Comparing Relevance Feedback Algorithms for Web Search

Vishwa Vinay
University College London, London, UK
v.vinay@cs.ucl.ac.uk

Ken Wood
Microsoft Research Ltd., Cambridge, UK
krw@microsoft.com

Natasa Milic-Frayling
Microsoft Research Ltd., Cambridge, UK
natasamf@microsoft.com

Ingemar J. Cox
University College London, London, UK
ingemar@ieee.org

ABSTRACT

We evaluate three different relevance feedback (RF) algorithms, Rocchio, Robertson/Sparck-Jones (RSJ) and Bayesian, in the context of Web search. We use a target-testing experimental procedure whereby a user must locate a specific document. For user relevance feedback, we consider all possible user choices of indicating zero or more relevant documents from a set of 10 displayed documents. Examination of the effects of each user choice permits us to compute an upper-bound on the performance of each RF algorithm. We find that there is a significant variation in the upper-bound performance of the three RF algorithms and that the Bayesian algorithm approaches the best possible.

Categories and Subject Descriptors

H.3.3 Information Search and Retrieval – *relevance feedback*

General Terms

Algorithms, Performance, Experimentation

Keywords

Relevance Feedback, Web Search, Evaluation

1.INTRODUCTION

Relevance feedback is a classical information retrieval (IR) technique where users relay their agreement with the system's evaluation of document relevance back to the system, which then uses this information to provide a revised list of search results. Even with state-of-the-art search engines users are often dissatisfied with the returned results and have to manually alter their query in order to retrieve relevant documents. Despite this problem, Web search engines of today do not provide the option for relevance feedback (AltaVista initially included this facility and Google does have a "Similar pages" option, but these are different from the mechanisms used in traditional IR). This is partly due to the fact that users do not understand the mechanisms of the RF algorithms, and partly to the fact that providing relevance judgments requires additional effort on the users' part.

In this study we explore the effectiveness of relevance feedback methods in assisting the user to access a predefined target document. We devise an innovative approach to study this problem by exploiting the fact that while the number of user choices is large, it is still limited. It is therefore feasible to generate and study the complete space of possible user's choices, for one round of feedback on the relevance of presented documents, and obtain the upper bound on the effectiveness of the applied relevance feedback algorithm. This upper bound can be interpreted as the result achieved by an "ideal user" whose choices enable the system to gather optimal information for relevance feedback. This approach has the further advantage of permitting the study of relevance feedback without dealing with complexities of user studies.

2.RELEVANCE FEEDBACK

Search results are typically presented to the user as a ranked list of documents, in the decreasing order of relevance to the user's query. In a system that involves user relevance feedback the user is given an opportunity to inspect the ranked list and indicate which documents are relevant to the user's query and which are not. This information is then used by the relevance feedback algorithm to induce a new ranking of documents. The new ranking, possibly including new documents, is displayed to the user and the process repeats.

In this study, we consider only one iteration of feedback. This is motivated by the fact that real users rarely go beyond the first two screens of search results. It is also a practical constraint of our approach since we explore all the possible document choices for feedback and thus have to deal with a high branching factor in generating the space of possible selections. Therefore, focusing on one feedback iteration, we perform exhaustive evaluation of the feedback model but look at the immediate effect only.

The feedback process comprises several phases. The display phase is the presentation of ten documents from the ranked list. The user feedback phase is a single action where the user nominates some subset of the displayed documents as being relevant to his or her information need. The document ranking phase applies one of the relevance feedback algorithms to create a new ranking of the document collection. In this paper, we compare three standard relevance feedback algorithms: The Rocchio algorithm[7] the Robertson/Sparck-Jones algorithm[6]and a Bayesian feedback algorithm[1]

3.EXPERIMENTAL PROCEDURE

We chose to compare different systems based on the position of a known target in a ranked list after feedback. For comparison purposes, this number is compared with the rank of the document after the initial query, i.e. before any relevance feedback is applied. We examine the effectiveness of relevance feedback over the complete space of possible user's interactions with the system within this particular scenario. Assuming a display size of 10 for web search results, this gives us 2^{10} = 1024 ways of choosing relevant documents from the displayed set. Each such combination can be fed back into the feedback algorithm and the position of a known item can be noted. This position can be compared with the rank of the same item in the initial ranked list to measure the potential (dis)advantage of relevance feedback. The one combination which pushed the target to have the highest position

Copyright is held by the author/owner(s).
WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

is the optimal feedback. The average rank improvement can also be calculated.

To use this evaluation paradigm, we need specific query-result pairs. These are referred to as *definitive queries*, i.e. queries which have a single HTML page as their target. These were collated from an internal MSN study of relevance judgments in which real users matched short web-style queries to URLs which were the answers to these queries.

For every query, the following metric is calculated:

$$\text{Best\_Improvement}_{\text{tree\_n}} = \max(R_{\text{initial}} - R_i) \text{ for } i = 1,2..,1024$$

where R_{initial} is the rank of the target in the initial list and R_i is the position in the re-ranked list caused by combination 'i' as the set of relevant documents.

The experiments were performed using an API that allowed querying the MSN Search Engine ([3]). The API allowed access to the publicly available MSN search engine during the period from June to August 2004. Up to 500 results were gathered for each of the definitive queries – up to 60% of the queries returned the result in the top 10, i.e. the first page. The subset of queries which contained the target between rank 11 and rank 500 of the returned results were used for building the trees. Over 30% of the remaining queries did not contain the target URL in the set of results returned – for many of these cases, the updated index of the search engine did not contain the target page because the page no longer existed on the Web. Up to the first 1000 words of the HTML page pointed to by each URL in the set of initial results returned by the search engine for the remaining 54 queries was used as its *local database*, against which relevance feedback was performed.

4.RESULTS

The following are graphs of the results produced from the data gathered. Each point in the graphs corresponds to one query-result pair. The best improvement for that pair is plotted as the y-axis and the initial rank is on the x-axis. By plotting this information, we expect to see how close our algorithms are to the ceiling imposed by the line y = x – 11 (the dotted line in the graphs). This best case scenario is when the target document is ranked first after one iteration of relevance feedback. The closer the points in the graph approach this line, the closer their performance approaches optimum. In each set of graphs, the best fit straight line for the data points is also provided, as is the equation of this line. It is clear from Figures 1, 2 and 3 that the Bayesian algorithm's performance is superior.

5.REFERENCES

[1] Cox, I. J., Miller, M.L., Minka, T.P., Papathomas, T.V., and Yianilos, P.N. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical Experiments. IEEE Transactions on Image Processing, 9(1): 20-37, 2000.

[2] Dean J., Henzinger M. R. Finding Related Pages in the World Wide Web. In Proceedings of the 8th International World Wide Web Conference 1998, pages 389-401

[3] Harman, D. Relevance feedback revisited. Proceedings of SIGIR 1992, Copenhagen, 1992.

[4] Jansen, B. J., Spink, A. & Saracevic, T. 1999. The use of relevance feedback on the web: Implications for web IR system design. 1999 World Conference on the WWW and Internet, Honolulu, Hawaii

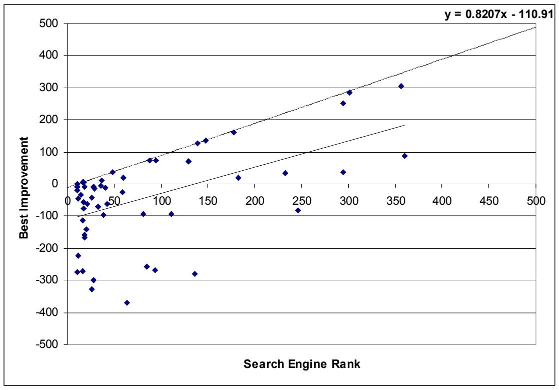

Figure 1: Rocchio RF algorithm

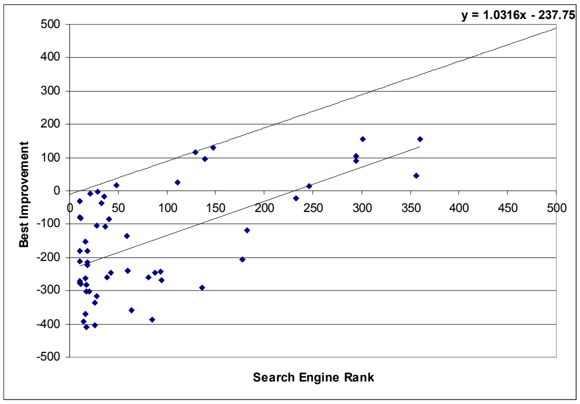

Figure 2: RSJ RF algorithm

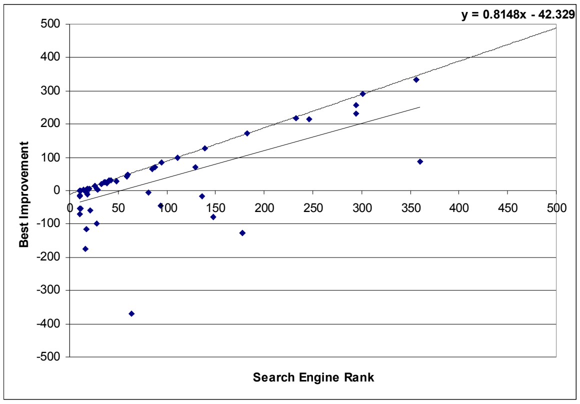

Figure 3: Bayesian RF algorithm

[5] MSN Search (http://search.msn.com)

[6] Robertson, S.E., Sparck-Jones, K. Relevance weighting of search terms. Journal of the American Society for Information Science 27, 1976, pp. 129-146.

[7] Rocchio, J. Relevance feedback informarian retrieval. In Gerard Salton (ed.): The Smart Retrieval System — Experiments in Automatic Document Processing, pp. 313–323. Prentice-Hall, Englewood Cliffs, N.J., 1971

[8] Vinay, V., Cox, I. J., Milic-Frayling, N., Wood, K. Evaluating Relevance Feedback Algorithms for Searching on Small Displays. ECIR 2005.