To appear in: The Seventh International Conference on Information and Knowledge Management (CIKM '98), Washington, D.C., Nov 3-7, 1998.

Associated Biological Information Retrieval From Distributed Databases

Mousheng Xu, Susan Gauch

Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS 66045

sgauch@eecs.ukans.edu

Abstract

One major goal of bioinformatics is to find ways to quickly offer available information to biomedical researchers. Currently, there are hundreds of biological databases which contain inter-related entries. However, very few entries have been manually cross-referenced, making it difficult and time-consuming for researchers to find related information within and between databases. The system developed under this study – the Bioinformation Search Center (BSC) – to automatically relates database entries and creates links without human involvement. BSC (http://www.itttc.ukans.edu/~mxu/cgi-bin/thesis/bi.cgi) also provides Web-based searching across distributed, disparate biological databases. It enables a researcher to find related data from multiple sources with a single input of either the accession ID of a database entry included in the system, or a Medline ID, or a set of keywords. The Protein Information Resource (PIR), the Human Genome Database (GDB) and the GenBank (GB) are the constituent databases in BSC. BSC can usually find more related information than the constituent database search engines used alone. BSC is a small, efficient system written in Perl and Java to make it easily portable across platforms, and is easy to expand to include more databases.

1 Introduction

1.1 Driving Problem

Today is the era of information, particularly for biological researchers. Because the amount of biological data is exploding dramatically [3], it is increasingly difficult to utilize the rich information available. Bioinformatics is coming to the forefront to address this problem and play a critical role in biological research.

Fueled by the explosion in biological data, the number of biological databases is increasing rapidly. However, these databases are distributed all over the world. This raises the issue of how one can effectively and efficiently draw information from multiple, distributed biological databases. For example, the Gene Bank database (NCBI) and the Protein Information Resource (PIR) database are in Washington DC, the Human Genome Database is in Baltimore, Maryland, the Europe protein database (SWISS-PRO) is in Switzerland, and the Japan International Protein Information Database (JIPID) [10] is in Tokyo, Japan. There are more than 100 biological databases [12] and most of them have independent owners and are setup independently. The growing amount of online information provides new opportunities for researchers, providing them the ability to locate a wide range of information related to their research. However, there is currently no way to effectively and efficiently search for information within this growing collection of heterogeneous distributed database.

1.2 Goal

Suppose a researcher has only one piece of information about some biological object. He may also want to know all the other information available about it. For example, if a biologist is beginning to work with a DNA sequence, he may want to know what gene it is, what the mRNA sequence (if any) is, what protein it produces, how the gene is transcribed into mRNA, how the mRNA is translated into protein, how the protein functions, what disease is related to it, what gene family it belongs to, etc. With the currently available biological databases and tools, a researcher needs to search the relevant databases one by one and to then locate the information of interest within the returned results. This is a very time consuming process. The goal of this project is to create a Bioinformation Search Center (BSC) to allow a researcher to use the information they do know to locate related information all over the world. With a single query containing keywords, a Medline number, or an accession number, the user can use the BSC system (http://www.itttc.ukans.edu/~mxu/cgi-bin/thesis/bi.cgi) to search for information of interest from biological databases distributed all over the world.

1.3 Related Work

One common approach to querying multiple, remote databases is to send the query to the remote databases, get

	PIR	GDB	GB
Structure	text based	relational	text based
Size	~150 k entries	~476 k entries	~2,392 k entries
Methods of Search	PIR unique ID, PIR accession ID, MUID, GB accession ID, GDB accession ID, É (no keyword search)	GDB accession ID, keyword, MUID, Sequence ID (no foreign accession ID search)	keyword, GB accession ID, MUID, author, gene name, journal name, title word, text word,... (no foreign accession ID)
Location & Organization	Washington, DC. National Biomedical Research Foundation	Baltimore, MD. Johns Hopkins University	Bethesda, MD. National Institute of Health

Table 1. Biological Database Statistics.

the results back, do some analysis and arrangement, and then yield the final results to the user. This approach is called *database fusion* or *database merging* [2, 11, 15, 5]. Web meta-search engines such as ProFusion [9] (http://www.profusion.com/) and MetaCrawler [14] (http://www.metacrawler.com/) apply this strategy. A major advantage is that no local database is needed, however search can be somewhat slow since all queries must be sent out over the Internet to the remote databases. This strategy is not appropriate for us because there is no consensus searching method over all biological databases. For example, both GDB and GB offer keywords search, but PIR does not. In addition, this approach doesn't provide the ability to build cross-references between database entries, a key goal of our system.

To support distributed search from multiple databases, a global schema over “exported schemas” from participating distributed databases could be provided [14]. With this approach, a global query is mapped into local queries and executed at local participating databases. This method requires a consensus schema design, which may exist someday for distributed biological databases, but there is no such global schema today. Even with such a global schema, because of the lack of consensus searching methods, it still might not be possible to map a global query into appropriate local queries.

An alternative approach uses spiders [13] or data file exchanges [1] to collect all the information into one central repository which processes all queries. This approach creates many problems for our application due to the amount of data to be transferred and the amount of data to be stored and searched. Also, the lack of a global schema makes it difficult to create a single database schema capable of storing the information collected from different databases

1.4 Approach

Our approach is a hybrid between these two extremes of meta-search and centralized search. We retrieve a small but critical portion of information from each database and store that information, creating a miniature local version from part of the distributed databases. This local information uniquely identifies each original database entry and is also used to relate database entries to each other, within and across databases. When a user enters a query to BSC, the search engine searches the local database and presents the matching entries. Since the entry inter-relationships are pre-calculated and stored in a local database, the user can find more information in less time. Since the information returned includes URL pointers to the original database entries, the user can quickly access the full entry by clicking on the associated URL.

The critical issue is how to automatically relate data entries within and among different databases. One obvious idea is to use keywords. Since most biological database Web front ends offer keyword search, a simple approach would be to provide a meta-search front end that broadcasts the user’s keywords to the remote databases and fused the search results in some way. The difficulty with this approach is that “keywords” mean different things in different databases. Additionally, the set of keywords used to index one database entry may be entirely different from those used to index related entries in a different database. The effect of synonyms on information retrieval is a well-known problem that is compounded by the fact that different databases may employ different criteria for keyword selection.

We employ two basic approaches: 1) find something in common for database entries that identifies them as surely related, for example, the Medline ID. A Medline ID is the unique identifier for a published article stored in the Medline database. Most biological articles are in Medline and have their unique Medline ID’s. If two database entries have the same Medline ID, we can be almost certain that they are referred to in the same Medline article and are thus related; 2) construct a set of keywords for each database entry using a consistent set of rules. The keywords for a database entry come from the titles of the article (if there is an article) for the entry, plus some other carefully selected fields (generally, the keyword field included in individual databases). The original keywords are kept for obvious reasons. By adding words from the title, the keyword set for each entry is expanded, increasing the number of database entries that may share a keyword. As always, whenever you increase the inter-relationships, you must consider the effect on accuracy. Since titles are chosen carefully by the author, we believe them to be an accurate source of additional keywords. Once the keywords are selected, standard information retrieval techniques such as stemming [6] and term weighting [6] are used to improve the quality of the match.

	PIR	GDB	GB
Total Number of Entries (till 2/5/98)	151,473	476,875	2,392,288
Insertions/Day	42.6 (0.028%)	29.8 (0.0062%)	4,023 (0.17%)
Modifications/Day	?	25.0 (0.0052%)	0
Deletions/Day	0	0	0

Table 2. Database Daily Update Rate.

2. Implementation

2.1 Information Sources

There are more than 100 biological databases distributed all over the world. To study the feasibility of our approach, we will focus on a proof of concept using three major databases: 1) GenBank (GB), the largest biological database in the world, which contains information about DNA, RNA and protein sequences for all species; 2) Human Genome Database (HGD), which is the major component of the world famous Human Genome Project; 3) Protein Information Resource (PIR), the USA protein sequence database (see Table 1). These databases reside in different locations, and belong to different organizations. They each offer a rich collection of searching methods, but there is no consensus searching method across all databases. For example, PIR does not provide keyword search whereas GDB and GB do not provide foreign accession ID search (i.e., the ability for remote users to retrieve a unique database entry using its accession ID).

2.2 Preprocessing

Four pieces of information are collected for each database entry: 1) the accession ID for the original database; 2) the Medline ID (if there is one); 3) the keywords (if any); and 4) the title (if there is one). Next, we inter-relate entries based on the above information. More information (e.g., species, author) could be collected and might be helpful, but the space needed for the local database would be significantly increased.

We preprocess the keyword set to remove unimportant information (e.g., frequently appearing words) and improve the number of inter-relationships found by standardizing the word forms (e.g., stemming the keywords). We remove the typical English stopwords such as "the" and "an" and most words of three characters or less. In addition, other words such as "adult", "amino", "acid", "human", and "chromosome" occur so frequently that they are not good discriminators for our database, and they too are removed. We do keep short words that are important because they represent gene names, protein names, gene family names, etc. For example, "MT" stands for metallothionein, "IX" in "Human Factor IX" distinguishes human factor IX from all other human factors. These words are easily identified since they appear in all capital letters.

Words with the same "stem" often are highly related, e.g., "bind", "binds", and "binding" all have the same meaning, as do words "develop", "development" and "developing". Indexing documents by word stems rather than the full words has been found effective where documents and queries are short [6]. Since the keyword sets are typically fairly small, we use Porter's stemming algorithm [6] to stem the keywords to increase the likelihood of finding relationships among the database entries.
Once the pre-preprocessing is complete, we create four inverted files to provide efficient mapping from keywords to accession IDs and vice versa and Medline IDs to accession IDs and vice versa.

2.3 Query Processing

Three types of search are offered in the BSC system:

1) Medline ID search. Given a Medline ID, the accession IDs of all entries that share the same Medline ID are returned.

2) Keyword search. Given a keyword or set of keywords, the accession IDs of all entries that share some or all the keywords should be returned, rank ordered by the degree of similarity between the query keyword set and the entries' keyword sets.

3) Accession ID search. Given an accession ID, the system is responsible for finding out all accession IDs related to it. The automatically generated "links" between accession IDs are based on the Medline ID and keywords. If two accession IDs have the same Medline ID, which means they are from the same article, then they are very likely related. Similarly, if two accession IDs have the same or similar keyword sets, they may also be related. When a user enters an accession ID, the system first finds all the Medline IDs and all the keywords associated with this accession ID, and then use the algorithms applied for case 1) and 2) above to find all related accession IDs.

2.4 Information Update

To keep information up to date, the database needs to be updated frequently. Since the constituent databases are themselves updated daily, we also update BSC daily. The databases included in the system are all extremely large. For a relational database, retrieving information through an SQL query is fast and thus *de nova* update of the whole database can be done within a few hours. If the information is retrieved through an httpd socket, *de nova* update may take days. However, only the information that has been changed since the last update is needed, which keeps the update process reasonably efficient.

There are basic three types of database updates. One is modification of existing entries, another is deletion of existing entries, and the last is insertion of new entries. Insertion happens more often than modification and deletion (see Table 2). GB allows insertion only, not modification or deletion (personal communication, Dr. John Garavelli, PIR). Everything in the database is final once it has been entered. In PIR, no accession ID is ever deleted. Although an accession ID may be merged with other accession IDs to refer to the same database entry, it is never deleted and is always searchable on its own. Modification is allowed, but happens only infrequently.

In general, updating each database takes from a few minutes to a few hours, depending on the database. PIR and GB

Database	Original Size (MB)	BSC Size (MB)	BSC/Original
PIR	47.9	5.4	11.3%
GDB	10,600*	18.6	0.18%
GB	1,453	29.3	2.0%

*** size of SQL database dump.**

Table 3. Database Size Comparisons.

Medline ID	BSC time (Results)	PIR time (No.)	GDB time (No.)	GB time (No.)
94065665	2.01s (0+0+1=1)	0.91s (0)	1.55s (0)	1.17s (1)
91286212	2.32s (0+0+1=1)	1.45s (0)	1.57s (0)	1.57s (1)
90272652	1.92s (0+0+1=1)	1.07s (0)	1.71s (0)	1.21s (1)
95021230	1.66s (0+0+1=1)	3.80s (0)	3.55s (0)	1.71s (1)
88015565	2.26s (1+0+1=2)	3.41s (1)	1.99s (0)	1.01s (1)
Avg. time	2.03s	2.12s	2.07s	1.33s
Avg. entries	1.2	0.2	0	1
Avg. relevant entries	1.2	0.2	0	1
Avg. time/entry	1.68s	10.6s	N/A	1.33s
Avg. time/relevant entry	1.68s	10.6s	N/A	1.33s

Table 4. Comparison of Medline ID search from BSC, PIR, GDB and GB.

allow incremental updates which speeds the process, however GDB must be downloaded in its entirety. PIR is updated through an httpd socket, and it may take a few minutes to 1 or 2 hours, depending on the number of insertions. GDB is updated through a database client/server connection, which is relatively fast. It usually takes about 4 hours to process all relevant information of GDB. Updating of GB is also through an httpd socket, and it usually takes approximately 6 hours. Note that these updates can be done simultaneously.

As shown in Table 3, the disk space needed to store the information for the BSC system is rather small, compared to the size of the original databases. This is especially true for GDB and GB, the largest databases.

3. Evaluation

The main purpose of this project is to develop a system that allows a researcher to easily find related information from multiple distributed databases. Cross-referencing is currently done manually (by database annotators or the researchers themselves) for all databases, which is a time-consuming and dollar-consuming process. One of the strengths of the BSC system is that cross-referencing is based on some basic, readily available and natural characteristics (Medline ID and/or title and/or keywords) of database entries. It is independent of human involvement in finding cross-references, and is done automatically and dynamically.

Tables 4 through 6 show the comparison of time and number of database entries returned by BSC from PIR, GDB

User input keywords	BSC time (Results: PIR+GDB+GB) (Relevant in top 10 results)	PIR time (Results) (Relevant in top 10 results)	GDB time (Results) (Relevant in top 10 results)	GB time (Results) (Relevant in top 10 results)
Electrophoresis	6.05s (23+218+231=472) (10+10+10)	(no keywords search)	9.53s (2) (0)	6.91s (334) (10)
Chloride channel	5.94s (424+84+214=722) (10+5+10)	(no keywords search)	7.65s (193) (10)	11.97s (294) (6)
Dopa decarboxylase	4.88s (227+51+645=923) (8+9+10)	(no keywords search)	5.44s (28) (6)	12.43s (103) (6)
Borrelia	2.41s (66+0+860=926) (10+0+10)	(no keywords search)	8.04s (0) (0)	28.72s (1295) (10)
Presenilin 2	8.06 (4+31+30=65) (3+0+6)	(no keywords search)	6.36s (25) (5)	4.93s (4) (2)
Avg. time	5.47s	N/A	7.40s	12.99s
Avg. entries	622	0	50	406
Avg. relevant entries	22	0	4	7
Avg. time/entry	0.0088s	N/A	0.15s	0.032s
Avg. time/relevant entry	0.25s	N/A	1.85s	1.86s
Avg. precision (top 10)	(82%+48%+92%) /3 = 74%	0%	42%	68%

Table 5. Comparison of keyword search from BSC, PIR, GDB and GB.

and GB for three different search methods shown in order of search method breadth of scope: Medline ID, then keywords, and finally accession ID. The time taken overall, the average number of relevant documents returned and the average precision (the number of relevant documents returned / the number of documents returned) are shown.

3.1 Medline ID search

Table 4 shows the result of querying using 5 Medline IDs randomly selected from GB. All results found are relevant, but very few new entries are located this way. However, the results demonstrate the potential of associating information across databases using Medline IDs. GenBank (GB) returns one associated entry each time, which is also found by BSC. Although BSC returns only this one item the researcher still benefits. In general, they would not know *a priori* which of the databases would contain information indexed by a particular Medline ID. BSC's efficiency is demonstrated since if only one item is found, the researcher is saved time by using BSC rather than using each database in turn. In particular, the query processing time for BSC is 2.03 seconds on average versus an average of 5.52 seconds to search each database in turn. More importantly, the time to connect to the three sites and enter the same query three times is reduced since they only need one query at one site. The final Medline ID search demonstrates BSC's effectiveness. In that case, BSC returns an additional entry from PIR as well as the entry from GB.

3.2 Keywords search

BSC's effectiveness and efficiency are dramatically demonstrated through 5 keywords queries randomly selected from queries submitted to our site by end-users. Table 5 shows that, on average, BSC returns 622 entries from all three databases. Queried directly, PIR does not support keyword search, but BSC can provide that capability since it creates keywords for PIR entries from the title field. In addition, because it uses stemming, BSC can also often locate more items in GDB and GB than the keywords search those databases provide directly. Because the local database is comparatively small, BSC search is quick, taking 5.47 seconds on average, compared to 7.40 seconds for GDB, and 12.99 seconds for GB.

However, much more important than finding more information quickly is BSC's ability to find more *relevant* information quickly. We measure the accuracy of the query results by calculating the precision (percentage of relevant entries) for the top 10 results for each search engine. The results for BSC show that it has higher precision for each constituent database (82% versus 0% for PIR, 48% versus 42% for GDB, 92% versus 68% for GB) and even a higher precision in its top 30 entries (74%) than any constituent

Accession ID	**BSC time (ASRL* No.) (PRL** No.)**	**PIR time (No.)**	**GDB time (No.)**	**GB time (No.)**
PIR:PC1072	5.53s (0+0+0=0) (350+0+702=1052)	2.30s (1)	(no foreign accession ID search)	(no foreign accession ID search)
PIR:S21712	14.15s (0+0+0=0) (1636+357+2256=4249)	1.91s (1)	(no foreign accession ID search)	(no foreign accession ID search)
PIR:A01985	6.67s (0+0+0=0) (102+84+1429=1615)	1.74s (1)	(no foreign accession ID search)	(no foreign accession ID search)
GDB:185423	9.74s (0+1+0=1) (0+91+0=91)	1.13s (0)	5.61s (1)	(no foreign accession ID search)
GDB:175847	6.57s (1+3+2=6) (90+57+262=409)	0.90s (0)	7.33s (1)	(no foreign accession ID search)
GDB:9787144	4.67s (0+5+0=5) (0+64+1=65)	4.01s (0)	9.08s (1)	(no foreign accession ID search)
GB:L31381	9.16s (0+4+14=18) (518+327+875=1720)	2.39s (0)	(no foreign accession ID search)	4.85s (1)
GB:M25781	8.53s (1+0+1=2) (322+458+3075=3855)	1.49s (1)	(no foreign accession ID search)	10.30s (1)
GB:AF030383	6.26s (0+0+0=0) (45+233+187=465)	1.08s (0)	(no foreign accession ID search)	7.57s (1)
Avg. time	7.92s	1.88s	(7.34s)	(7.57s)
Avg. entry	1525	0.44	(1)	(1)
Avg. relevant entry+	3.6	0.56	0.33	0.33
Avg. time/entry	0.0052s	4.24s	(7.34s)	(7.57s)
Avg. time/relevant entry	2.2s	4.24s	(7.34s)	(7.57s)

***Almost Surely Relevant Link**
****Possibly Relevant Link**
+Relevant == Almost Surely Relevant Link

Table 6. Comparison of accession ID search from BSC, PIR, GDB and GB.

database provided when only their top 10 entries were evaluated. Thus, not only does BSC provide more information than any one system, it provides more relevant information and does so efficiently.

3.3 Accession ID search

The results of querying BSC with 3 randomly selected accession IDs from each database are shown in Table 6. Three accession IDs are selected for each database. GDB and GB only offer accession ID search for their own databases, but do not offer foreign accession ID search. Accession ID search is the combination of Medline ID search and keywords search. When an accession ID is input into the BSC system, the Medline IDs and keywords associated with this accession ID are retrieved, then each Medline ID is used for a Medline ID search, and all the keywords are used together as a single keyword search.

GDB and GB do not offer search by foreign accession IDs, but PIR offers search for both GDB accession ID and GB accession ID. On average, a single accession ID search in BSC results in 1500 entries of which 3.6 are related by Medline ID and almost surely relevant and the rest are related by keywords and are quite possibly relevant. In contrast, PIR returns 0.44 entries on average and GDB and GB return one entry if the accession ID belongs to their own database, none otherwise, for an average of 0.33 entries.

4. Discussion and Future work

The primary goal of this work was to produce a system that could automatically cross-reference information from multiple biological databases and use those cross-references to provide fast, effective search for researchers. To achieve this, we designed and implemented the Bioinformation Search Center (BSC), which extracts relatively small amounts of information from remote databases. This information is used to link database entries within and between databases through their keywords and Medline IDs. The replicated database feature has advantages of availability and reduced response time in network communication [4]. We have developed a Web-based search page to access BSC, and to date it has received over 1,000 queries. Through our evaluation studies, we determined that BSC provided more information than any individual database and, on average, more information than the three systems would have provided together. The results returned were quite accurate -- BSC produced higher precision numbers for the limited result sets we examined than the constituent databases. Finally, BSC produced its results quickly.

The BSC system is the first available Web search engine that can automatically and dynamically relate entries from independent biological databases without human involvement in cross-referencing. It is a fast, effective, easy to expand and maintain, and takes relatively little disk space. However, we could improve BSC by increasing the quantity of the cross-references it stores by also retrieving any existing manual cross-references from the databases. We could also automatically create sets of similar words and use them for query expansion to increase the cross-references [8, 7]. We could make many user interface improvements, particularly by adding Boolean search capabilities at our user's requests and by improving the summary information provided with the results. Once we preprocess the titles and keywords, we discard the original information to conserve disk space. Therefore, BSC prints out the tokenized, stemmed keywords associated with an entry as the summary for that entry and it is not always meaningful for the end-user. The current BSC system applies exact word matching, although the words are tokenized and stemmed. Finally, we could examine ways to increase the time necessary to update the database, the major drawback to this approach.

REFERENCES

[1] Andleigh, P.K and Gretzinger, M.R. (1992) *Distributed Object-Oriented Data-Systems Designs*, PTR Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

[2] Arens, Y., Knoblock, C.A. and Sheng, W.M. (1996) "Query Reformulation for Dynamic Information Integration," *Journal of Intelligent Information Systems*, **6** (2/3), 99-130.

[3] Benton, D. (1996) "Bioinformatics – Principles and Potential of a New Multidisciplinary Tool," *Tibtech*, **14**, 261-272.

[4] Buretta, M. (1997) in *Data Replication: Tools and Techniques for Managing Distributed Information*, Wiley Computer Publishing, John Wiley & Sons, Inc., New York, 6-9.

[5] Callan, J., Lu, Z. & Croft, W.B. (1995) "Searching Distributed Collections with Inference Networks," *18th Ann. International ACM SIGIR Conf.*, Seattle, WA, ACM Press, 21-28.

[6] Frakes, W. B., Baeza-Yates, R. (1992) *Information Retrieval, Data Structures & Algorithms* (Ed: Verde, K., Goodwin, B., Doench, G., and Papanikolaou, S.), Prentice-Hall International (UK) Limited, London.

[7] Gauch, S., and Chong, M. K. (1996) "Automatic Word Similarity Detection for TREC-4 Query Expansion," *The Fourth Text Retrieval Conference* (Ed: D. K. Harman), NIST publication 500-236, U.S. Government Printing Office, Washington, 121-130.

[8] Gauch, S., and Wang, J. (1997) "Corpus Analysis for TREC-5 Query Expansion," *The Fifth Text Retrieval Conference* (Ed: E. M. Voorhees and D. K. Harman), NIST publication 500-238, U.S. Government Printing Office, Washington, 537-546.

[9] Gauch, S., Wang, G. and Gomez, M. (1996) "ProFusion: Intelligent Fusion from Multiple, Distributed Search Engines," *Journal of Universal Computer Science*, Springer-Verlag, **2** (9), September, 637-649.

[10] George, D. G., Dodson, R. J., Garavelli, J. S., Haft, D. H., Hunt, L. T., Marzec, C. R., Orcutt, B. C., Sidman, K. E., Srinivasarao, G. Y., Yeh, L. L., Arminski, L. M., Ledley, R. S., Tsugita, A., and Barker, W. C. (1997) The Protein Information Resource (PIR) and the PIR-International Protein Sequence Database, *Nucleic Acids Research*, **25** (1), 24-27.

[11] Hawking, D., Thistlewaite, P., and Bailey, P. (1997) "ANU/ACsys TREC-5 Experiments," *The Fifth Text Retrieval Conference* (Ed: E. M. Voorhees and D. K. Harman) , NIST publication 500-238, U.S. Government Printing Office, Washington, 359-376.

[12] Karp, P. D. (1996) "Database Links are a Foundation for Interoperability," *Tibtech*, **14**, 273-279.

[13] Koster, M. (1995) "Robots on the Web: Threat or Treat?" *ConneXions*, **9** (4), April.

[14] Missier, P., Rusinkiewicz, M., and Silberschatz, A. (1996) "Database Reengineering and Interoperability," (Ed: Cheung, T., Fong, J., and Siu, B.), Plenum Press, New York and London, 337-348.

[15] Selberg, E. and Etzioni, O. (1995) "Multi-Service Search and Comparison Using the MetaCrawler," *The Fourth World Wide Web Conference (WWW4)*, 195-208.

[16] Voorhees, E. M. (1996) "Siemens TREC-4 Report: Further Experiments in Database Merging," *The Fourth Text Retrieval Conference* (Ed: D. K. Harman) , NIST publication 500-236, U.S. Government Printing Office, Washington, 121-130.