To appear in *Proc. of WebNet '96: The First World Conference of the Web Society*, San Francisco, CA, October 1996.

Information Fusion with ProFusion*

Susan Gauch, Guijun Wang

Department of Electrical Engineering and Computer Science
The University of Kansas, Lawrence, KS 66045, USA
{sgauch, gwang}@eecs.ukans.edu

*http://www.designlab.ukans.edu/ProFusion.html

Abstract: The explosive growth of the World Wide Web, and the resulting information overload, has led to a mini-explosion in World Wide Web search engines. This mini-explosion, in turn, led to the development of ProFusion, a meta search engine. Educators, like other users, do not have the time to evaluate multiple search engines to knowledgeably select the best for their uses. Nor do they have the time to submit each query to multiple search engines and wade through the resulting flood of good information, duplicated information, irrelevant information, and missing documents. ProFusion sends user queries to multiple underlying search engines in parallel, retrieves and merges the resulting URLs. It identifies and removes duplicates and creates one relevance-ranked list. If desired, the actual documents can be pre-fetched to remove yet more duplicates and broken links. ProFusion's performance has been compared to the individual search engines and other meta searchers, demonstrating its ability to retrieve more relevant information and present fewer duplicates pages. Future developments include analyzing the documents for improved ranking, automatically submitting queries to the most appropriate search engines, and modifying ProFusion to be an information filtering and dissemination system.

1. Introduction

There are a huge number of documents on the World Wide Web, making it very difficult to locate information that is relevant to a user's interest. Search tools such as InfoSeek[InfoSeek] and Lycos[Lycos] index huge collections of Web documents, allowing users to search the World Wide Web via keyword-based queries. Given a query, such search tools search their individual index and present the user with a list of items that are potentially relevant, generally presented in ranked order. However large the indexes are,

still each search tool indexes only a subset of all documents available on WWW. As more and more search tools become available, each covering a different (overlapping) subset of Web documents, it becomes increasingly difficult to choose the right one to use for a specific information need. ProFusion has been developed to help deal with this problem.

2. Related Work

There are several different approaches to managing the proliferation of Web search engines. One solution is to use a large Web page that lists several search engines and allows users to query one search engine at a time. One example of this approach is All-in-One Search Page [Cross]. Unfortunately, users still have to choose one search engine to which to submit their search.

Another approach is to use intelligent agents to bring back documents that are relevant to a user's interest. Such agents [Balabanovic et al. 1995][Knoblock et al. 1994] provide personal assistance to a user. For example, [Balabanovic et al. 1995] describes an adaptive agent that can bring back Web pages of a user's interest daily. The user gives relevance feedback to the agent by evaluating Web pages that were brought back. The agent them makes adjustment for future searches on relevant Web pages. However, these agents [Balabanovic et al. 1995][Knoblock et al. 1994] gather information from only their own search index, which may limit the amount of information they have access to.

A different approach is the meta search method which builds on top of other search engines. Queries are submitted to the meta search engine which in turn sends the query to multiple single search engines. When retrieved items are returned by the underlying search engines, it further processes these items and presents relevant items to the user. ProFusion [ProFusion], developed at the University of Kansas, is one such search engine.

The idea of using a single user interface for multiple distributed information retrieval systems is not new. Initially, this work concentrated on providing access to distributed, heterogeneous database management systems [Arens et al. 1993]. More recently, meta searchers for the WWW have been developed. For example, SavvySearch [Dreilinger] selects the most promising search engines automatically and then sends the user's query to the selected search engines (usually 2 or 3) in parallel. SavvySearch does very little post-processing. For example, the resulting document lists are not merged. MetaCrawler [Selberg et al. 1995][MetaCrawler], on the other hand, sends out user's query to all search engines it handles and collates search results from all search engines. What distinguishes ProFusion from others is that it uses sophisticated yet computationally efficient post-processing.

3. ProFusion

3.1 General Architecture

ProFusion accepts a single query from the user and sends it to multiple search engines in parallel. The current implementation of ProFusion supports the following search engines: Alta Vista [Alta Vista], Excite [Excite], InfoSeek [InfoSeek], Lycos [Lycos], Open Text [Open Text], and WebCrawler [WebCrawler]. By default, ProFusion will send a query to InfoSeek, Lycos, and Excite, but the user may select any or all of the supported search engines. If the user prefers, the system will analyze the user's query, classifying it into a subject or multiple subjects. Based on this analysis, the system will automatically pick the top three search engines that perform best on this subject or these subjects. However the search engines are selected, the search results they return are then further processed by ProFusion. The post-processing includes merging the results to produce a single ranked list, removing duplicates and dead references, and pre-fetching documents for faster viewing and further analysis[1].

3.2 User Interface

ProFusion queries are simple to form; they are merely a few words describing a concept. Online help is available via a Help button that leads users to a page explaining the query syntax, including sample queries. Users need only enter a query and press the "Search" button, however there are several options available which give the user more control over their search. The first option specifies whether or not the user wants to have a short summary displayed for each retrieved item. The benefit of displaying retrieved items without a summary is that a user can more quickly scan retrieved items by title. The second option allows users to manually select the search engine(s) to which their query is sent, or to have the system choose automatically (described in Section 3.1). If the user is selecting the search engines, they may choose any number of search engines from one to all six. When "Automatic Pick Best 3" is selected, the system to selects the best three search engines based on the words in the query. [Fig. 1] shows current ProFusion user interface.

3.3 Duplicate Removal

Since the underlying search engines overlap in the Web pages they index, it is highly likely that they will return some of the same pages in response to a given query. ProFusion attempts to remove these duplicated pages, using a few simple rules. The simplest case is when the identical URL

---

[1]Note: Some of the more computationally expensive features (e.g., pre-fetching and broken link removal) are only available through the private ProFusion interface. They may be added as options on the public page.

has been returned by multiple search engines. Clearly, if two items have exactly the same URL, they are duplicates. More complex rules are necessary to handle the case where the identical page is referenced by slight variations on the same address. For example, the URLs is "http://server/" and is "http://server/index.html" reference the identical page. Handling the previous two cases removes approximately 10 - 20% of the retrieved URLs. However, duplicates may also occur because multiple copies of the same page may exist at different locations. Thus, if two items have different URLs but the same title, they might be duplicates. In this case, we break a URL into three parts: protocol, server, and path. We then use n-gram method to test the similarity of two paths. If they are sufficiently similar, we consider them as duplicates. This appears to work very well in practice, removing an additional 10 - 20% of the URLs, but runs the risk that the URLs point to different versions of the same document, where one is more up-to-date than the other. To avoid this risk, we could retrieve the potential duplicates in whole or in part, and then compare the two documents. However, this would increase network traffic and might be substantially slower. This capability has been developed, and will soon be added as an option.

3.4 Merge Algorithms

How to best merge individual ranked lists is an open question in searching distributed information collections [Voorhees et al. 1994]. Callan [Callan et al. 1995] evaluated merging techniques based on rank order, raw scores, normalized statistics, and weighted scores. He found that the weighted score merge is computationally simple yet as effective as a more expensive normalized statistics merge. Therefore, in ProFusion, we use a weighted score merging algorithm which is based on two factors: the value of the query-document match reported by the search engine and the estimated accuracy of that search engine.

For a search engine *i*, we calculated its confidence factor, CF_i, by evaluating its performance on a set of over 25 queries. The CF_i reflects the number of total relevant documents in top 10 hits and the ranking accuracy for those relevant documents. Based on the results, the search engines were assigned CF_is ranging from 0.75 to 0.85. More work needs to be done to systematically calculate and update the CF_is, particularly developing CF_is which vary for a given search engine based on the domain of the query.

When a set of documents is returned by search engine *i*, we calculate the match factor for each document *d*, M_{di}, by normalizing all scores in the retrieval set to fall between 0 and 1. We do this by dividing all values by the match value reported for the top ranking document. If the match values reported by the search engine fall between 0 and 1, they are unchanged. Then, we calculate the relevance weight for each document *d*, R_{di}, by multiplying

its match factor, M_{di}, by the search engines confidence factor, CF_i. The document's final rank is then determined by merging the sorted documents lists based on their relevance weights, R_{di}. Duplicates are identified during the merging process. When duplicates are removed, the surviving unique document's weight is set to the maximum R_{di} value of all the copies.

3.5 Search Result Presentation

The merge process described in the previous section yields a single sorted list of items, each composed of a URL, a title, a relevance weight, and a short summary. These items are then displayed to the user in sorted order, with or without the summary, depending on user's preference.

3.6 Other Implementation Details

ProFusion is written in Perl and is portable to any Unix platform. It contains one Perl module for each search engine (currently six) which forms syntactically correct queries and parses the search results to extract each item's information. Other modules handle the user interface, the document post-processing, and document fetching. Due to it's modular nature, it is easy to extend ProFusion to additional search engines.

ProFusion's main process creates multiple parallel sub-processes, and each sub-process sends a search request to one search engine and extracts information from the results returned by the search engine. The main process begins post-processing when all sub-processes terminate by returning their results or by timing out (60 seconds in the current prototype).

3.7 Performance Evaluation

We invited every student in our Spring 1996 Information Retrieval class to select a query he/she was interested in. They then were asked to perform a search on that query using each of 9 search engines: the six underlying search engines used by ProFusion (Alta Vista, Excite, InfoSeek, Lycos, Open Text, WebCrawler); ProFusion; and two other meta search engines (MetaCrawler and Savvy Search). Each participant provided relevance judgments for the top 20 retrieved items from each search engine, noting which were broken links and which were duplicates. The performance of each of the search engines was then compared by accumulating the information on the number of relevant documents, the number of irrelevant documents, the number of broken links, the number of duplicates, the number of unique relevance documents, and the precision. Here, precision = number of unique relevant documents divided by total number of documents retrieved (20 documents in this evaluation). The following is a summary of the results from the 12 independent queries,

evaluated by the top 20 retrieved documents (total 240 documents evaluated for each search engine).

Search Engines	Total number of relevant (Σ)	Total number of irrelevant (Σ)	Total Broken links (Σ)	Total number of unique relevant document (Σ)	Average Precision number unique / 240
Single Search Engines					
Alta Vista	108	101	31	99	0.41
Excite	129	104	7	122	0.51
InfoSeek	99	125	16	87	0.36
Lycos	119	104	17	93	0.39
Open Text	72	136	32	54	0.23
WebCrawler	92	130	18	73	0.30
Meta Search Engines					
MetaCrawler	98	118	24	85	0.35
Savvy	127	84	29	112	0.47
ProFusion	142	85	13	134	0.56

Table 1: Performance Comparison

From this table, we see that ProFusion achieved the best average precision of all 9 search engines, since it returned the most relevant documents. We attribute this performance to our sophisticated yet efficient merging algorithm, combined with the removal of duplicates. When more of the documents in the top 20 are unique, there is a better chance that more of them are relevant. ProFusion did a better job in duplicate removal than Savvy and MetaCrawler. ProFusion has 142-134=8 duplicates among 142 relevant documents (5.6%), whereas Savvy Search has 127-112=15 duplicates among 127 relevant documents (11.8%) and MetaCrawler has 98-85=13 duplicates among 98 relevant documents (13.3%). Similar numbers were observed for duplicates among irrelevant retrieved documents. The percentage of broken links retrieved by ProFusion was also lower than any system except Excite.

4. Future Work

Enhancements that are underway include analyzing the retrieved documents to improve the ranking, incorporating user preferences (e.g., do they prefer content-bearing pages which contain mostly text or summary

pages which primarily contain links to further pages), and improving the automatic search engine selection. In addition, we plan to add the ability to automatically rerun searches on a periodic basis, presenting only new or updated URLs to the user. This will provide a personal search assistant/information filtering capability.

Acknowledgments

This work was funded by the University of Kansas General Research Fund and National Science Foundation Awards CDA-9401021 and IRI-9409263.

References

[Callan et al. 1995] James Callan, Zhihong Lu, Bruce Croft (1995). "Searching Distributed Collections With Inference Networks", 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1995

[Voorhees et al. 1994] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird, "The Collection Fusion Problem", in The Third Text REtrieval Conference (TREC-3), NIST special publication 500-225 (D. K. Harman, ed.)

[Balabanovic et al. 1995] M. Balabanovic, Y. Shoham, Y. Yun (1995). "An Adaptive Agent for Automated Web Browsing", Journal of Image Representation and Visual Communication 6(4), December 1995

[Knoblock et al. 1994] A. Knoblock, Y. Arens, C. Hsu (1994). "Cooperating Agents for Information Retrieval", Proceedings of the second international conference on cooperative information systems, University of Toronto Press, Toronto, Canada, 1994

[Arens et al. 1993] Y. Arens, C. Chee, C. Hsu, C. Knoblock (1993). "Retrieving and Integrating Data From Multiple Information Sources", Journal on Intelligent and Cooperative Information Systems, 2(2), 1993, Page 127-158

[Selberg et al. 1995] Erik Selberg, Oren Etzioni (1995). "Multi-Service Search and Comparison Using the MetaCrawler", WWW4 conference, December 1995

[MetaCrawler] MetaCrawler search home page, URL: <http://www.cs.washington.edu:8080/>

[Dreilinger] Daniel Dreilinger, Savvy Search Home Page,
URL: <http://www.cs.colostate.edu/~dreiling/smartform.html>

[ProFusion] ProFusion search home page, URL: <http://www.designlab.ukans.edu/ProFusion.html>

[Sun] Sun Microsystems, Inc., Multithreaded Query Page, URL: <http://www.sun.com/cgi-bin/show?search/mtquery/index.body>

[Cross] William Cross, All-in-one Search Page, URL: <http://www.albany.net/allinone/>

[Alta Vista] Digital Equipment Corporation, Alta Vista Home Page, URL: <http://altavista.digital.com/>

[Excite] Excite home page, URL: <http://www.excite.com/>

[InfoSeek] InfoSeek Corporation, InfoSeek Home Page, URL: <http://www.infoseek.com/>

[Lycos] Lycos Inc., Lycos Home Page, URL: <http://www.lycos.com/>

[Open Text] Open Text, Inc., Open Text Web Index Home Page, URL: <http://www.opentext.com/omw/f-omw.html>

[WebCrawler] WebCrawler home page, URL: <http://www.webcrawler.com/>