

Chapter 1.

INTRODUCTION

The research project reported here represents an attempt to conduct an evaluation in an operational context (with real users, queries, databases, hosts, networks etc) of an information retrieval system using weighting, ranking and relevance feedback.

The relevance weighting theory has been the subject of a large number of laboratory experiments (1-5), and the theory has performed well enough in these tests to suggest that an operational test would be desirable. Furthermore, it would be appropriate to compare a retrieval technique based on relevance weighting with conventional retrieval methods using Boolean and pseudo-Boolean techniques. One method for implementing this type of experiment is a front-end system, providing access to an existing host with the option of employing either one of the two methods of retrieval.

Jamieson and Oddy proposed an experiment very much along these lines (6), but unfortunately ran into technical difficulties, and did not produce a system which could be tested. The present project is the second part of a two stage plan to design and test a front-end system. The first project was concerned with the separate development of the prototype front-end, Cirt, and was successfully completed by Robertson and Bovey in 1983 (7). This second project utilises Cirt in an attempt to evaluate the two types of retrieval, ie Weighted, incorporating weighting, ranking and relevance feedback, and traditional Boolean. (For the remainder of this report, "weighting ranking and relevance feedback" will be abbreviated to Weighted searching, and traditional methods including Boolean and pseudo-Boolean operators and intermediate search sets will be referred to as Boolean searching.)

An overview of the aims and methods of the present project is given in (8).

1.1. The theory

Cirt is based on the probabilistic approach to information retrieval, as applied to search term weighting. The theory leads to a weight for each search term. This search term weight is calculated by the front-end system from the frequency characteristics of the term in relation to relevant and non-relevant documents. The weight may be estimated either from partial relevance information derived from viewing references and tagging them as relevant or not, or (in the absence of such information) from raw frequency data, i.e. term postings and total size of the collection. The theory also determines that the match function should be a simple sum of weights (1).

A complete technical specification of the formula used would be as follows:

- (a) The basic relevance weighting formula is formula 4 as specified in reference (1);
- (b) Estimation is by the point-five version of formula 4 (1);
- (c) The non-relevant parameter is estimated by the complement method (5);
- (d) In the case of no relevance information, the simplest estimate ($p=0.5$) of the relevant parameter is used (9);
- (e) Where a search is performed on two databases in succession, the occurrence of each term in any relevant documents is identified in the first database, and this contributes to the calculation of the term's weight in the second (10).

1.2. Aims

The aims of the present project are, firstly, to compare the two types of retrieval in an operational environment; and secondly, to establish the operational feasibility of a system which uses weighting ranking and relevance feedback implemented as a front-end to a traditional Boolean system. For the current experiment the front-end has been developed purely as a tool to provide insights into the capabilities of an interactive system. The hope is that the prototype will be robust enough to respond to the varied demands imposed upon it, and supply sufficient data to establish the feasibility of weighted searching in comparison to Boolean searching in as many circumstances as possible. Any further development of the front-end for either

commercial exploitation or additional exploration of weighting would be the subject of subsequent investigations.

1.3. Functional description of Cirt

Cirt permits the execution of both Weighted and Boolean searches. At present the only host Cirt talks to is Data-Star, and although two other databases are available on Cirt (Psychological Abstracts and Inspec) the most heavily used database is Medline and its various divisions. For Boolean searches, Cirt operates transparently, and once beyond the initial logging on phase searching is conducted in the usual Data-Star vernacular. For Weighted searches, on the other hand, Cirt operates opaquely: the searcher uses Cirt's command language, and Cirt then generates commands and Boolean statements comprehensible to Data-Star. Data-Star's responses are then interpreted by Cirt before transmission to the searcher.

Because Data-Star (in common with other hosts) does not offer any kind of weighted retrieval, Cirt operates a search algorithm (11) which involves translating a weighted search into a series of Boolean searches and sending them off one by one to Data-Star, waiting each time for a response. This procedure is somewhat protracted, given typical Data-Star response times and transmission speeds, making Cirt's responses to the user slower than one would like.

The number of user interface refinements required to make the system more user friendly have been kept to a minimum. Nevertheless the command driven system can permit the intermediary to: add and delete terms; perform a search; examine and evaluate records from the top of the ranked list; print references offline; change databases; assign limits; and save and transfer searches from one database to another.

A typical weighted search would proceed as follows. After logging on the intermediary would first be asked to specify any limits. The choices include year, language, human/ animal, female/ male, and any other acceptable Medline check tag or negative Boolean statement. The intermediary would then add the requisite search terms, either singly or in a string, using any Mesh search term facility (explosion, Mesh headings etc), natural language terms or Data-Star search capability such as truncation or adjacency. Having added or deleted terms as required, the intermediary would then search. Subsequent to the search a display of sets in rank order is automatically provided. At this stage it is possible to enrich the search by once again adding or deleting terms or alternatively displaying a selection of individual documents

within the sets, marking those which are relevant. Sets are taken in rank order, but a set can be skipped or printed offline. The display can be stopped at any point; the system will then automatically re-weight the terms according to the relevance information provided. This procedure can then be iterated any number of times until it is decided to terminate the search.

1.4. Technical description of Cirt

Cirt is written in the C programming language and run under Unix on an LSI 11/23 machine. In addition to the normal C and Unix facilities, Cirt makes extensive use of: (a) Lex, a lexical analyser generator in the form of a C preprocessor, which is used for the parts of the program concerned with interpreting Data-Star's responses; and (b) the "York Box" University of York Unix-X25 (packet-switched) network interface.

The network facilities take the form of a connection to JANET (the UK Joint Academic Network), which provides links both to UK academic institutions, and to national and international networks through which Cirt gets access to Data-Star in Switzerland.

1.5. Structure of the report

The first few months of the project were spent on further development of Cirt, to ensure that we had a system that was useable under real-life conditions. The changes made are described in Chapter 2, and their relation to the relevance feedback model is considered. The methodology of the evaluation experiment is described and discussed in Chapter 3, and the results and discussion in Chapter 4. The various problems encountered during the course of the project are covered in Chapter 5. Finally Chapter 6 contains our conclusions and recommendations.