear to be more accurate on shorter queries. This is a desirable property for two reasons: most users tend to write short queries and the query refinement paradigm assumes that users start with short queries.

Query refinement can become an essential information retrieval service in future search engines, provided that fast and accurate algorithms are developed. Directions for future work include experimenting with more adequate collections (more and larger documents and smaller queries) and incorporating measurement methodologies that discriminate between "ideal" terms.

References

[1] G. Salton. *The SMART Retrieval System-Experiments in Automatic Document Processing*. Prentice Hall, 1971.

[2] Mark A. Sheldon, Andrzej Duda, Ron Weiss, James W. O'Toole, Jr., and David K. Gifford. A content routing system for distributed information servers. Technical Report MIT/LCS/TR-578, M.I.T. Laboratory for Computer Science, June 1993.

[3] Ron Weiss, Bienvenido Vélez, Mark A. Sheldon, Chanathip Namprempre, Peter Szilagyi, and David K. Gifford. HyPursuit: A hierarchical network search engine that exploits content-link hypertext clustering. In *Proceedings of the Seventh ACM Conference on Hypertext*, Washington, DC, March 1996.