MERGING MULTIPLE SEARCH RESULTS APPROACH FOR META-SEARCH ENGINES

By

Khaled Abd-El-Fatah Mohamed

B.S-------- Cairo University, Egypt, 1995

M.A-------Cairo University, Egypt, 1999

M.A------- University of Pittsburgh 2001

Submitted to the Graduate Faculty of

School of Information Sciences in Partial Fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2004

UNIVERSITY OF PITTSBURGH
INFORMATION SCIENCES

This dissertation was presented by

Khaled Abd-El-Fatah Mohamed

It was defended on

Janauary 29, 2004

and approved by

Chris Tomer, PhD, Associate Professor, DLIS
Jose-Marie Griffiths, PhD, Professor, DLIS
Don King, Research Professor, DLIS
Amy Knapp, PhD, ULS
Dissertation Director: Chris Tomer, PhD, Associate Professor

MERGING MULTIPLE SEARCH RESULTS APPROACH FOR META-SEARCH ENGINES

Khaled A. Mohamed, PhD

University of Pittsburgh, 2004

Meta Search Engines are finding tools developed for enhancing the search performance by submitting user queries to multiple search engines and combining the search results in a unified ranked list. They utilized data fusion technique, which requires three major steps: databases selection, the results combination, and the results merging.

This study tries to build a framework that can be used for merging the search results retrieved from any set of search engines. This framework based on answering three major questions:

1. How meta-search developers could define the optimal rank order for the selected engines.
2. How meta-search developers could choose the best search engines combination.
3. What is the optimal heuristic merging function that could be used for aggregating the rank order of the retrieved documents form incomparable search engines.

The main data collection process depends on running 40 general queries on three major search engines (Google, AltaVista, and Alltheweb). Real users have involved in the relevance judgment process for a five point relevancy scale. The performance of the three search engines, their different combinations and different merging algorithm have been compared to rank the database, choose the best combination and define the optimal

merging function.

The major findings of this study are (1) Ranking the databases in merging process should depends on their overall performance not their popularity or size; (2)Larger databases tend to perform better than smaller databases; (3)The combination of the search engines should depend on ranking the database and choosing the appropriate combination function; (4)Search Engines tend to retrieve more overlap relevant document than overlap irrelevant documents; and (5) The merging function which take the overlapped documents into accounts tend to perform better than the interleave and the rank similarity function.

In addition to these findings the study has developed a set of requirements for the merging process to be successful. This procedure include the databases selection, the combination, and merging upon heuristic solutions.

To my family in Egypt who support and raise me

AND

To my little family here,

my wife, my daughter Nada, and my new born Ahamd,

who bear of being away from homeland with me.

Acknowledgments

Many people have supported this dissertation and must be recognized and appreciated for their efforts and time. First and foremost is my dissertation committee. I would like to thank especially Dr. Christinger Tomer who has provided me from the beginning of choosing the research topic to the final examination arrangement excellent advice and support as advisor. Professor José-Marie Griffiths has contributed many invaluable insight that influenced every part of this dissertation and provided good suggestion to deal with them. Mr. Donald King has identified a couple of major problems associated with writing the correct statistical formula and phrasing the hypotheses in a correct statistical format. Dr. Amy Knapp has helped me putting things together to make final product more readable.

The completion of this dissertation is also attributed to the support and cooperation from numerous graduate students from the University of Pittsburgh who voluntarily provided their research topic to be used as queries and conducted relevance judgments for the retrieved set of documents. Special thanks should be given to Professor Edie Rasmussen who helped me in preparing the proposal of this dissertation and provided many good advices and papers related to the topic. Special recognition should also be given to Aaron Brenner my colleague and friend in the Digital Research Library in the University of Pittsburgh. He has been helping me in a number of different ways in preparing and debugging the Perl scripts. Special thanks and recognition should be given to Professor Hishmat Kasm who has a great influence on my success in this career. Special recognition should also be given to professor Elain Robinson who helped me in conducting the statistical test.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE: INTRODUCTION

1. Introduction:

Meta-search engines are searching tools that have mainly developed to enhance the retrieval performance of the World Wide Web finding tools. They are based on data fusion technique which requires three major steps including: Selecting the most comprehensive databases and ranking them properly, combining the retrieved results then merging them in a single list of documents using the most appropriate merging algorithm. This study tries to build a framework for meta-search engines developers that can be utilized in achieving these steps**.**

1.1. The World Wide Web

The Internet, and particularly the World Wide Web has become one of the major features of the current information age because of the huge amount of information and the enormous number of users who get access to this information. Although the Web plays a significant role in disseminating information, there is a lack of centralized control or authority statistics in terms of number of web pages, web sites, and users, even though the World Wide Web grows by exponential rate at 50 % a year, which represents an ever-increasing proportion of human knowledge is becoming available on line (Bokor, B, 2002).

The Internet domain survey estimates that the Internet host machines increase from 1,313,000 in January 1993 to 147,344,723 in January 2002, (Internet Domain Survey, 2002) which means that the number of host machines increased in nine years by about 147 %. This estimation provides also an indication about the incredible increase in the number of web sites and web pages. The Internet users also incredibly increase, for

example the CommerecNet estimates the total number of the users are about 490 million by the end of 2002 and expected to be over 765 million by the end of 2005. (CommereceNet/ Nielsen, 2002[1]).

Lawrence and Giles (1999) estimated that the number of publicly indexable web pages is about 800 millions page in 1999, encompassing about 15 terabytes of information or about 6 terabytes of text after removing HTML tags, comments, and extra white-spaces. A more recent estimation of the number of web sites indicated that this number exceeded two billion sites and the number of web pages is much larger than this number (Bokor, B, 2002).

The World Wide Web has become the major hyperspace for getting access to the digital information through useful information services. Although the Web facilitates many applications and information services, e.g. E-mail, FTP (File Transfer Protocol), electronic publishing, E-commerce, distance learning, Tele-conferences, etc., the primary use of the web after the E-mail is for finding information. However, finding a specific piece of information among such incredible amount of information would be impossible without powerful tools that automatically browse and search the web (Khan & Locatis, 1998; Wang, Hawk, & Tenopir, 2000).

Gordon and Pathak (1999) identify four major methods for finding information on the web, which include: (1) Using a known URL, (2) Using Hypertext links to navigate from a web page to another web page, (3) Narrowcast services or Portals which push web pages to users according to their particular profiles, (4) Search engines which allow users to search the web exploring traditional and advanced information retrieval techniques.

[1] The number of the Internet population compiled from different sources and collected in the CommerceNet site (www.commerce.net/research/stats/wwstats.html

While the other three methods of locating information are important, Lawrence and Giles (2002) estimated that 85 % of web users use search engines to find their information needs. Jansen and Pooch (2001) indicated that 71 % of web users' access search engines to reach other web sites. They also stated that user's rate searching as the most important activity conducted on the Internet. However, web search engines are limited in terms of coverage, currency, interface options, how well they retrieve relevant information and how well they rank the relevance of the results. In short, although the search engines limitations, they are indispensable for searching the web. They utilize a variety of relatively advanced IR techniques, and there are some peculiar aspects of search engines that make searching the web different than conventional information retrieval (Gordon & Pathak, 1999; Seamton & Crimmins, 1997; Lawrence & Giles, 1999).

1.2 Web Based Search Engines:

Lancaster (1998) indicated that search engines operate by building indexes to the network resources by extracting of words or phrases from the text itself. In principal, these searchable files are nothing more than the conventional inverted files used to facilitate information retrieval ever since random access began to replace serial searching of records in the early 1960s but with more sophisticated capabilities powered by the software and hardware improvements.

Schwartz and Pu (1998) stated that search engines began to appear in 1994, most of them started as research projects undertaken by graduate students, faculty, and system staff. Bradley (1998) estimated that by January 1998 there were at least 2000 searching tools available covering both general and specific purposes. As of August 2002, there were at

least 25 general purpose search engines (see Search Engine Watch[2]), as well as numerous of special purpose search engines, while the Big Search Engine Index generate a list of 912 search engines available for use by March, 20, 2003. (see Big Search Engine Index[3])

Web searching tools include three major categories: directories, which are no more than subject catalogs or classification to the web, Yahoo is considered one of the most famous example of this category; search engines, which are indexes for the web and sometimes they add to their indexes subject directories (i.g Google, AltaVista, Excite, Alltheweb); and Meta-Search Engines, which run the same query in more than one search engine, so they do not include any databases. (i.g. Ixquick, Vivisimo, QbSearch, and ProFusion).

Search engines indexes and subject directory catalogs vary in the number of pages they contain, with 390 million and 1500 million pages at the small and large ends of the scale, respectively. Most major search engines contain 500 to 600 million pages as reported in December 2001 (Sullivan, 2002). Lawrence and Giles (2002) found that the average of six major search engines (Altavista, HotBot, Excite, Infoseek, Lycos, and Northern Light) varies significantly and none of them cover more than about a third of the estimated publicly indexable web. They also found that the coverage of search engines has decreased substantially since December 1999, with no engine indexing more than about 16 % of the estimated size of the publicly indexable web. As December, 2001, Google reported indexing, directly and indirectly, about 2 billion pages, although this includes documents in formats such as PDF and Microsoft Office (Sullivan, 2001).

---

[2] Search Engine Watch: http://www.searchenginewatch.com

[3] Big Search Engine Index: http://www.search-engine-index.co.uk/

There are four major criteria that differentiate one search engine from another. These criteria include: (Gabe, 2002)

1. Size (the number of sites or pages indexed).
2. Speed (how fast the engine can find the information requested).
3. Relevance (how many of the "hits" are relevant to the actual query).
4. Update rate (How current is the information contained in their databases).

Search engines consist of three main components: a robot or spider, which crawls the web and captures new web pages; database which include serial files, indexes, and inverted files for the captured web pages; and agent which perform the search process. In general search engines work as follows: first they build their own databases by visiting web sites or web pages on regular basis and indexing those that are appropriate to be included in the databases, then when users' submit a query to the search engine, the engine match the query terms with the database with support of sophisticated searching algorithms (which vary from one search engine to another). Finally the search engine retrieves a list of items ranked according to their relevancy for the query terms.

Search engines use different algorithms to define page relevancy and rank order for a particular query. Dwork et al. (2001) indicated that few years ago, query term frequency was the single main method in ranking web pages; since the influential work of Kleinberg, and Brin in page link analysis has come to identified as a very powerful technique in ranking web pages and other Hyperlinked documents. Several other methods have been added, including anchor-text analysis, page structure (headers, etc.) analysis, the use of keywords listings and the URL text itself, etc.

Griffith and King (2000) noted that the IR landscape is changing dramatically with the growing influence of the web search services. They also stated that the ease of use of

these searching services may come at a significant loss in quality without users being aware of the risk.

Maze et al. (1997) states that, while the exact algorithm for indexing, retrieving, and ranking web pages are commercial secrets, the companies publicize some general information about these techniques, and experiments can reveal other details.

Since search engines employ different search algorithms, even if several engines process the same query against the same set of documents, the way search engine rank those documents may differ. Lighton and Srivastava (1999) stated that in most engines, a web page will be highly ranked if it frequently uses the same word or phrases found in the query. The appearance of these words in a page title, heading, or early in its text tends to raise the relevance score of the document.

Although search engine providers claim that they capture and retrieve all the relevant pages and ranking the results from highly relevant to less relevant to satisfy the user requests, there is no enough evidence that search engines retrieve all the possible relevant documents because no one single search engines covers all the possible indexable web pages as indicated by Lawrence & Giles (2002).There isn't also enough evidence that search engines retrieve relevant documents according to the user queries. To improve search engines deficiencies, some search engine providers allow the same search query to be submitted to other search engines (see Altavista, Lycos, HotBot) in order to provide more comprehensive results and retrieve relevant pages. Another solution is found in utilizing a data fusion approach by combining the retrieved results generated by using multiple document or query representations or multiple retrieval strategies . In the context of the web these tools are known as meta-search engines, which

utilize data fusion for merging results retrieved from different sources. (Selberg & Etzioni, 1995; Savoy, La Clave & Varjitoru, 1996; Tsikrika & Lalmas, 2001).

As of September 5, 2002, there were 46 major meta-search engines generated by the big search engine index (see Big Search Engine Index).

1.3 Met-Search Engines

Meta search engines include a list of the most famous and comprehensive search engines. In general meta-search engines (e.g. MetaCrawler, SavvySearch, ProFusion) merge results from multiple search systems into a single ranked list using data fusion strategy and merging function. In addition, some form of query translation is necessary, to interact between different search systems and utilize the value of interoperability of the merging function (Dwork, 2001).

Fusion and aggregation of information are major problems for all kinds of knowledge based systems, from image processing to decision making. The two words are often used for the same general purpose: how to use simultaneously pieces of information provided by several sources in order to come to a conclusion or a decision. Nevertheless, there are two general approaches to this scheme, depending on the problem to be dealt with. The first one corresponds to the aggregation of preferences given by several individuals of a group or the aggregation of criteria to satisfy in order to make a decision. The second approach concerns the fusion of evidence provided by several sources. In many cases, the available information is imperfect. Several methodologies are useful to manage such imperfect information. Among the most important ones are probability theory, evidence theory, fuzzy set theory and possibility theory (Bernadette Bouchon-Meunier. Ed., 1998).

Meta-search engines face two serious challenges: (1) Choosing the best combination of search engines to retrieve the most relevant sets. This process is know in the IR literature as database selection, which includes also database ranking. (2) Choosing the appropriate method to aggregate the rank order of the retrieved sets. This process is known as result merging, fusing, or rank aggregation.

Smeaton & Crimmins (1997) identify two major methods of results fusing: (1) Fusing the results from one source using different search strategies; (2) Fusing the results from different sources using the same search strategy (Yuwono & Lee, 1997; Yang & Zhang, 2000).

Over the past few years, many meta-searching tools have been developed. Some of the best efforts have been surveyed to collect lists of these efforts (See Search Engines Watch, Big Search Engines Index, and Search Engines.com[4]). As of February 5, 2003, there were 46 major meta search engines generated by the Big Search Engine Index. Some meta-search engines display a list of search engines that candidate to be searched (e.g. IXquick, VIVISIMO, QPsearch, etc.,). Others do not show which search engines are queried (e.g. Dogpile, Mamma, Profusion). They provide a list of search engines to submit the queries by default. However the search engines list could be reached by using the advanced or the customized search options.

Meta-search engines use different techniques for fusing the search results. For example, Dogpile [5] does not merge the search results together. Instead, it keeps the results from each major search engine separate from the others. Ixquick[6] and Mamma[7]

[4] Search Engines COM: http://www.searchengines.com
[5] Dogpile: http://www.dogpile.com/index.gsp
[6] Ixquick: http://ixquick.com
[7] Mamma: http://www.mamma.com

merge the retrieved results according to the overlapping documents. The major assumption here is that if the page appears in multiple top ten lists is likely to be very relevant (Tsikrika & Lalmas, 2001). Ixquick upholds the democratic ideal of one search engine, one vote, even when the search engine mentions the same document often in its top ten. It also shows the document rank in the different engines (Tsikrika & Lalmas, 2001). MetaCrawler[8] utilizes the document retrieval score and fuses the rank position, so duplicate document have their score summed, and documents are penalized if they are not overlapped (Yang & Zhang, 2000). ProFusion[9] uses a weighted score merging algorithm, similarly to where ranking is defined using both the initial retrieval score assigned by the search engines and the score expressing the quality of that document and the quality of that search engines. The major problem here is that not all search engines retrieve similarity score with document and search engines using different weighting schemes to calculate the document similarity value (Callan, Lu, & Croft, 1995; Gauch, Wang & Gomez, 1996). In MetaGer[10], pages are ranked based not only on their original order relationships but also on word counts within the title, the URL and the description of the hits (Beuermann & Schomberg, 1998). In Inquirus[11], the actual pages are downloaded and analyzed. Then a uniform ranking measurement is applied to documents retrieved by different engines to produce a global similarity score. It considers the number of query terms presented in the document, the proximity between query terms, and term frequency (Lawrence & Giles, 1998; Yang & Zhang, 2000).

Dwork et al. (2001) indicated that using individual ranking functions for rank aggregation

[8] MetaCrawler: http://www.metacrawler.com/index.html
[9] ProFusion: http://www.profusion.com
[10] MetaGer: http://meta.rrzn.uni-hannover.de/
[11] INQUIRUS: http://inquirus.nj.nec.com/i2/inq2.pl

in meta-search engines are inadequate for a more fundamental reason: the data being ranked are simply not amendable to a simple ranking function. They also indicated that any rank aggregation should take into account user preferences. Tziztzikas (2001) identifies two major approaches have been used by meta-search engine providers to aggregate the rank order of the fused results. The first approach assumes that the degrees of relevance returned by each system are comparable, and they use them for ordering the results, while the other approach just interleaves the returned orderings using defined preferences based on search engine performance or provider assumption.

Meta-search engines have become a very important tool for searching the web because they increase the search coverage, solve the scalability of searching the web, facilitate the innovation of multiple search engines, and improve the retrieval performance (Meng, Yu & Liu, 2002). Achieving these goals requires effective and efficient merging techniques which is considered the major challenge facing any meta-search engine.

Meta-search engines provider utilized different approaches as indicated for combining and ranking the search results based on heuristic solutions. There isn't any evidence to indicate which one of these solutions provide better results. Therefore, this study tries to investigate the optimum method for ranking the databases in the databases selection process and define the best method for combining and merging the search results based on simple solution.

CHAPTER TWO: LITERATURE REVIEW

2. Introduction:

This chapter introduces prior studies covering search engines and merging multiple evidence[12] especially those conducted in the traditional information retrieval system (Online Systems), TREC studies, and those conducted for meta-search engines. It starts with discussing previous studies related to web based searching tools including search engines and metasearch engines. The second part of the review discusses the data fusion studies and merging multiple evidence approaches including merging algorithms and merging techniques for traditional IR and web search engines. The discussion will focus on the methods and techniques used to evaluate and/or examine the searching tools and the merging algorithms.

2.1 Web Based Search Engines Evaluation

2.1.1. Search Engines Reviews:

Since the availability of the first web based search engine in 1994, an extensive number of studies describing their nature and evaluating their performance[13] have been published. A number of reviews have been conducted to explore their historical overviews, methods and techniques used in indexing and retrieving information on the web, and methods used in evaluating their performance. Examples of these early reviews include Gudivade et al. (1997), Schatz (1997), and Schwartz (1998).

[12] Merging Multiple Evidence is a term known in the data fusion literature refers to combining data from different sources or combining data using different strategies.

[13] The term effectiveness and performance are used exchangeable in Web retrieval referring to recall and precision ratio. They are used exchangeable for this study referring to precision and ranking performance (effectiveness).

Griffith and King (2000) stated that limited number of researches have been done on search engines. These studies tend to involve a relatively small number of searches and an assessment of the relevance of a limited number of documents (for example the first 20 ranked items)

Kobayashi and Takeda (2000) reviewed information retrieval on the web compared with traditional IR techniques. Arasuu, et. el. (2000) focused on the effect of the search engines design on the performance[14] issues related to these systems specially the effectiveness and efficiency. A more comprehensive review of experimental studies is provided by Oppenheim, Morris, and McKnight (2000), who identify the need for a set of benchmarking tests and specify criteria that should be included in the benchmarks to make the study informative and provide valuable results. Jansen and Pooch (2001) present a recent review including an extensive analysis for web searching studies. They also compared traditional IR systems, OPAC, and web searching studies and finally they present a framework for the design and implementation of web user studies, exploring log analysis technique for user queries, search sessions, and failure rate. Rasmussen (2002) conducted a more recent review for indexing and retrieving of web materials. In this review she provides a comprehensive coverage describing the characteristics of the web, evaluations and performance measurements for web search engines (in operational and laboratory environment), indexing, retrieval, ranking techniques used in the web, and user issue in indexing and retrieving on the web including user satisfaction, query length, and query structure. The major point in most of these reviews is that there is an urgent need for building rigorous method for evaluating single or multiple search engines.

[14] Performance in this study refers to the IR performance measurements which include recall and precision.

2.1.2 Single and Multiple Search Engines Studies:

The early studies of search engines were descriptive in nature and general in coverage. These studies have been conducted to explore the major characteristics of one or more search engines such as search and display features. These descriptive (testimonial) researches conducted to compare among search engines on the basis of interfaces design, search features, coverage, duplication or overlapping, and/or uniqueness (e.g. Brinkley and Burke 1995, Courtois et. al 1995, and Courtois 1996). There are two problems in the exploratory studies. First, they are absolute or ephemeral very quickly because of the fast changing nature of the web in general and search engines in particular, Second, they do not provide in-depth analysis of the performance of the web based search engine and they just provide some statistical and descriptive information about database size and their increasing rate, search features and capabilities, and response times

Another type of researches have been performed to evaluate user query structure, length, and reformation, exploring the log analysis technique. Example of these studies include a large number of studies which examined three log files provided by the Excite, AltaVista and Ask Jeeves search engines providers for ASIS[15] conference. These files have been analyzed by different group of researchers (i.e. Spink, Bateman & Jansen, (1998); Jansen et al, (1998); Saracevic & Kantor (1988); Spink, Bateman & Jansen, (1999); Spink & Ozmutlu, (2001); Goodrum & Spink, (2001); Spink et al., (2001); Jansen, Pfaff & Spink; (2000), and Spink, 2002).

[15] ASIS: American Society of Information Science and Technology

These analyses tried to model search engine users, investigate their queries and identifies general patterns for user behavior when searching the web. These types of log analyses provide a snapshot for comparison of public behavior while searching the web. They found that a great majority of web queries posed by public are short, not much modified, and very simple in structure. Very few queries incorporate advanced search features and when advanced queries are posed half of them have mistakes in the structure. Web users tend not to browse beyond the first or second page of results. Users are not much interested in relevance feedback. Overall a small number of terms are used with very high frequency, while there is large number of terms used only once. The language of web queries is very rich and even unique. The distribution of the subject of web queries does not follow the distribution of the subject content of sites. The number of queries posed on the web is huge, but mostly searching pornographic and low art sites. The following part will present some significant studies to demonstrate their results compared with other research in the same area. Some of these studies will be discussed in more details when discussing the study principles (see Chapter 3).

This type of research should also be classified as descriptive or exploratory studies for search engine users and their queries. Thus, they still have the same two major problems of the testimonial studies.

The other type of research explores the experimental design[16] in operational and laboratory environments for evaluating search engines performance. Some of these researches have been conducted to explore methods and techniques to control the search

[16]. The experimental design means at least one independent variable is manipulated such as measuring precision of two different search engines, precision of the individual search engines is the dependent variable and the search engines performance is the independent variable. Other independent variables could include user queries, query length etc,.

engine environment in order to experimentally evaluate their performance; others have been done to evaluate their performance using traditional techniques known in the field of information retrieval. The proliferation of search engines naturally leads to an interesting question of which search engines perform better and a growing number of studies address this question in operational environments.

Most of the time the operational studies examined the performance of three to six search engines and a limited number of these studies exceed that number (3-6 search engines) to evaluate a lager number of search engines for particular purposes.

Chu and Rosenthal (1996) study considers one of the early efforts, explored the experimental design to examine the performance of three search engines (Altavista, Excite, and Lycos). Ding and Marchionini (1996) also conducted an early evaluation of search engines performance. They examined three of the most popular search engines at that time (Infoseek, Lycos, and OpenText). Tomiauolo and Packer (1996) examined the performance of five search engines include Magellan, Point, Lycos, Infoseek, and AltaVista). Su (1997) stated that experimental studies require user oriented evaluation, systematic methodology involving real users that capture information on participant's characteristics as well as precision, relevance ranking by users, and value of search results as a whole. This methodology was employed in a pilot study with faculty and graduate students. Wishard (1998) conducted one of the early studies, evaluated the performance of search engines in particular subject area. She examined the usefulness of 37 search engines in retrieving relevant items in the field of earth science.

Lighton and Sirvastava (1999) conducted an experiment to examine the effectiveness of five search engines (Altavista, Excite, HotBot, Infoseek, and Lycos). The most important

design feature of this study is the blinding procedure, which was used to lessen evaluator bias in judging the quality of the retrieved items. The blinding procedure is conducted through randomizing the search results, so users do not know which results are retrieved from which search engines. Gordon and Pathak (1999) conducted a study to see how effective eight search engines are? They recruited students as mediated searchers, and utilized real user queries and real user relevance judgments. They provided a list of seven criteria an operational experimental study should meet in order to be accurate and informative. These criteria include using "real" queries, employing a large number of searchers, studying most major search engines, having relevance judgments made by the user rather than surrogate judges, using rigorous performance measurements, and conducting experiments rigorously. Chignell et al. (1999) carried out two experiments to study the relative effectiveness of different search engines under different conditions. In the first experiment they examined the performance of Excite, HotBot, and Infoseek, and in the second study they examined the performance of AltaVista, HotBot, and Infoseek. Dennis et al. (2002) experimentally compared search effectiveness when using query based Internet search (via Google), directory based search (via Yahoo), and phrase based query reformulation assisted search (via Hyper Index Browser).

One of the common design features in all of these studies is using the same procedure in evaluating search engine performance including the procedure explored in the Cranfield studies in the 1950s, examined in evaluating the MEDLARS database in the 1960s and exploited for evaluating the online systems in the 1970s and 1980s (Lancaster, 1998). The basic feature in these studies is using mediated or expert searchers to prepare the search strategy and perform the search process to guarantee more interactive and

relevant results. They also examined between the first 10 to the first 20 retrieved items to evaluate the performance of the search engines by using assessors for judging the relevancy of the retrieved documents. The most significant problem in the design of the operational studies is collection control, because each search engine indexes only proportion of the web, which is different –to some extent- from one search engine to another. For example the web pages indexed by HotBot are not exactly the same as those indexed by Lycos or Google or Altavista. Therefore, it is difficult to compare among search engines including different sets of documents without exploring a technique to control these different set of documents. Thus, these studies tried to compare the effectiveness of search engines in terms of the precision ratio for different test collections. In spite of that there are some other factors affecting the performance of the search engines such as the size of the database, the search techniques, and indexing model (e.g. Boolean model, vector space model, and probabilistic model). Other limitations includes relevance judgments which is not available for the assessors, and comprehensive information cannot be obtained for such a large collection (Rusmussen, 2003). Therefore most of these studies artificially provide the relevance judgments by the evaluators or recruited assessors for that purpose.

Most evaluation studies of web search engine performance focused on precision alone, either because of the difficulty of measuring recall, or because precision is claimed to be more important to users. Few studies have been conducted to evaluate search engine performance in terms of recall. For example Gordon and Pathak (1999) used pooling techniques to measure recall in terms of the first 200 retrieved pages. Clarke and Willett (1997) examined 30 queries and three search engines to measure recall. They utilized

pooled recall in which relevant items from each query on all three search engines adjusted for inclusion in the index of the individual search engines and formed the basis for the recall calculation.

The second approach utilized for testing the performance of web based search engines is the laboratory approach which provides an overall control for the different variables that might effect on the IR experiments and systems performance. The laboratory studies evaluate the performance of web retrieval methods by creating a test collection of static web pages and make them available to researchers which allow comparisons to be made between search engines on the basis of the same data. A static web collection allows researchers to isolate specific retrieval algorithms or system components to measure their impact on retrieval performance (Hawking et al. 2001).The laboratory experiment allows for many variables to be controlled: the document collection is static, the queries are provided in a standard form, and the documents that are relevant to a query are known prior to the experimenter. This control makes it possible to compare precision and recall for a set of queries across systems, or for the same system while varying internal parameters (Rasmussen, 2003).The TREC (http://trec.nist.gov/) introduced an annual web track with the goal of building a test collection that laboratory tests this collection for web retrieval. This annual conference hosted by the National Institute of Standards and Technology (NIST) is intended to encourage research in text retrieval based on large test collections, encourage the development of new evaluation techniques, and promote exchange and implementation of research ideas (Rasmussen, 2003). TREC participants are provided with test collections and queries, and results are pooled prior to relevance judgments by TREC assessors. Standardized evaluation measures are used. For the web

track, a 1997 snapshot of the web was obtained and several test collections were produced. In TREC-8, a 2-gigabyte subset (WT2g) was used for the Small Web Task, with performance tested on the TREC ad-hoc topics. This was increased to 10 gigabytes (WT10g) in TREC-9. In both cases a 100 gigabyte subset was used for the Large Web Task employing queries adapted from search engines query logs. Overall goals in the web track were an assessment of how well the best methods in non-web TREC data performed on the web collections and data gathering on the impact of link information. Individual participants had goals related to their own interests, such as Boolean ranked output comparisons, issues related to speed of retrieval, and the role of parallelism (Hawking, et, el, 2002, Rasmussen, 2003).”Using a static web test collection eliminates problems inherent in experimentation on the dynamic web, removing the impact of web crawlers from the assessment of the text retrieval system. It also allows the evaluation of individual retrieval techniques in isolation from specific search engines” (Rasmussen, 2003).

Savoy and Picard (2001) used the 2-gigabytes web TREC track collection to evaluate the effectiveness of established IR techniques. These techniques include a variety of term weighting schemes such as binary, tf*idf, and normalization for document length. They also evaluated the use of stop words, stemming of index terms, and query expansion in the web test collection. Hawking et al. (1999, 2001) compared the TREC retrieval system used in TREC-7 Very Large Collection Track with Web search engines by submitting TREC-7 short queries to five search engines and the TREC collection and compared the results. They found that TREC search engines outperform the web search engines. This study combines the basic feature of the operational and laboratory experiments to

overcome the problem of comparing traditional IR techniques and web search engines. The major problem with the TREC studies is isolating the experiment in a laboratory which might be totally different than what happens in reality. It also evaluates the test collection in a static environment while the web is a dynamic environment. Web Track studies evaluate very small test collections compared to real search engine database sizes.

2.2. Merging Multiple Evidences:

One of the methods has been explored in information retrieval systems to improve search performance is combining search results from multiple sources or strategies. This method is known in the IR literature as results fusion, which refers to merging the results into a unified list of ranked documents. These documents are retrieved in response to submitting a user query to meta-search engines (Yuwono & Lee, 1997; Tsikrik & Lalmas, 2001). Two major approaches have been used for merging the search results. The first approach known as data fusion uses a combination of retrieved results generated through multiple document or query representations or multiple retrieval strategies. The second approach known in the IR literature as collection fusion, which combine the search results from different systems. (Voohrees; Gupta & Laird, 1994, 1995; Savoy, Le Calve &Vrajitoru 1996). "In the context of web, the process is still referred to as data fusion; even though the individual search engines operate on neither the same nor disjoint document collections, but on overlapping sets of web pages"(Tsikrik & Lalmas, 2001). The merging algorithm is the most important component of the data fusion problem. The following part will provide an overview for some of the well known merging algorithms.

2.2.1 Merging Algorithms:

The multiple evidences merging problem in IR is a difficult problem because search engines may use different ranking algorithms and may base their ranking on corpus statistics that vary widely (Tsikrika & Lalmas, 2001). Merging based on un-normalized (raw) documents score (Local Similarity Score) or document ranks works well when search engines and corpora are very similar, but can be very inaccurate when they differ. Usually, documents returned from each component search engine are ranked based on local similarity functions. Some search engines make the local similarity of returned documents available to the user (e.g. Northern Light and FirstGov) while others do not make them available (e.g. Google and AltaVista). "Local similarities returned from different search engines, when made available, may be incomparable due to the heterogeneities among these search engines. Furthermore, the local and the global similarity of the same document may be incomparable" (Meng et el., 2002).

Merging based upon weighted document score or rank has been the state of the art for merging quickly (Voorhees; Gupta & Laird, 1994). The alternate solution is to download the contents of the retrieved documents and then to re-rank them at the search client, which produce a consistent ranking but could be very time consuming and require special algorithms for analyzing and may be indexing these documents because it has to analyze the downloaded documents, then assign a score for each document and rank them according to this new score (Si & Callan, 2002). The following part will focus on the functions used for this second approach of merging multiple search results.

2.2.1.1 Downloading and Analyzing:

One of the major approaches used for analyzing the retrieved documents is know as document fetching. The document fetching method depends on downloading returned documents from their local servers and computes their global similarities using a term weighting function such as the cosine function (Meng et el. 2002). After a document is downloaded, the term frequency for each term in the document can be computed. As a result, the global similarity for each document could be compared and used for ranking the returned documents (Lawrence & Giles, 1998). There are many resource ranking algorithms used for merging search results based on downloading and analyzing the retrieved documents. For example, gGOISS, CORI, and CVV are three of the best known resource ranking algorithms. These three algorithms require downloading the retrieved set of documents from the different engines and then re-analyzing the document according to the defined algorithms. The algorithm also provides a function for ranking the databases according to their similarities to user query to choose the most relevant databases (database selection) then analyze the returned results using global similarity functions based on the underlying algorithms to rank the retrieved results.

gGOISS (**G**eneralized **G**lossary **O**f **S**ervers' **S**erver) is based on the vector space model. It represents a database by the document frequency of each word in the database, and the sum of the term weight in each document in the database. It uses the sum of the document similarities that are higher than a threshold. (Gravano & Garcia- Molina, 1995). The **CVV** (The **C**ue **V**alidity **V**ariance) resource ranking algorithm uses a combination of document frequency (DF) and cue validity variance information. The variability of the fraction of documents in the database that contains a specific word is

characterized by the cue validity variance (Yuwono & Lee, 1997). The CVV algorithm works well when the underlying search engines cooperate with the meta-search engine by providing statistics about their databases. The CVV function performs two major tasks: database selections and results merging (Meng et el 2002). The **CORI Net** (**Co**llection **R**etrieval **I**nference **Net**work) resource ranking represents each database by its terms, their document frequency and summary corpus statistics such as total word count (Callan & Connel, 2001). This algorithm has been very effective in cooperating systems[17] but it has not been applied to search engines other than INQUERY (Callan & Connel, 2001). Different researchers using different datasets have shown the CORI algorithm to be the most stable and effective of the three algorithms (Si & Callan, 2002). Other approaches utilizing document analysis have been developed for meta-search engines. For example, **OptDocRetv** (**Opt**imal **Doc**ument **Ret**rie**v**al) uses the product similarity between an expansion query and a database similarity. This Collection fusion algorithm is based on the global similarity of documents. That means if the databases are ranked optimally, the algorithm will guarantee the retrieval of all N most similar documents, analyze these documents and rank them based on the global similarity score. (Yu et el. 1999).

The major advantage of the downloading solution is that it utilizes consistent methods for analyzing all the retrieved documents which eliminates the multiple systems interoperability problem. But the major disadvantage of this approach is that it requires powerful systems and large disk space. Furthermore, it is very time consuming because the systems first require to search the multiple engines, then download the document in the client server and analyze the document to produce the rank similarity score. All

[17] Cooperating Systems are systems that cooperate in providing data about their database including database description, indexing and ranking algorithms.

these steps should take place on the fly, which is not a simple task.

2.2.1.2 Merging Upon document Rank Score:

The second approach used for merging the search results is based on a simple solution by using the documents' similarity score or rank score. There are several algorithms utilizing this approach:

I. Use the Local Document Rank (Interleave)

This method first arranges the searched databases in descending order of usefulness, depending on some rigorous measurement such as database performance in the database selection step. Next, a round-robin method based on the database order and the local document rank order is used to merge the results. This solution is known as the interleaved merging solution (Meng et el. 2002). Specifically, the first document in the merged list is the top ranked document from the highest ranked database. The next document in the merged list will be the first-highest ranked document in the second highest ranked database and the process continues until the required number of documents are included in the merged list. A randomized version of this method is proposed in Voorhees et el (1995). In this version, instead of using the same order of the databases, they used simple random method for ranking the documents within each step. So the retrieved documents are ranked based on the probabilistic model and each search engine has the same chances to be ranked first. The basic assumption here is, documents retrieved from more important search engines might be better than other documents having the same rank order retrieved from less important search engine.

II. Convert Local Document Rank to Global Similarity Score:

Lee (1997) designed a simple approach for merging the search results known as rank similarity (RankSim). This approach use the document original rank to merge the combined list based on the following function:

$$\text{RankSim (Rank)} = 1 - \frac{\text{Rank} - 1}{\text{Number of Document Retrieved}}$$

The basic assumption here is: A document retrieved within a large set is better than another document that has the same rank order retrieved within a smaller set of documents.

Yuwono and Lee (1997) convert the local document rank score to a global similarity score in D-WISE by employing the following method. For a given query, suppose r_i is the ranking score of database, D_i, r_{min} is the lowest database ranking score (i.e. $r_{min} = \min\{r_i\}$), r is the local rank of document from database D_i, and g is the global converted similarity of the document. The conversion function is:

$$g = 1 - (r - 1) * F_i$$

Where F_i is defined to be:

$$F_i = (r_{min}) / (m * r_i)$$

Where **m** is the number of document desired across all searched databases. As an example, consider two database D_1 and D_2. Suppose $r_1 = 0.2$ and $r_2 = 0.5$. Furthermore, suppose four documents are desired. Then $r_{min} = 0.2$, $F_1 = 0.25$, $F_2 = 0.1$, and $m = 4$. Based on the above conversion function, the top three ranked documents from D_1 will have converted similarities 1, 0.75, and 0.5, respectively and the top three ranked documents from D_2 will have converted similarities 1, 0.9, and 0.8, respectively. As a

result, the merged list will contain three documents from D_2 and one document from D_1. The documents will be ranked in descending order according to the global similarity score in the merged list. Note the database rank (r_i could be defined according to the database performance or database size or any other arbitrary judgments).This function has been used in some meta-search engines after considering the overlapped documents by summing up the similarity scores of the overlapped documents in MetaCrawler or using the max score as in Profusion meta-search engines (Selberg and Etzioni, 1997).

III. Merging Upon the Overlapped Documents:

Fox and Show (1994) designed some of the most simple, popular, and effective data fusion functions to date. These functions include the following parameters: **(**Aslam & Montague 2001).

Name	New Relevance. Score is:
CombMIN	Minimum of individual Rels.
CombMED	Median of Individual Rels.
CombMAX	Maximum of Individual Rels.
CombSUM	Sum of Individual Rels.
CombANZ	CombSUM ÷ num nonzero rels
CombMNZ[18]	CombSUM * num nonzero rels.

These six ranking functions used to calculate the rank score for the overlapped documents which appear in more than one run. The basic assumption here is that a document retrieved in more than one run is better than another document that has the same similarity or rank order retrieved in a single run. The COMBSUM and

[18] CombMNZ: is the combined similarity or rank score for the overlapped document multiplied by the number of runs which have non zero similarity score.

COMBMNZ algorithm are the best known and used algorithms for merging multiple search engines results. These approaches are also based on normalizing search engines scores. The main weakness of this approach is that they rely on a lot of overlap among the results from different search engines which is not guaranteed if the search engines are disjoint. Therefore it is better if it is used as a secondary approach as in MetaCrawler and Profusion.

The major advantages of these three methods of merging are:

- They do not require documents processing, they only require rank score normalization.
- They do not require similarity score which is not available in most popular search engines, they only require documents retrieved with rank order.
- These functions are simple and require less processing time and disk space than the downloading methods.

Two major approaches have been used for merging the multiple search results in Metas-earch engines. The first approach partially or totally downloads the retrieved documents, then analyze the contents of these documents to produce a global similarity score for merging them. Downloading documents and analyzing them on the fly can be very expensive and time consuming especially when the number of documents and their sizes are large. Meng et el. (2002) suggests several solutions for these problems. First, downloading from different local systems can be carried out in parallel. Second, some documents can be analyzed first and displayed first, so further analysis could be done while the user display the initial results. Third, downloading the first portion of each large document to be analyzed, then work on the fly with that portion.

On the other hand, downloading based approaches also have some clear advantages (Lawernce & Giles, 1998). First, when trying to download documents, obsolete URLs can be identified. Second, query terms in downloaded documents could be highlighted when displayed to the user.

The second approach utilizes simple solutions by converting the document rank score into a global similarity score. This approach is the major approach for merging in meta-search engines as indicated above. The major advantage of this approach is that it is simple, quick, cheap, and does not require any special information to be retrieved with the list of documents. The major disadvantage is that most the available function is based on heuristic assumption. For example the rank similarity function is based on the assumption that a document retrieved within a bigger set is better than another document having the same rank order retrieved within smaller set. The global similarity score function assumes that a document retrieved from a higher ranked database is better than another document having the same rank order retrieved from lower ranked database.

It is clear from the previous demonstration that finding an effective combination function in meta-search engine environments is an area that still needs further research.

This study chooses to compare among the three simple functions (The Interleave function, the RankSim function and the global similarity function combined with CombSUM as it is used in the MetaCrawler) of merging the search results to identify the Optimal merging function.

2.3 Data Fusion in IR.

The following part of the literature review will focus on the approaches used in fusing multiple evidences in traditional and web IR.

2.3.1 Data Fusion in Traditional IR.

The problem of merging multiple results from different databases has been addressed in the literature of IR in the late 1970s.

Williams (1977) discussed the problem of automatic Data Base Selector (DBS) being tested at Illinois University. The Selectors operated on user query terms to rank data bases according to their applicability to a query. The test version has: a file of terminology from 20 major data bases; programs for data management, file generation and query processing; a mathematical model for normalizing the variability among different natural language data bases. A DBS would facilitate database and vocabulary comparisons and help overcome vocabulary compatibility problems.

She (1979) also addressed the problem of merging monographic databases of duplicate records in multiple records in multiple files. In a research project entitled "A State Wide Union Catalog Feasibility Study" was funded by Illinois State Library and carried out within the Information Retrieval Research Laboratory (IRRL) of the Coordinated Science Laboratory at Illinois. The project aimed to develop a machine algorithm for locating and eliminating the duplicate records in machine readable bibliographic files from different libraries to be used for union catalogue. A prototyped system IUCS (IRRL Union Catalog System) was developed and tested on sample files from OCLC, Northwestern university, and university of Chicago.

Williams et al. (1979) addressed the problem of searching multiple data bases through building a mapping model and search scheme to facilitate resource sharing. They examined a set of bibliographic databases and identified a generalized set of data elements were formed into a hierarchical structure of compound and simple elements,

and the feasibility of automatically mapping existing databases into that structure was demonstrated. A directory of 161 chemical databases has been assembled using data gathered from this structure.

There are four possible approaches for combining multiple evidences which have been explored in traditional IR experiments. The first approach, based on combining different query representations as examined by Saracevic and Kantor (1988). They asked different experts to construct Boolean queries based on the same description of information problems in operational online IR systems. Belkin et al. (1995) also show that combining different Boolean query formulations could lead to improvements in retrieval effectiveness. They provide a rationale for the data fusion problem as "different representations of the same query, or of the documents in the database or different retrieval techniques for the same query, retrieve different sets of documents (both relevant and irrelevant". Shaw and Fox (1995) indicated that experiments involving all the possible combinations of two types of queries (P-norm Extended Boolean Queries and Natural Language Vector Queries) reveals that combining two of the same type of runs, either both vector queries or P-norm queries shows little improvement over the individual runs, and performs worse than the better of the two runs in many instances. However, combining one of the two vector queries with one of the P-norm queries always shows an improvement. This indicates that the primary source of improvement seen in the combination runs submitted for TREC-3 derives from the combination of retrieval paradigms and not simply from the combination of multiple queries. This may be due to the similarity inherent in the five queries; combining two queries composed of two widely different sets of query terms may well result in significant improvements. But

given a single set of query terms, it is still possible to achieve significant improvements by combining different retrieval paradigms.

The second approach is based on combining different document representations. Katzer et al. (1982) consider the effect of different document representations, e.g. title, abstract, on retrieval effectiveness. They discovered that various document representations gave similar retrieval effectiveness, but retrieved quite different set of documents. Their results suggest that the combined run may retrieve more relevant documents than any individual run, therefore providing high recall. Turtle and Croft (1991) developed an inference network based retrieval model to combine different document representations and different versions of a query in a consistent probabilistic framework. The model treats different representations as evidence that is combined to estimate the probability of a document that is satisfying a user's information need. They implemented their model using the INQUERY retrieval system and demonstrated that multiple evidence increase retrieval effectiveness in some circumstances.

The third approach is based on combining document retrieved from single system using single retrieval technique for single query and document representation implementing different term weighting schemes. This approach has been introduced by Lee (1995). His study shows that significance improvements can be obtained by combining the retrieved results from different properties of weighting schemes. He applied the combining method to pair-wise combinations for six runs using different two weighting schemes for each run. He indicated that the combination achieve improvement in the precision at the 11 point recall average ranged from 2.9 % to 14.5 %.

The fourth approach is based on combining document sets retrieved from different IR systems with different collections including a considerable amount of overlapping among these systems. There are numerous studies have been conducted using this approach which shaped the basic feature of most meta-search engines available today. (discussed in more details in the following section). Lee (1997) examined multiple combination function and showed that function called CombMNZ provides better retrieval effectiveness. He investigated the rank rather than the similarity values for merging multiple evidence. He evaluated the rank function (explored in this study) using six selected retrieval results from TREC 3 track ad hoc.

These four approaches used for merging multiple evidences in traditional IR systems confirmed that combining multiple results from different sources or from the same source could improve the performance depending on the method used for merging the search results. Most of these studies confirmed that the multiple runs for different query structures or different document representations retrieve different set of documents both relevant and irrelevant which increase the recall ratio but might decrease the precision ratio.

2.3.2 Data fusion in the Meta-Search Engines

Yang and Zhang (2000) identify and classify the potential cases of fusion in Meta-search engines. They classified these cases into four major types: (1) an equivalent case; (2) an inclusion case; (3) a disjoint case; (4) an overlap case. The equivalent case is applicable when the search engines retrieve the same set of documents. The inclusion case appear when one search engine retrieve set of documents that include the set of document retrieved by the second search engines. The disjoint case means that search engines retrieve different set of documents. The overlap case which is more applicable

for meta-search engines appear when search engines retrieve overlapped set of documents. They believe that existing meta-search engine merging algorithms do not satisfy the necessary constraints and the performance of these algorithms are in doubt.

Most if not all the reviewed techniques used for merging multiple search results is based on using collection fusion approach and different ranking techniques. Yuwono and Lee (1996) evaluated four ranking algorithms based on keywords matching and hyperlinks: Boolean Spreading Activation; Most-cited; the *tf*idf* vector space model; and vector spreading activation, which combine *tf*idf* with spreading activation. The major motivation of their study is to define which two algorithms could be combined together to provide a way for merging the search results for metasearch. They found that term-based approached worked better than link based ones.

Smeaton and Crimmin (1996) examined the data fusion approach where the output from six search engines (AltaVista, Excite, InfoSeek, Lycos, OpenText and WebCrawler) combined into a unified ranked list to build a meta-search engine using a client server architecture. The rank of the documents/pages based on their Retrieval Status Values (RSV), which compute a score for each document based on the some variant of weighting of search terms. Dong (2000) investigates the effect of applying multiple evidences combination technique on 30 questions submitted to four search engines (Excite, HotBot, Lycos, and Infoseek). He examine two ways, three ways, and four ways combination and its influence on the precision ratio to investigate the effect of combining search engine results on the overall performance and 11 point recall precision for different query length. He examined the Rank Similarity function developed by Lee (1997) for merging the search results but without assessing whether it is an effective way

for merging or not. He found that the combined results did not change significantly when combination was done at higher level. He also found that the average precision over all relevant documents and 11 point recall levels obtained at 4 ways combination is significantly better than the other three ways of combinations.

Dowrk et al. (2001) consider the problem of combining and ranking results from various sources or search engines. The main application of their study was building a meta-search engines. Most data fusion studies indicate that the relevancy of the retrieved data could not be defined without analyzing the data, but this study assume that the retrieved data could be judged without analyzing it by using user judgment to indicate the data relevancy. They developed a ranking aggregation algorithm depending on methods known in the voting systems. A primary goal of their work was to design a rank aggregation technique that can effectively combat "spam" the search results. Basically the combination algorithms depends on voting system by trying to insert a document retrieved at the top of the list at the end (bottom) of the aggregated list; but they have to bubble it up toward the top of the list as long as a majority of the voter's insists that it should be there. They identify the method of Kemeny and Markov chains, originally proposed in the context of social choice theory as the principle model of their rank aggregation function. They indicated that while there is no guarantee on the quality of the output, their method is extremely efficient, and usually match or outperform other methods. The problem of their method is that it is very sophisticated approach and requires voter to order the retrieved set of document, whether those voters are human or systems it still required a lot of processing time.

Tsikrika and Lalmas (2001) investigate merging techniques, which aim at improving the effectiveness of meta-search engines by processing more of the information provided to them by the participants search engines. They explored four major search engines (Google, Infoseek, Northern Light, and WebCrawler), for the first 30 retrieved documents, and 10 general queries. The proposed merging techniques utilize not only the rank positions[19] of the retrieved documents, but also their title and summary accompanying them to describe their content[20]. Furthermore, the data fusion process is viewed as being similar to the combination of belief in uncertain reasoning and is modeled using Dempster-Shafer's theory of evidence[21]. The list of retrieved documents corresponds to bodies of evidence, which are merged (aggregated) using Dempster's combination rule. Finally, it is indicated whether the effectiveness of the proposed merging strategies, which are based entirely on the information provided by the underlying search engines, is comparable to the approach that merges ranked lists by downloading and analyzing the retrieved web documents or not comparable, it require less processing time and disk space.

Aslam and Montague (2000) developed a probabilistic model for combining ranked lists of document obtained by a number of retrieval systems according to a given query. Their model based on the average performance of combined systems. They calculated the relevance of a document for ranking purpose using the sum of the log of the ratio of the probabilities over all systems using the following formula:

---

[19] The simplest method of merging lists in metasearch engines by taking into account the rank positions of the documents. In this method, the duplicate documents have their ranks summed up and the rest of the document interleleaved.

[20] indexing the content of the title and summary using vector space model, using ***similarity function*** for term weighting.

[21] It is an extension of the probability theory and it allows the explicit representation of uncertainly and the combination of evidence. The combination rule computes the agreement between two bodies of evidence.

$$\sum_{I} \log \frac{Pr[ri|rel|}{Pr[ri|irr|}$$

Note that Pr[ri|rel| is the probability that a relevant document would be ranked at level ri by system *i*. Similarly Pr[ri|irr| is the probability that an irrelevant document would be ranked at level ri. They tested this model using the TREC data set and compare it with the CombMNZ using the precision at 11 point recall. They claim that their model is outperform the CombMNZ model and achieve significance performance improvements. The major problem with this model is that it requires resource description, judged documents and the combination process based on the assumption that the similarity score retrieved with each document which is not true in most of web search engines cases because most search engines only retrieve ranked list of documents without any other information.

Aslam and Montague (2001) explored new technique for normalizing relevance score for un-retrieved documents. They showed that the techniques used so far for normalizing and estimating the relevance scores of un-retrieved documents can have a significant effect on the overall performance of meta-search engines. They used a simple score estimator for the un-retrieved documents by assigning a relevance score two standard deviations below the mean. They explored both the CombSum and CombMNZ to compare normalization algorithms. They used TREC benchmark including the data set and 50 web track queries.

Meng et el. (2001) developed an approach essentially for database selection and collection fusion for meta-search engines. Their framework first tries to rank local databases optimally using the OptDocRetv algorithm developed by Yu et el (1999). The measure used to rank a database is the similarity of the most similar document in the

database. Then they developed an algorithm to determine what database should be searched and what documents from each database should be returned to meta-search engines. Finally the global similarity of returned documents is used to merge all returned documents. They tested their model using 1000 queries collected at Stanford University. These queries have no more than six terms and have a mean of 2.4 term per query. They used three TREC collection sets and explore 221 databases. The results of their study show that their approach is working well with short queries (1 and 2 terms) which represent the major type of queries in the web. But their model returned poor results with longer queries. The major problem with their approach is that it requires specialized database in order to provide effective database selections and in the same time it requires database description which require very long processing time and database provider cooperation. Si and Callan (2002) address the problem of merging search engines results obtained from different databases and search engines in a distributed information retrieval environment based on single search and multiple search engines. They combined the retrieved results on a single database using query based sampling to provide resources description. Then they used this database as source for training data for adaptive results merging algorithm which based on the CORI algorithm. The major problem with this approach is that it requires downloading a large number of documents for resource description and normalizing the document scores using regression formula which require a lot of time and powerful engine.

Although these previous studies have explored the fusion technique, they include some major deviances from the basic principles of this study, which are: non of them tries to explore real user queries through the whole process of building the search strategy, no

real users involved in the process of evaluating the documents relevancy, and non of them tries to examine the rank aggregation of the search engine results based on user preferences and relevance judgments and how it might effect on the ranked results of meta-search engines. There are also few number of studies tried to investigate the rank aggregation process in the web environment. Only one study used the average performance of search engines, only two studies used the rank similarity function, and three studies explored the CombMNZ and CombSUM function, but non of the pervious studies compared these three function of merging. These three functions have been chosen for the comparisons because they do not require document processing which is more realistic in the web environment.

CHAPTER THREE: METHODOLOGY

3 Introduction:

This chapter of the study presents the research problem statement and the proposed methodology for exploring the stated problem. The methodology will discuss the study hypotheses, the study principles including the test sets (the queries set, and the search engines set), the query construction and the statistical tests.

3.1 Problem Statement:

So far, the studies that have been done for merging multiple search results provide neither a systematic approach for merging the search results based on user preferences for system training nor have them compared different merging function to detect the best algorithm. It has been known that different sources retrieve different sets of documents for the same query. A number of studies suggest that significant improvements in retrieval performance can be achieved by combining multiple sources such as retrieved results from different search engines. The primary motivation of this research is developing a rigorous procedure for meta-search engines. This procedure could mainly help meta-search developers in three major steps of the building process. These steps are ranking the selected databases based on their optimal retrieval performance rather than their popularity or size, choosing the best combination from any set of search engines, and evaluating different heuristic merging functions to select the optimal one. The preliminary system aggregation technique could be used for merging the search results of N number of runs then the system log file could be used for training the meta-search engines to detect the average performance after N number of search. The three stage approach could be utilized by meta-search developers in choosing the best combinations

from any set of search engines based on rigorous measurements for database selection and database ranking, and the best rank aggregation method for the combined list without having to analyze the retrieved documents. So this approach could be used as a baseline for developing meta-search engines and it also could be used for evaluating existing meta-search engines.

This approach depends on exploring preliminary relevance judgments collected from users as a starting point for ranking the selected engines, to examine the combination, and to select the appropriate rank aggregation method according to system performance. Then the system could utilize this preliminary information for merging the results in a unified list. The system could utilize this information for the second run and learn from user interaction with the system to enhance the rank aggregation. This means that the system does not have to reanalyze the retrieved documents which is considered the most difficult and time consuming task for meta-search engines. This approach could also decrease the required processing time and system cost as long as no data analyzing is required. The approach could resolve three major problems:

1) How meta-search developers can define the optimal rank order for the selected search engines?
2) How metasearch developers can choose the best combination from any set of search engines?
3) What is the optimal heuristic merging function that could be used for aggregating the rank order of the retrieved documents form incomparable engines?

In short, this study have included three major parts: the first part provides a framework for database selection, specially database ranking based on the overall

performance of the search engines, the second part examines the different combination performance to define the optimal combination based on rigorous measurement in IR, the third part compares among three merging function to define the optimal one. The merging methods are the interleave function which utilize the search engines performance examined in the first part, rank similarity function developed by lee (1997) and the global similarity function developed by Yuwono and Lee (1996) combined with CombSUM function developed by Fox and Show (1995) in the second TREC conference. These three merging functions are based on different assumptions that could improve the search performance.

3.2 The Research Hypotheses:

Hypothesis 1:

Larger databases tend to rank higher than smaller databases because their overall performance tends to be better than smaller databases in terms of precision ratio.

To test this hypothesis three search engines with different size have been selected.

$$\mu_{Gp} \neq \mu_{Ap} \neq \mu_{Fp}$$

$$\mu_{Gp}[22] = \mu_{Ap}[23] = \mu_{Fp}[24]$$

Hypothesis 2:

Given a set of test queries run against a set of search engines, the fusion or combination of more than one search engines tends to achieve higher precision ratio in terms of their general performance and their different query length. To test this hypothesis the general combination performance for each combination method will be compared against the

[22] The mean of Google precision ratio overall all whole number of queries.
[23] The mean of Altavista precision ratio overall all whole number of queries.
[24] The mean of FAST (Alltheweb) precision ratio overall all whole number of queries

other two combination functions to detect the best method of combining and the best combination.

There are three functions used for combining and merging the results which are: interleave, rank similarity and global similarity. Each combination has been formed according to these three functions. For example, there are three possible pair-wise combinations include: Google – AltaVista; Google – Fast and AltaVista – Fast. These three possibilities have been compared according to each combination function. Then the best three combination and combination methods have been compared to detect the best combining method and the best combination. For example, there is one possibility for the three way combinations and the methods of combining the search results.

$$\mu_{\text{p3 way combination Interleave}} = \mu_{\text{p3 way combination Rank\_sim}} = \mu_{\text{p3 way combination Global\_sim}}$$

$$\mu_{\text{p3 way combination Interleave}} \# \mu_{\text{p3 way combination Rank\_sim}} \# \mu_{\text{p3 way combination Global\_sim}}$$

Testing this hypothesis provides a baseline for the best combination.

$$\mu_{\text{p2 way combination general}} \neq \mu_{\text{p2 way combination length 2}} \neq \mu_{\text{p2 way combination length 3}}$$

$$\mu_{\text{p2 way combination general}} = \mu_{\text{p2 way combination length 2}} = \mu_{\text{p2 way combination length 3}}$$

Hypothesis 3:

If the Global Similarity Function (GSF) combined with the CombSUM (sum of individual relevance) for the overlapped documents, the rank similarity function which uses the number of the retrieved documents from each database as an indication to the importance of the database, and the interleave function which will be based on the mean of the relevance score of the system performance are compared as ways of merging

search results, the mean of the combined list performance tend to provide no statistical significant differences among the three functions of merging.

$$H_0 = \mu_{Interleave}[25] = \mu_{Rank\_sim}[26] = \mu_{Global\_sim}[27]$$

$$H_1 = \mu_{Interleave} \neq \mu_{Rank\_sim} \neq \mu_{Global\_sim}$$

3.3 The Study Principles:

The target approach of this study tries to include six of the seven criteria provided by Gordon and Pathak which discussed in chapter two(1999) for considering search engine experiments an accurate and informative. These six criteria include "real user queries, topic queries; different level of complexity; studying major search engines; having relevance judgments made by user rather than surrogate judge; and conducting experiment rigorously. The study does not utilize large numbers of searchers because it is practically difficult and might effect on the final results.

3.3.1 Principles of Selecting the Queries:

Queries are the primary means of translating user information needs into a form that IR systems can understand and terms are the basic building block of queries (Jansen et al, 2000). Although web search engines utilize the basic principles of IR, web users and their queries seem to differ significantly from traditional IR system such as DIALOG or assessors which are used to provide relevance judgments in TREC (Jansen et. el., 1998).

---

[25] $\mu_{Interleave}$: The mean of the relevance score of the combined list ordered according to the individual search engine performance, using the database order. So document one from database will rank first, document 1 from database 2 will rank second and so on (see 24)

[26] μ_{Rank_sim}: The similarity rank score calculated by using Lee formula (see p. 23).

[27] μ_{Global_sim}: Global Similarity Function. Will be used for calculating the global score and the CombSUM will be used as a way for indicating the importance of the overlapped documents (see p. 24)

A series of studies have been recently conducted to analyze and describe search engine queries. The analysis focuses on the query length, structure, reformation, and other components in order to modeling web queries and tailoring web retrieving system. Jansen et al (2000) analyzed an Excite transaction file containing 51,473 queries posed by 18,113 users to identify queries based length (i.e. number of terms), structure (use of Boolean operators and other modifiers), and failure analysis (deviation from published rules of query construction). They identified that web queries are short, because 62 % of all analyzed queries contained one or two terms, fewer than 4 % had more than 6 terms. This is less than the mean number of term search used in searching regular IR system, which ranged from 7 to 15 terms.

Spink et al (2000) analyzed the same transaction log file to examine the use of query reformulation, and particularly the use of relevance feedback by users of Excite search engine. Results showed limited use of query reformulation and relevance feedback, only one in five users reformulated queries and most relevance feedback were successful. Jansen & Pooch (2002) demonstrate that the query length (number of terms per query) in the web searching tools is two terms per query, while in traditional IR systems ranged from 6-9 terms per query. They also demonstrate that 89 % of web users do not use the advanced features in the search process, while more than 85 % of the traditional IR users run their queries exploiting the advanced features in the search process such as Boolean and proximity operators.

Since this study is designed to develop a framework for combining multiple search engines results some rules have been designed to control the query articulation process, include:

I. No single term query will be used in the study because as mentioned above the mean of the web query terms is two terms.

II. Only subject queries are included which exclude known-item queries that seek information about particular person, a specific institution, and/or given product, which could be satisfied once the item is found in the top of the list. However a subject query seeks for comprehensive coverage form the system will be used.

III. The queries should take the form of noun phrase. This form of queries indicated as the norm pattern in searching the web by Jansen et al (2000).

IV. All the possible variations of the query terms should be indicated in order to develop a reliable search strategy and run the most appropriate terms.

3.3.2 The Size of Test Suite:

Most of the experiments have been conducted to evaluate the effectiveness of search engines performance have had a small test suite for general information needs. The queries reflected real information needs in some cases and artificial queries in others. Chu and Rosenthal (1996) examined 10 general queries represent real reference questions. Ding and Marchionini (1996) tested five unreal general queries. Clarke and Willett (1997) tested 30 queries in topics dealing with library and information science using unstructured search expressions. They stated that thirty queries allows them to treat data as normally distributed. Lighton and Srivastava (1999) explored 15 queries actually asked at a university library reference desk. The queries include general topics in undergraduate academic settings. Dong (2000) also examined 30 general queries generated from different sources including previous studies and reference questions. He

stated that 30 queries is a suitable number for testing web search engine performance. Simon et al (2002) generated eighteen general queries and blocked them into three sets of six queries in each set. Undergraduate student from department of psychology have been used to run the experiment and evaluate the results.

A web statistical java program (Lenth, 2002) has been used to calculate the appropriate sample size and the level of the reliability of the results. The program indicates that 40 queries could be used in level of significance .95, with a power of .05 and a standard error of 0.0684. So, this study examines 40 general queries to detect an approach for merging multiple search results.

In order to examine the effect of complexity degree of query length two different degree of complexity will be examined and compared. The degree of complexity is defined as the number of terms in case of using noun exact phrase search strategy and number of Boolean Operators (AND, OR, +) used to link the query terms involved in the search process in case of using Boolean strategy. Two different degrees have been indicated:

Degree 1 = 1 Operator (two terms) Degree 2 = 2 Operators (three terms)

3.3.3 Building the Test Suite:

The query set is one of the major requirements of IR experiments. Three methods have been used for building the query set. These methods include:

I. Real User Queries:

Using real queries requires users declaring their information needs. The pilot Cranfield and MEDLARS IR experiments utilized real user queries (Lancaster, 1998). In the web search engines evaluation studies there was a tendency for using real user queries submitted to library reference desks (i.e. Chu & Rosenthal, 1996 and Lighton &

Sirvastava, 1999). The major advantage of using real user queries is the variability in the query structure and length which is more appropriate for the real IR system such as web search engines.

II. Artificial Queries:

The second approach for building the query set is artificially creating the queries. Few search engines studies utilized this approach. For example Clarke and Willett (1997) artificially created 30 queries in topics dealing with library and information science. The major advantage of this approach is the overall control on the query structure and length which more appropriate for the artificial setting.

III. TREC Queries

Like the previous method, the TREC queries are suitable for the artificial setting. TREC queries provide advantage only if they used with TREC collection and their relevance judgments.

This study is designed to utilize real user queries in real IR settings and real relevance judgments. A plan has been made to collect two groups of queries from real users with real information needs each group include 20 queries. These two groups represent two and three terms as different degree of complexity.

3.3.4 Performing the Search Strategy:

In order to build a complete real search procedure, the participants have been asked to fill a request form. The request form has been used to conduct a reference interview with the participants. The form includes the problem statement, the keywords, keywords variations, and the search strategy. The search strategies have been performed

according to the participants' syntax without adding any additional terms to the original problem statement.

3.3.5 Running the Queries:

The best approach to search the same query in multiple IR systems is to set up a single interface from which the query can be redirected to all the target search engines at the same time. This approach has been adopted by many meta-search engines (Lawrence & Giles, 1999; Dong, 2000). This approach is technically and practically difficult since it requires a lot of time for building metasearch interface and query translation routines. This study choose to submit the same query to the individual search engines approximately in the same time using the advanced search options as preferable option because it provides more sophisticated and accurate search syntax.

3.3.6 The search Engines:

Since the purpose of this study is to build a framework for merging multiple search engine results for metasearch engines, some criteria have been followed in selecting the search engines. These criteria include:

A. It should be one of the most comprehensive and popular search engines.

B. It should be general in its coverage and free in its service.

C. The selected search engines should be different in terms of their databases size and include fairly amount of overlapped documents.

According to the Search Engines Showdown[28] web site which constantly assess the size of web search engines databases, Google, Alltheweb, and WiseNut respectively are the most comprehensive search engines in their coverage. There is a competition between WiseNut and AltaVista but it is indicated in many other web site interested in search

[28] http://www.searchengineshowdown.com/

engines analysis that AltaVista is more comprehensive and popular than WiseNut (see Search Engines Watch, Search Engines.com, Infopeople[29], and Search Engines Index, 2003). AltaVista is also a major engine in many meta-search engines aggregation process such as Ixquick and ProFusion.

This study explores Google, Alltheweb (FAST) and AltaVista as the most popular and comprehensive search engines. A decision have been made to explore three search engines because if no overlapping detected among the three search engines that means the participants have to judge 30 web pages and if the number of search engines increased this will make the relevance judgments process more sophisticated and require more time and effort from the assessors and the searcher as well.

3.4 Post Processing Results:

Once the query is processed by the three search engines the first 20 items and the total number of the retrieved items from each search engines have been saved in a text file. A spreadsheet has been used for collecting the first top 10 items from the three search engines in a text format. This spreadsheet includes 30 items and record for each item the page original rank order, a random number selected from random table, the page title, and the page URL. The random numbers eliminate the participants' biasness in determining the relevance judgments since the participants neither know where these items retrieved from nor what is the rank order of these items.

Two PERL scripts have been created to process the spreadsheet items. These scripts perform the following tasks:

I. Detecting and eliminating the internal and the external overlapping.

[29] http://www.infopeople.org/

II. Randomizing the rank order of the retrieved items.

III. Producing an index HTML file includes the title of the item linked to the original page.

3.5 Relevance Judgments:

Korfhage (1997) defined two methods for judging relevance, binary and n-array. The binary relevance is the simplest to implement and to use but presents coarse judgment for the user. Each document is either accepted (assigned a score of 1) or rejected (assigned a score of 0), while the n-array allows the user to consider levels or degree of relevance. Griffith and King (2000) defines relevance as the relationship between the expression of user's request and the system's response, which could be an abstract or other surrogate information. This definition represents the system's assessment of relevance.

In Web TREC track (TREC-7 and 8) a three and four relevance scale have been used for web evaluation (Sormunen, 2002). This study will use the N- array relevance scale. A scale of 5 (0-4) has been chosen to measure document relevancy. The relevancy scale defined as: a scale

4: Highly Relevant 3: Relevant 2: Marginally (Partially Relevant)

1:Irrelevant 0: Highly Irrelevant

This scale has been used and proofed to be more reliable for the web environment as indicated by Sormunen (2002). Jarvelin and Kekalainen (2000), Voorhees (2001) who suggested that IR systems effective in finding highly relevant documents might suffer of binary and liberal relevance criteria.

This scale is more appropriate for the nature of web documents because some web pages

might include hyperlinks lead to relevant documents, some might include some relevant information and irrelevant information.

Assessors have been guided to base their judges according to the query topic not the query terms. The assessors have been asked to use the following instructions for the relevance judgments (Modified from Sormunen, 2002):

4. **Highly Irrelevant**: The document does not contain any information about the topic and is not related to the topic at all
3. **Irrelevant:** The document does not contain any information about the topic but it deals with query terms such as mentioning the query terms but dealing with another topic..
2. **Partially Relevant:** The document only points to the topic. It does not contain more or other information than the topic description. Typical extent: one sentence or fact such as topic definition or description.
1. **Relevant:** The document contains more information than the topic description but the representation is not exhaustive. In case of multi-faceted topic, only some of the sub-themes or viewpoints are covered. Typical extent: one text paragraph, 2-3 sentences or facts.
0. **Highly Relevant:** The document discusses the themes of the topic exhaustively. In case of a multi-faceted topic, all or most sub-themes or viewpoints are covered. Typical extent: several text paragraphs, at least 4 sentences or facts.

Others: If the page represents an advertisement for a book or if it represent a list of publications related to the topic it takes a score of one but if the page is not related to the topic it takes a score of zero. If the page link is dead the page assigned

score of zero.

The Index HTML files have been sent to the participants to judge the relevance of the retrieved items using the previous five categories. Assessors uses a web based relevance judgment form.

3.6 Data Analysis:

In order to define the appropriate combination of multiple search results and the appropriate rank order for the unified list, the following statistical tests will be conducted:

3.6.1 Search Engines Rank Order:

In order to detect the optimal rank order for the selected search engines, the total number of items retrieved per query will be compared for the three search engines and the precision ratio of each search engines to examine the effect of the database size on the number of document retrieved per query and on the precision of the first 10 documents retrieved by each search engines. This test has been run for general performance and for the different query length. Search engines are ranked according to their optimal performance (see section 3.7.3.1).

3.6.2 Overlapping Test:

There are two distinct types of duplication among web search results. Internal duplication which appears within the result list returned from a search engine. External duplication which appears when the same document retrieved by more than one search engines. Internal duplicate means lack in the search engines capability to detect duplicate and remove them, while external duplicate is a normal thing. The study decided to remove internal and external duplicate from the retrieved combined list. External

duplicate detected as a ratio for all the possible combinations**.** The effect of query length in the number of overlapped document has also detected using two way ANOVA test for the two different pairs of complexity.

3.6.3 The Precision Ratio (Hypothesis 1: H1):

Cleverdon (1991) suggests that precision is quite important but for most users high recall is not very important. And this could be ultimately true in the web search engines environment because search engines retrieve thousands of documents for most queries.

3.6.3.1 Individual Search Engines Precision Ratio:

This study measures the precision in the level of the first 10 items. The average precision ratio(APR) for each search engines equal the sum of precision (P) for each query (Q) divided by the query number (N). For example the Average precision for Google has coined as (APRg)

$$APRg = \frac{\sum_{Q=1}^{N} Pi}{N}$$

Where Q = 1, 2,3,….,40.

3.6.3.2 Precision of the Combined results (H2):

The average precision ratio for all the possible combinations has also measured for the two way and three way combinations. For example: For two way combination include Google(g) and AltaVista (a)

$$APRga = \frac{\sum_{Q=1}^{N} Pi\ (ga)}{N}$$

For the three way combinations:

$$APR_{gaf}{}^{30} = \frac{\sum_{Q=1}^{N} Pi\ (gaf)}{N}$$

The precision ratio has also calculated for the two levels of complexity for the different combinations and compared to detect the performance within these degree of complexity (H2)**.** The two combinations has measured for the first 10 according to three merging functions and compared to detect the best combination. The optimal two ways combination compared to the optimal three ways combination to detect the consistency of the merging function in detecting the best combination.

3.6.3.3 Precision at 11 Point Recall Values (P11):

The precision at the 11 cut off value has been computed using the recall level at the standard 11 points. These standard levels allow measuring the performance of the possible combinations in major the areas of the retrieved results distribution. For example if the system retrieved only 4 relevant documents at rank order 2, 3, 5, and 7. Then at recall point 0.30 precision is 2/3 = .667 since among the top three documents only two documents are relevant. At recall point 0.60 precision is 3/6 = 0.50 since among the first 6 documents three documents are relevant. At recall point 0.90 precision is 4/9 = 0.444. The 11 point average precision has been calculated using the following formula.

[30] g: Google a: Altavista f: FAST (Alltheweb)

$$P11 = \frac{\sum \text{precision}_{\text{relevant, Q}}}{N}$$

Where N = 40 queries

The precision for the different combinations has been calculated using the average precision at the 11 point recall so for three way combinations, the average of the standards cut off has been used. For example if the precision of 0.30 cut off recall are 0.5, 0.37 and 0.40 for Google, AltaVista, and FAST respectively, then the average precision equal (0.5 + 0.37 + 0.40) / 3 = .423. All the possible combinations are compared at the 11P cutoff values. The precision at the standards cutoff for the different degree of complexity in the query constructions are calculated and compared across all the possible combinations (H2).

According to the search performance, the best combination and rank order of the search engines are used in the Interleave function for merging the search results.

3.7 Results Merging (H3):

This study is trying to develop a framework for ranking the combined list based on user preferences and interaction by collecting primary relevance judgments. So the system will be adaptive with the user preferences.

This study compares among three simple methods of merging the combined list. Analysis of variance has been used as the standard statistical test to compare between the three functions performance.

3.7.1 Merging Search Engines Results:

Lee (1997) indicated two major possibilities for combining multiple search results (similarity and rank). He noted that in data fusion literature similarity is more often utilized to combine evidence than rank values. He indicated that people think that using

similarity gives more effective results and also the rank of the documents have not been available at fusion time. This study utilized the rank values for the combination function for three major reasons. First, the rank of the individual documents is available at the time of the fusion process. Second, non of the explored search engines return similarity values for the retrieved documents. Third, according to Belkin et al. (1995) the internal representations used by different system to produce their document rankings may be incommensurable; in such cases the combination of evidence form different system must be based on the rank order.

The rank similarity function is based on a combination function proposed by lee (1997). This function assumes that a document (D1) ranked at a position R from larger retrieved set is better than a document (D2) ranked at the same position from smaller retrieved set. Thus when merging those two lists D1 proceed D2 in the merged list.

Run 1:

$$\text{Rank\_sim(rank)} = 1 - \frac{\text{Rank} - 1}{\text{Number of total retrieved documents}}$$

For example suppose an individual run retrieves top ranked 1000 document.

Given a document ranked at 10, the similarity value of the document is equal to 0.991.

Another system retrieve 2000 document. The document ranked at 10 similarity value is equal 0.996. So the document retrieved within 2000 document at the 10^{th} rank order should proceed documents retrieved within 1000 documents at the same rank order.

Run 2:

This run is based on the Global Similarity Function(GSF) developed by (Yuwono, &Lee, (1996) combined with Fox and Show (1995) CombSUM merging function. The GSF has

been used for calculating the a similarity score for each document according to its rank order and the CombSUM has been used for summing the similarity score of the overlapped documents and multiply it by the number of runs it appears within. The global similarity function work as follows:

$$\mathbf{g = 1 - (r - 1) * F_i}$$

Where r is the document rank and F_i is defined to be:

$$\mathbf{F_i = (r_{min}) / (m * r_i)}$$

Where (r_{min}) is the minimum database rank, r_i is the database rank, and m is the number of document desired across all searched databases.

While the CombSUM = The sum of the global similarity score indicated through the individual runs. For example if there are three documents retrieved; the first document retrieved from three search engines and the second retrieved from two search engines, and the third retrieved from only one search engine. The rank score of each document could be calculated as follows:

doc1 has similarity scores (0.75, 0.56, and, 0.45); doc2 has similarity scores (0.66, and 0.22); and doc3 has similarity sore of 067. The final similarity score of these three documents could be calculated as follow: For doc1, similarity score sum * number of run = (0.75 + 0.56 + 0.45) * 3 = 5.28. For doc2, the similarity score = (0.66 + 0.22) * 2 = 1.76; and for doc3, the similarity score = 0.67 * 1 = 0.67. Finally, the document will be ranked according to the final similarity scores.

Run Three:

In this run, the average performance of the three search engines has been used for ordering the combined list using the interleave function. This average performance is

defined in the first part of this study using rigorous measurements in IR. The search engines have been ordered according their performance then the document are sorted according to system order by dealing with first document then second document, etc. **In sum,** For the rank similarity formula the combined list has been sorted according to the rank similarity score. For the Global similarity function the combined list has been sorted according to the rank position multiplied by the number of run. For the average performance methods (Interleave) the results has been sorted according to the search engines performance. For example if X search engine achieve higher average performance than Y search engine then document 1 from X should proceed document 1 from Y.

3.8 Programming Tools:

Two C++ programs have been developed to convert the rank order of the retrieved document to similarity scores according to the rank similarity function and the global similarity function. These programs facilitate merging the search results according to the defined function.

3.9 Rank Similarity Normalization:

The relevance judgment scores have been used for normalizing the score of the three function. For example, the similarity rank has been eliminated and replace it with the relevance score, the same procedure takes place for the CombSUM score and the average performance. This provides a baseline for the comparison using Two way ANOVA test to detect the best function.

CHAPTER FOUR: RESULTS AND ANALYSIS

4 Introduction:

This chapter presents the results of the combination and merging experiments in the following order:

1. General Description
2. Search Engines Performance
    - Individual Engines performance
    - Document Overlapping
3. Performance of Combination methods
    - Two Way Combination
    - Three way combination
    - Overlapped Document Relevancy
4. Performance of Merging Methods
    - Merging Two Engines Results
    - Merging Three Engines Results

4.1 General Description

4.1.1 Total Number of documents retrieved by each Search Engine (SE).

During July 7-30. 2003. The 40 queries listed in appendix 1 were submitted to the three search engines selected for this study. The 40 queries divided into two sets represent different query length (QL): 20 queries with length two terms and 20 queries with length three terms. shows the mean number of items found by each search engines for the whole set of queries and per query length.

Table1: Mean number of documents retrieved for each SE per query length

SE	Mean	Std. Deviation	N
GOOGLE2[31]	14882.55	23029.6045	20
GOOGLE3	9907.4	32147.584	20
AltaVista2	4703.2	7776.75565	20
AltaVista3	1686.35	3527.91228	20
Fast2	11678.9	22345.1892	20
Fast3	3988.75	13508.4384	20

Table (1) indicates that Google performs better than AltaVista and Fast in terms of number of documents retrieved per queries for the different queries length. On the other hand, Fast performs better than AltaVista in terms of the whole set and the three term queries while AltaVista performs better than Fast for the two term queries.

Appendix (2) shows the number of document retrieved by each search engines. This appendix indicates that Google performs better than AltaVista and Fast in all the cases except queries 11, 27, 29, 31, and 33. Fast achieved better than Goggle and AltaVista in these five cases.

Figures (1) indicates that, for the three search engines, the number of retrieved documents is positively related to the database size. Among these three systems, Google is the biggest in terms of the database size (2.5 billion web pages). Next is Fast (2.1 billions) and AltaVista is the smallest (500 millions)[32].

[31] Number 2 and 3 means the number of terms in the query, which refers to the query length in this study.
[32] See Search Engines Year Book (2003), Search Engines Showdown, and Search Engines Watch.

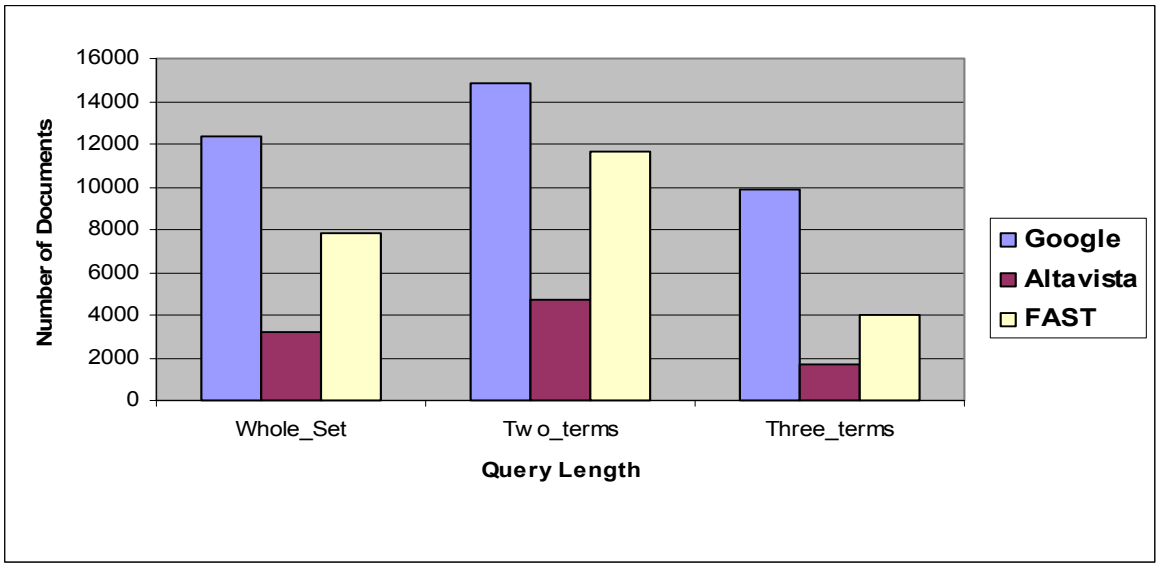

Figure (1) Number of documents retrieved

Figure (1) also supports the assumption that in general complex queries have fewer retrieved pages than simple queries. In general the average number of retrieved document for queries of two terms is 10421 pages, queries of complexity degree three terms retrieved on average 5194 pages. On average, Google ranked first, Fast second and AltaVista third in terms of the number web pages retrieved per different query length.

4.1.2 Two-Way (Repeated Measure) ANOVA Test.

The above table and figures stated that Google is performing better than the other two engines. A two-way ANOVA test has been carried out to examine if the difference between the search engines is significant. For the test to be significant the sig. value should be less than .05. Appendix 3 shows the results of this study. This test indicate significant difference between search engines in terms of the number of documents retrieved (sig. = .03 < alpha = .05) and no significant difference between search engines in terms of the query length (sig. = .317 > alpha = .05). It also indicate that there is no interaction between search engines and query length (sig. = .780 > alpha = .05).

In conclusion, although there is significant difference between search engines in terms of the number of documents retrieved per query, there is no significant difference between search engines in terms of the two level of query complexity. Google ranked first in terms of the database size and number of documents retrieved then Fast and AltaVista consecutively.

4.1.3 Document Overlapping

Figure 2 shows that there is a fair amount of overlapping in the results retrieved from the three search engines. Other studies report that overlapping among two search engines is about 10 to 15 % (Dong, 2000) and the search engines showdown web site reports estimated amount of overlapping among search engines[33]. Appendix four reports the query number, number of unique documents, document found in 1, 2 & 3 engines, then the number of overlapped document between each combination of two engines. The second and the third part of the appendix reports the number of overlapped documents. Among the 1200 (30 document x 40 queries) examined, 966 (80.5 %) are unique documents and 234 (19.5 %) are overlapped documents. Among the overlapped documents, 144 (12%) pages appeared in two engines and 45 (7.5%) pages appeared in three engines. Among the 12% which appeared in two engines 44.1% appeared in Google/AltaVista, 18.2% appeared in Google/Fast and 37.7% appeared in AltaVista/Fast.

[33] Search Engines Showdown. http://www.searchengineshowdown.com/stats/overlap.shtml

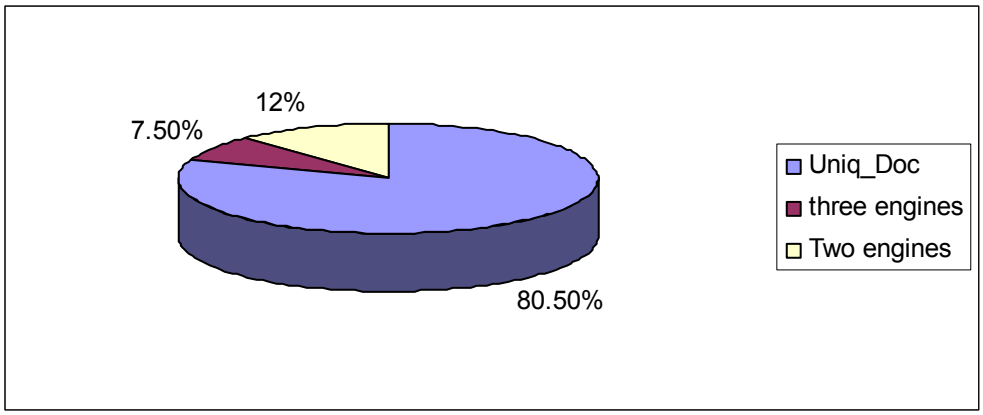

Figure 2: Percentage of Overlapped Document

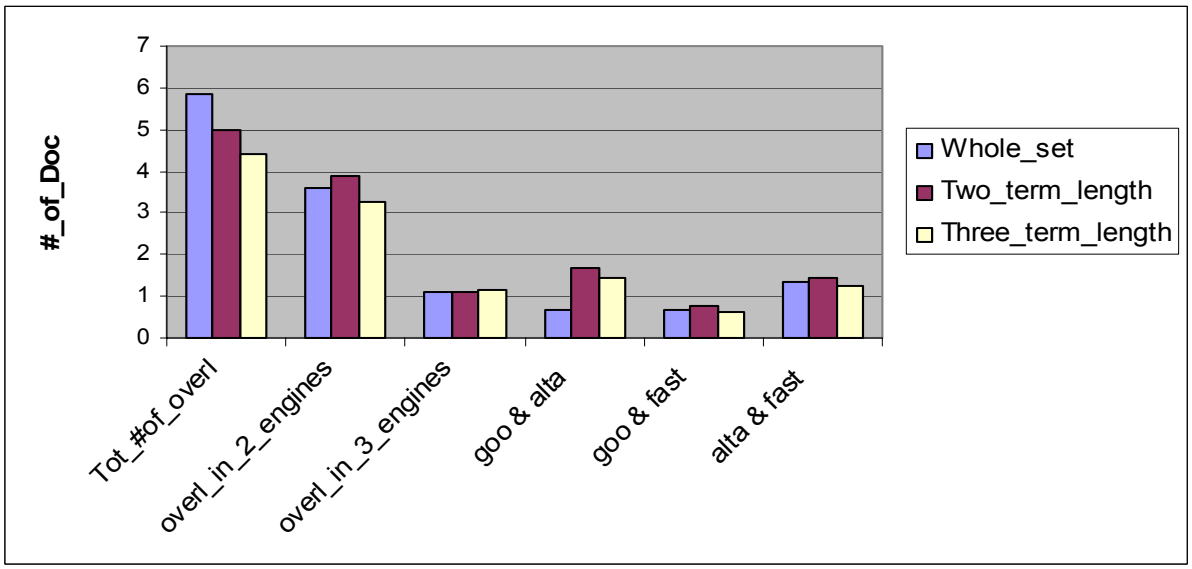

Figure 3: Overlapped Doc. across the different QL

It can be seen from figure (3) that the proportion of overlapping documents increases when the complexity degree of queries decreases. To examine if the number of query terms affects the number of overlap documents, the 2-way ANOVA test is employed for documents appearing in at least two engines result list (see appendix 4 second and third part). The ANOVA test shows significance effect for the number of engines on the overlapped documents (sig. = .000 < Alpha = .05). The test shows that there are more overlapped documents in two engines (Marginal means = 3.575) than overlapped

documents in three engines (Marginal means = 1.125). The test shows no significant effect for the number of terms on the overlapped documents (sig. = .216 > Alpha .05) and no interaction between search engines and query length (sig. = 1.0 > Alpha = .05).

4.2 Search Engines Performance.

The performance of the individual search engines can be used in determining which engine performs best in terms of precision values. The results of the individual search engines performance can then be used for ranking the search engines in terms of their performance and utilized in the combination and merging process. Two performances measures are calculated for each search engine and for each query: First Ten Precision (FTP), and precision at 11 recall cutoff levels (11P). Two-way ANOVA test has also been used to test the first hypothesis (Larger databases tend to retrieve more relevant document than smaller databases).

4.2.1 First Ten Precision (FTP)

Appendix (5) shows the three search engines precision values per query for the whole set (40 queries) and for each query length (2 terms and 3 terms).
Figures (4) shows that among the three SE, Google performed best when relevancy threshold was set to be 0.5, where this position is occupied by FAST. (See Grand Mean in Appendix 4).
To take into account the query length, the mean precision values per query length have been calculated. In both cases Google ranked first, AltaVista ranked second and Fast ranked last. Although, Fast database is larger than AltaVista database and the average number of document retrieved by AltaVista is smaller than the average number of documents retrieved by Fast, the precision performance of AltaVista is better than Fast.

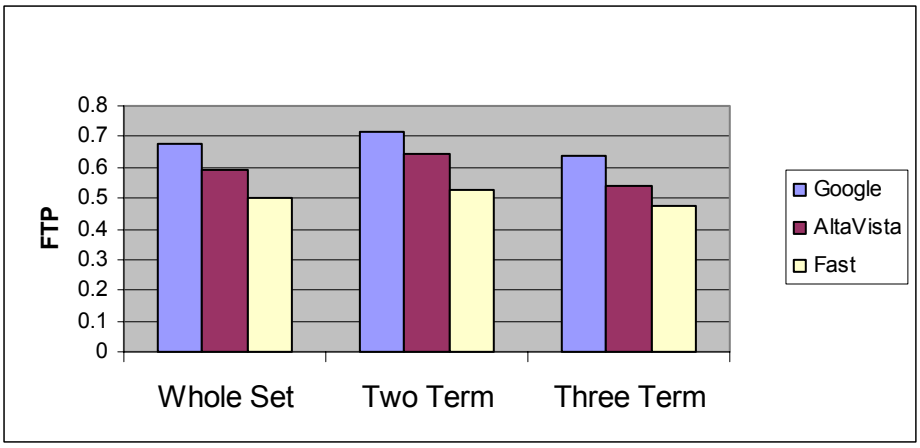

Figure 4: FTP for the Whole Set and different QL

4.2.2 Two-Way (Repeated Measure) ANOVA Test (Hyp. 1).

A test of two-way ANOVA has been conducted to examine the precision value differences between the three search engines, the two query length, and the interaction (see Appendix 6). The test for the main effect of engines indicates that there is significant difference between the three search engines, since (sig. = .000 < Alpha = .05). The difference between the search engines is for the whole set of queries. While the main effect of query length indicates no significant differences between the three engines (sig. = .183 > .Alpha = .05). To examine if the relative precision of the three engines depends on the query length, the test of interaction indicates no significant differences since (Sig. = .564 > Alpha = .05). In conclusion, although there is significant difference between search engines in terms of the their precision performance, there is no significant difference between search engines in terms of the two level of query complexity. Google ranked first in terms of the precision performance, then AltaVista and Fast consecutively. The rank order of the search engines in terms of their precision performances is not totally coincident with their performance in terms of the database size and the number of documents retrieved per query. The rank order of the search engines in terms of the

databases size and number of documents retrieved per query is Google, Fast, then AltaVista consecutively, while their rank order in terms of the precision performance is Google, AltaVista then Fast consecutively. This results suggest that larger search engines not always retrieve more relevant document than smaller search engines and indicate the importance of measuring the performance of search engines before ranking them.

4.2.3 Precision at 11 recall cutoff values (P11).

The averaged P11 data over the 40 queries are reported in table 2 and their interpolated graph are presented in figure (5). Each recall-precision average is computed by summing the precision scores at the specific recall cutoff value and then dividing by the number of queries, which is 40 in this study. By reviewing the graph, a pattern of best players similar to that with respect to the FTP can be detected. Google again appears as the winner then AltaVista is the second and Fast appears as the last. Google always performs above the average, AltaVista performs below the average in the first five cases and approximately as the average in the last six cases, while Fast performs below the average in all the cases. The 11 cutoff values can be divided into three different recall range, the first one is the high precision range which is from 0 to 0.2; the second one is the middle recall range and covers 0.2 to 0.8; the third one is the high recall range which covers from 0.8 to 1. If the cutoff values examined in this manner. The same performance pattern is appears, since Google outperform AltaVista and Fast. While AltaVista outperforms Fast in all the cases except the first spot where AltaVista and Fast looked tied for this spot.

Table 2: Search Engines P11 over 40 queries

Recall	Google	Altavista	FAST	Average
0	0.89	0.7275	0.7275	0.781667
0.1	0.796668	0.705	0.655	0.718889
0.2	0.794995	0.682495	0.644995	0.707495
0.3	0.788125	0.663125	0.594375	0.681875
0.4	0.755	0.6525	0.5875	0.665
0.5	0.737918	0.646255	0.575003	0.653058
0.6	0.730003	0.635355	0.544244	0.636534
0.7	0.704688	0.609063	0.536875	0.616875
0.8	0.6907	0.608613	0.52611	0.608474
0.9	0.6825	0.60375	0.50625	0.5975
0.1	0.6825	0.60375	0.50625	0.5975

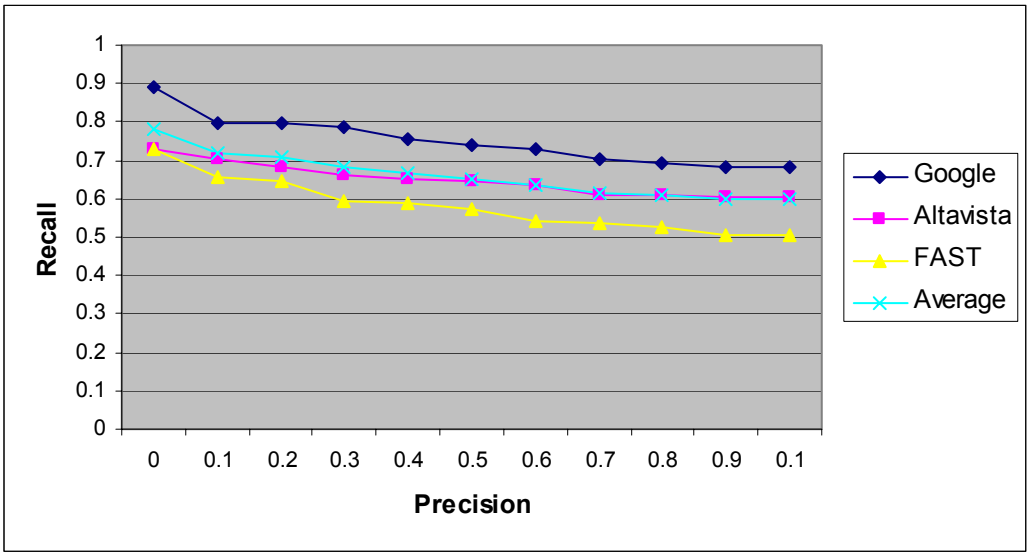

Figure 5: Precision at 11 point cutoff recall

The 11P can also be averaged over each query length. Figure 6 and 7 (see appendix 7) shows that the same pattern appears for the search engines performance over the different query length and the different range of recall precision plots.

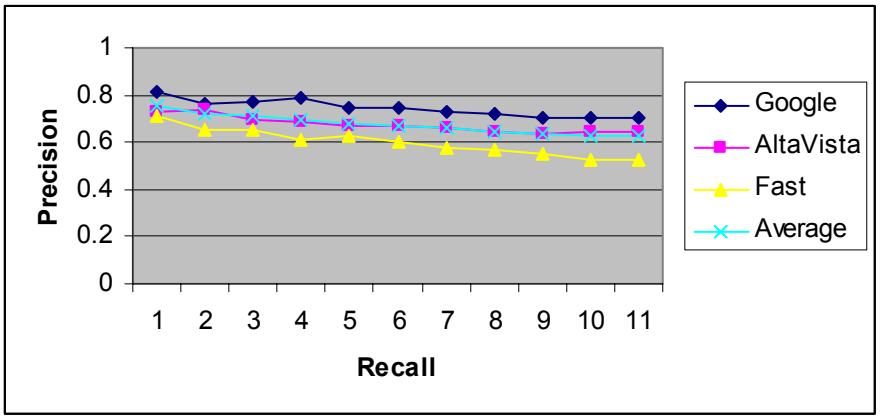

Figure 6: 11P Recall-Precision for QL-2

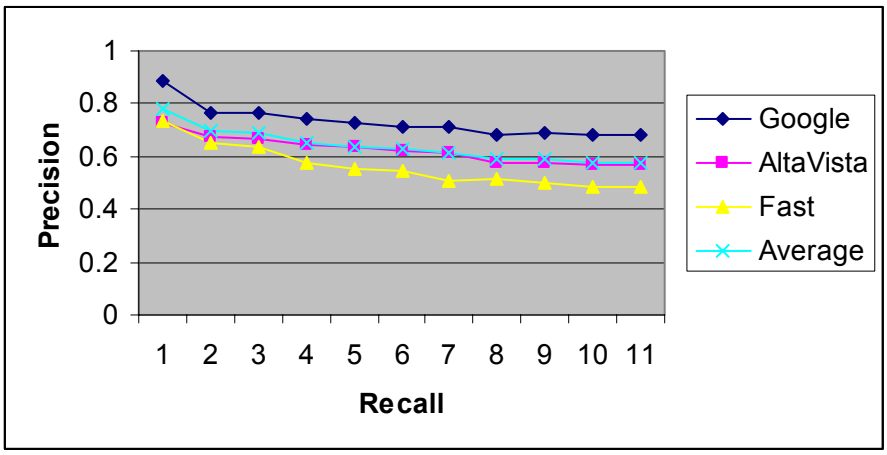

Figure 7: 11P Recall-Precision for QL-3

Figure 8 (see appendix 7) compares 11P for the two complexity levels. The figure shows that the performance of the query complexity 2 terms outperforms the performance of the 3 terms in all the cases except the first spot where the 3 terms outperforms the 2 terms.

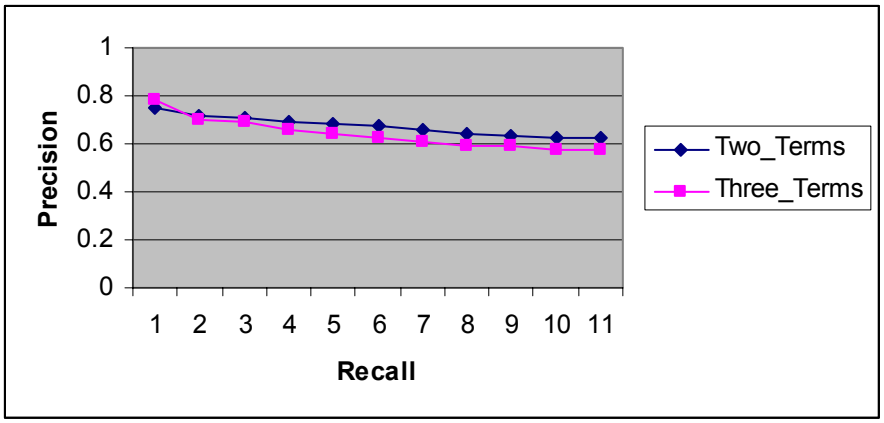

Figure 8: 11P Recall-Precision over 2-QL and 3-QL

4.3 Performance of Multiple Combinations

4.3.1 Procedure

This section addresses the multiple combinations run conducted at 2-way and 3-way levels. There are three functions used for combining and merging the results which are: interleave, rank similarity and global similarity. Each combination has been formed according to these three functions. For example there are three possible two combinations include: Google – AltaVista; Google – Fast and AltaVista – Fast. Each one of these combination has been formed according to the three merging functions. So for Google – AltaVista combination, there is three possible variations including Google – AltaVista according to the interleave function, Google – AltaVista according to the rank similarity function, and Google – AltaVista according to the global similarity function. The situation is the same for Google – Fast and AltaVista – Fast. For the three way combinations there is only one possibility for the search engine combination which is Google – AltaVista – Fast. This combination has been also formed according to three merging functions. This section first treats each combination separately, with the concern centered on comparing the averaged performance over 40 queries and locating the best

runs. So for the two combinations, it defines the best combination method for Google-AltaVista and compares it with the best combination method for Google – Fast and AltaVista – Fast to define the best combination and the best method of combining search results.

4.3.2 Two-Way Combination Performance.

This section compares all the possible 2-way combination to indicate the best run.

4.3.2.1 Google – AltaVista:

In terms of precision performance for Google – AltaVista combination, it can be seen from the figure (9) that the global similarity function and the rank similarity function perform slightly better than the interleave function, while the Global similarity performs slightly better in terms of QL 2 and performs the same as the rank similarity in terms of the QL3. (See Appendix 8)

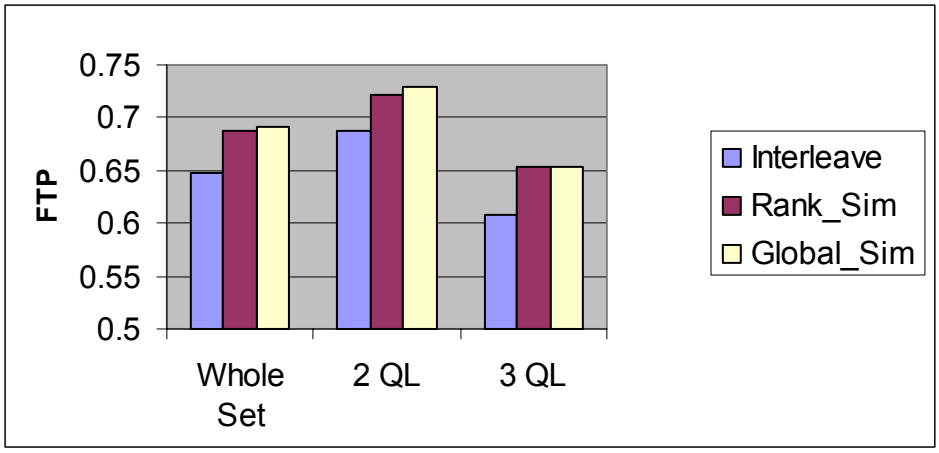

Figure 9: FTP Google - AltaVista Combination

To test if there is a significant difference among the three methods used for combining the search results, a two-way ANOVA test has been conducted. The test shows that there is significant difference among functions, since (sig. = .018 < Alpha = .05), no significant difference in terms of the different query length (sig. = 209 > Alpha = 0.05) and no

significant interaction (sig. = 925 > Alpha = 0.05) (see appendix 9).

In conclusion, the global similarity function performs slightly better than the rank similarity and the interleave functions for this run. The marginal means for the three functions in this run are .691, .687, and .648 respectively.

4.3.2.2 Google – Fast:

It can be seen from figure (10) that the rank similarity function performs better than the interleave function which performs better than the global similarity function for this run. (See Appendix 10)

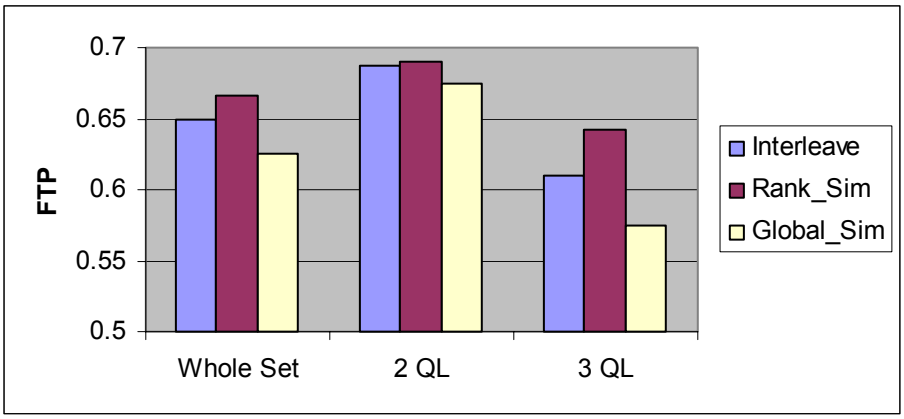

Figure 10: FTP Google-Fast Combination

The two-way ANOVA test shows no significant difference among the function in terms of mean differences (0.189), query length (0.207) and no significant interaction (0.663) (see appendix 12).

In conclusion, by looking at the marginal means, the rank similarity function performs slightly better than the Interleave and the global similarity functions for this run. The marginal means for the three functions in this run are .666, .649, and .625 consecutively.

4.3.2.3 AltaVista – Fast:

Figure (11) shows that the global similarity functions performs better than the

Interleave functions which performs better than the rank similarity function for this run (see appendix 12).

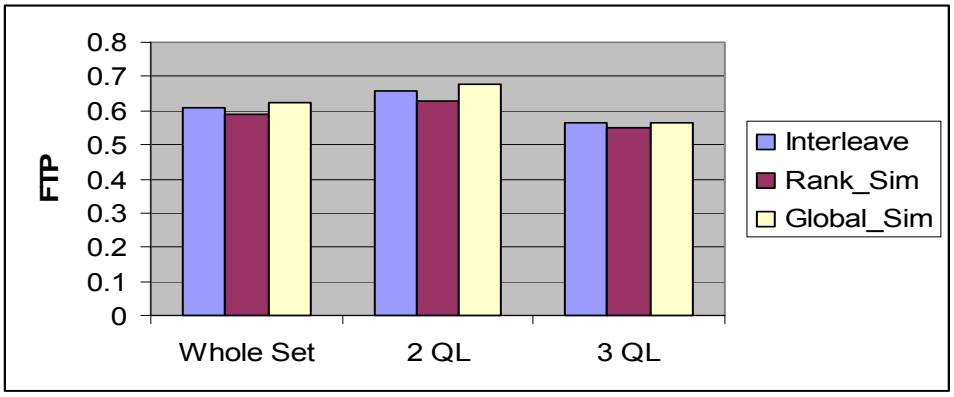

Figure 11: FTP AltaVista-Fast Combination

The two-way ANOVA test shows no significant difference among the function in terms of mean differences (sig. = 083 > Alpha = 0.05), query length (sig. 0.118 > Alpha = 0.05) and no significant interaction (sig. = 605 > Alpha = 0.05) (see appendix 13).

By comparing the marginal means of three functions for this run, it can be seen that the global similarity wins the run, while the Interleave function performs better than the rank similarity function.

In conclusion, for the FTP of the 2-way combination, the global similarity function wins the run twice and rank similarity function wins the run one time, while the Interleave function performs worse than them in the three runs. The 2-way ANOVA test shows only significant difference in terms of the mean difference for Google-AltaVista combination, while shows no significant difference in terms of the mean differences of the three function, the query length and the interaction.

4.3.2.4 Precision at 11 recall cutoff values (P11).

This section compares the performance of the three best 2-way combinations indicated in the previous section. This run will use the best performance of each combination. So for

Google – AltaVista and AltaVista – Fast, the global similarity combination has been used, and for Google – Fast, the rank similarity combination has been used.

Figure (12) shows that Google – AltaVista and Google – Fast combination performs better than AltaVista – Fast in the 11 points. Table (3) shows that Google – Fast combination which utilized the rank similarity function performs better in the upper tail (positions 0, 1, 2, and 3) while Google – AltaVista which utilized the global similarity function performs better in the middle and lower tail of the 11 point cutoff recall distribution (positions 4, 5, 6, 7, 8, 9, and 10). The table shows also that on average the global similarity function performs better than rank similarity function for the best runs comparison.

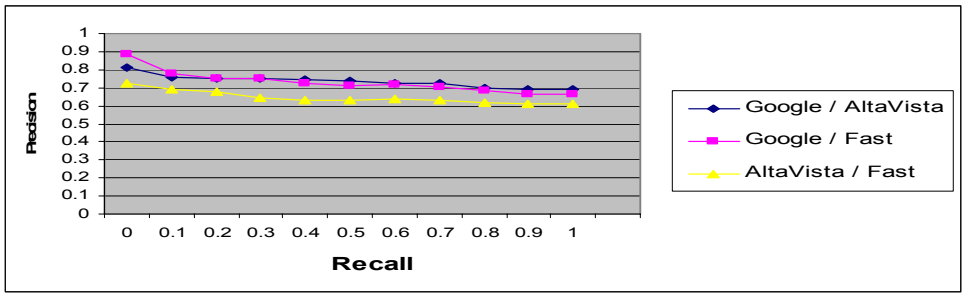

Figure 12: 11P for the Best 2-way combination

Table (3) 11P for the Best 2-way combinations

Recall	Google / AltaVista	Google / Fast	AltaVista / Fast
0	0.8125	0.8875	0.725
0.1	0.75625	0.78125	0.69375
0.2	0.74975	0.754	0.6745
0.3	0.750625	0.752	0.64525
0.4	0.745	0.725	0.6325
0.5	0.73775	0.71	0.633325
0.6	0.7275	0.71825	0.635425
0.7	0.72175	0.704	0.6315
0.8	0.69675	0.6836	0.61795
0.9	0.68875	0.66625	0.61375
1	0.69125	0.66625	0.61375
Average	0.734352	0.731645	0.646973

In conclusion, the data suggest that when combining higher performance search engines with lower performance search engines, the performance of the combination differs according to the performance of the combination method and the performance of the search engines. On average the global similarity function performs better than the rank similarity function which performs better than the Interleave function. The explanation for that is the global similarity function pops the overlapped document up to the list of the retrieved document, since it summed the overlapped document scores up. It is indicated in section (4.6) that search engines retrieved more relevant overlapped documents than irrelevant which increase the probability of achieving higher precision values for the first 10 precision.

4.3.3 Three-Way Combination Performance.

This section will compare the different combination for the three way in terms of the first ten precision (FTP) and precision at 11 point cutoff values (11P).

4.3.3.1. First Ten Precision

Each run at the 3-way combination has only one scheme include the three search engines. This run compares the FTP for the three combination functions: Interleave, Rank Similarity, and Global Similarity.

Figure (13) shows that on average, the global similarity functions performs better than the rank similarity functions which performs better than the Interleave function for this run (see appendix 14).

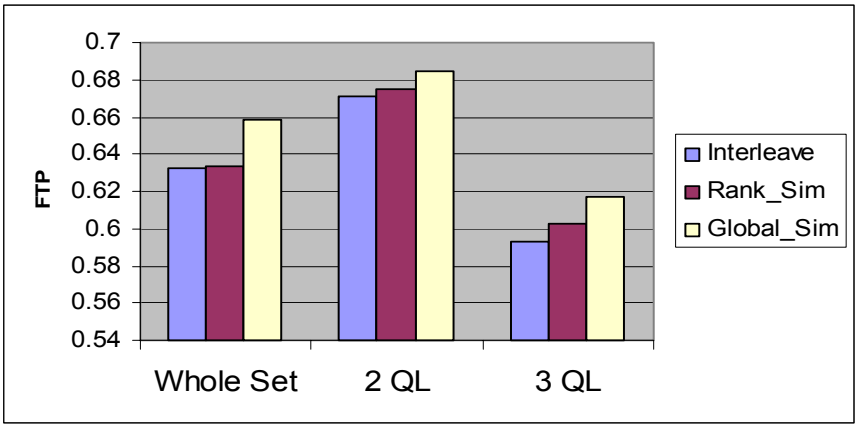

Figure 13: Three-Way Combination Performance

Although the 2-way ANOVA test shows no significant differences among the functions in terms of the mean differences (sig. = 0.309), the query length (0.197), and the interaction (0.921), the global similarity function had trivial advantage over the other two functions in terms of the marginal means (see appendix 15).

4.3.3.2 Precision at 11 recall cutoff values (P11).

This run compares the performance of the three way combination using the 11 point recall cutoff values. Figure (14) suggests that the global similarity function works better for the upper tail of the distribution while no significant difference appears in the middle and the lower tail of the distribution.

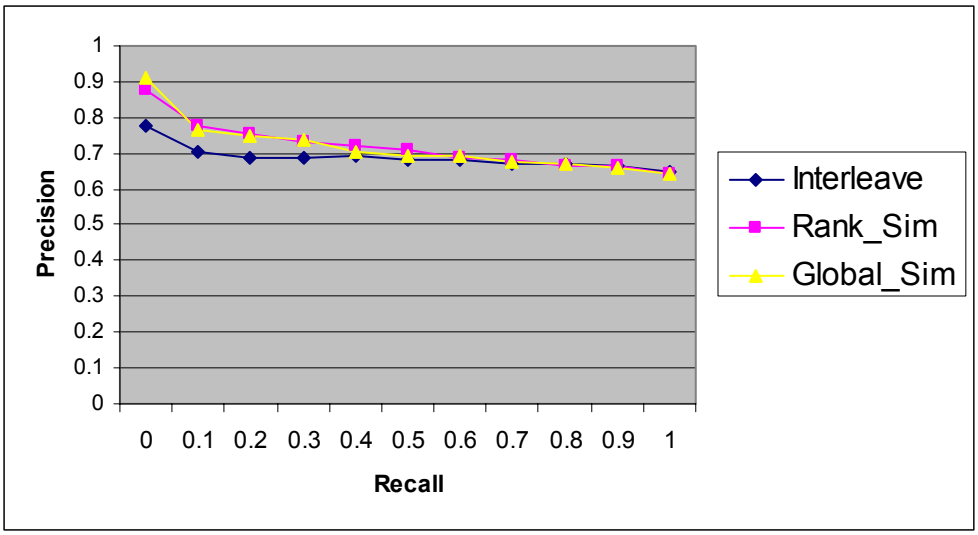

Figure 14: 11P for 3-Way Combinations

Table (4) shows that the global function wins the run in positions (0, .3, .6, and .8), the rank similarity function wins in positions (.1, .2, .4, .5, and .7), while the rank similarity function wins the run in positions (.9, and 1). It is clear that the global similarity function wins the run in terms of the precision performance but doesn't perform well in terms of the 11P because this function pop the overlapped documents up to the list which might effect positively in the top of the distribution and effect negatively in the middle and lower tail of the distribution.

Table (4) Precision at 11 Recall Cutoff Values

Recall	Interleave	Rank_Sim	Global_Sim
0	0.775	0.875	0.9125
0.1	0.7015	0.778	0.765875
0.2	0.685125	0.756875	0.74825
0.3	0.68825	0.731	0.736
0.4	0.69125	0.721	0.7045
0.5	0.6835	0.71	0.69
0.6	0.6815	0.687	0.69155
0.7	0.6725	0.68	0.675
0.8	0.669375	0.6661	0.6701
0.9	0.66525	0.665	0.657
1	0.650742	0.643933	0.643667

Table (5) Summary for the Multiple Combinations Results

Combination	Run Winners
Google – AltaVista	Global Similarity
Google – Fast	Rank Similarity
AltaVista – Fast	Global Similarity
Google – AltaVista - Fast	Global Similarity

In conclusion, for the FTP of the 2-way and the 3-way combinations, the global similarity function wins the run and the rank similarity function ranked second, while the interleave function ranked third for the two and three way combinations. The 2-way ANOVA test shows only significant difference in terms of the mean difference for Google-AltaVista combination for the benefit of the global similarity function, while shows no significant difference in terms of the mean differences of the three functions, the query length and the interaction.

4.3.3.3 Best Combination Performance

The individual search engines performance shows that Google performs better than the other two engines. The 2-way combination indicates that when Google and AltaVista are combined according to the global similarity function they perform better than the other two way combinations. The three way combination shows that the global similarity function performs better than the other two functions. To identify if the combination performs differently than the best individual engines performance, the best runs from the individual, 2-way and 3-way combination have been compared. Figure (15) shows that the three runs overlapped in their performance to the level that it is difficult to identify which one wins this run.

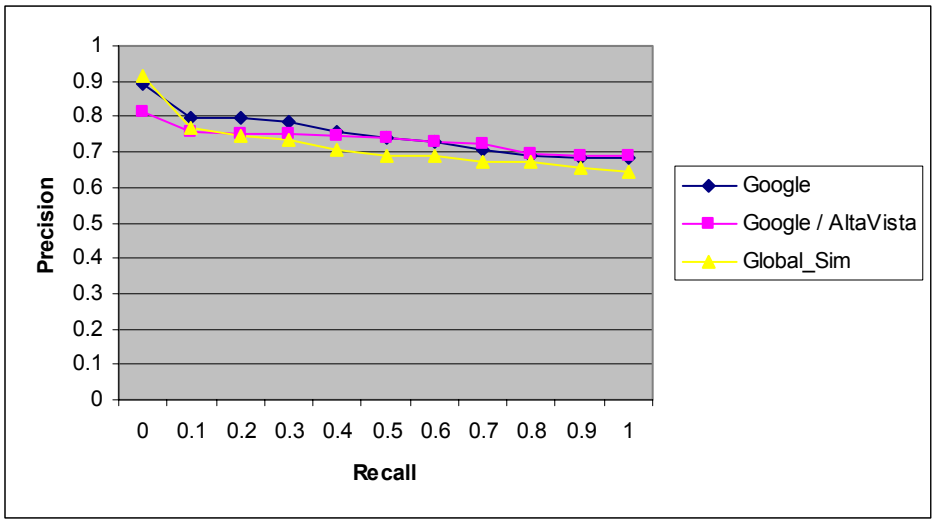

Figure 15: Best Performance for 11P Comparison

Table (6) shows that on average Google performs better than the two way and three way combinations but still the difference is not significantly large. The table also shows that the 2-way combination performing better than the 3-way combination but the difference is still not significantly large.

Table 6: Precision at 11 Point Recall Cutoff Values

Recall	Google	Google / AltaVista	Global_Sim
0	0.89	0.8125	0.9125
0.1	0.796668	0.75625	0.765875
0.2	0.794995	0.74975	0.74825
0.3	0.788125	0.750625	0.736
0.4	0.755	0.745	0.7045
0.5	0.737918	0.73775	0.69
0.6	0.730003	0.7275	0.69155
0.7	0.704688	0.72175	0.675
0.8	0.6907	0.69675	0.6701
0.9	0.6825	0.68875	0.657
1	0.6825	0.69125	0.643667
Average	**0.750281**	**0.734352**	**0.717677**

4.4 Overlapping Documents Relevancy.

Fox and Show (1994) data fusion functions and the global similarity function are based on the assumption that IR systems tend to retrieve more relevant overlapped documents than irrelevant documents. Figure (16) shows that search engines tend to retrieve more relevant overlapped documents then partially relevant and irrelevant documents.

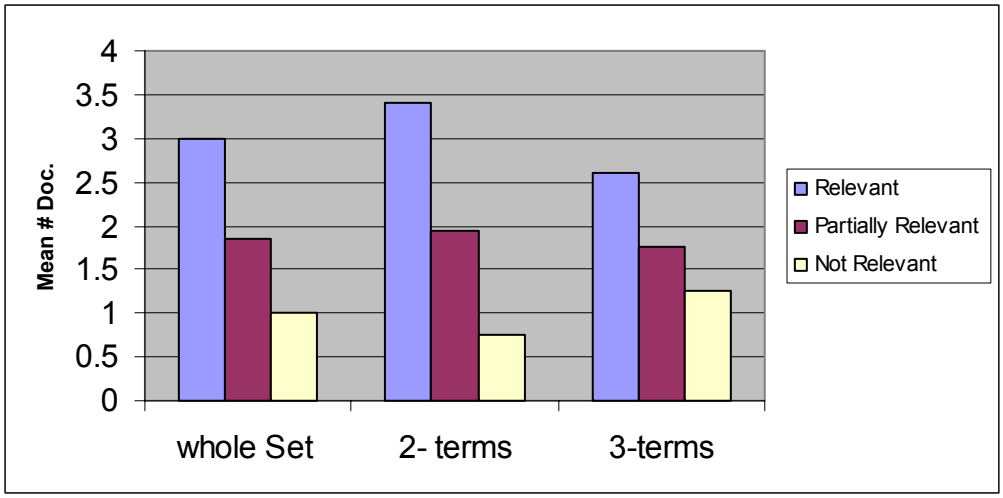

Figure 16: Degree of Relevancy among Overlapped Documents

This section tests this assumption to indicate the importance of the overlapped documents. The 2-way ANOVA test shows significant difference for the degree of relevance (sig. = .000 < Alpha .05), no significant effect for the number of terms in the number of relevant overlapped documents (sig. = .660 > Alpha .662) and no interaction. The ANOV test indicates that the mean number of relevant documents is statistically different then the mean number of irrelevant document. It also shows that, while the mean number of relevant document is descriptively larger than the mean number of partially relevant documents, it is not statistically different.

4.5 Performance of the Merging Schemes (Hy. 3)

Three methods have been used for merging the search results in the level of 2-way and three way combinations. The first 10 documents have been used for comparing the three methods of merging for the 2-way combinations and the first 15 documents have been used for the 3-way combinations. This section compares the performance of the three merging methods to determine the best way of merging multiple search results for metasearch engines

4.5.1 Merging Two Engines:

To compare the performance of the three merging methods in terms of 2-way combinations, each merging method has been used for sorting the search results for the three possible combination of the 2-way combination (Google – AltaVista, Google – Fast and AltaVista – Fast) then the average of the relevancy score has been used to normalize the ranking scores. For example the average score of the three possible 2-way combinations has been calculated for the first 10 document in the rank order according to each method.

Figure (17) and table (6) shows that none of the merging functions perform significantly different than the others in terms of ranking the search results. Although the Interleave functions performs better in four positions (1,6,7,9), the rank similarity function performs better in three positions (3,4,5) and the global similarity function performs better in three positions (2,8, 10).

Figure 17: Merging Two Engines

Table 7: Merging Two Engines

Rank Position	Interleave	Rank_Sim	Global_Sim
1	3	2.97	2.83
2	2.48	2.53	2.54
3	2.53	2.62	2.58
4	2.63	2.56	2.51
5	2.42	2.45	2.43
6	2.47	2.26	2.61
7	2.47	2.6	2.46
8	2.06	2.25	2.37
9	2.43	2.21	2.26
10	2.07	2.23	2.17

A test of 2-way ANOVA has been conducted to examine if there is a significant difference among the three merging methods.(see Appendix 16). The test for the main effect of the merging functions shows no significant difference among the three merging functions, since (sig. = .0.111 > Alpha = .05). The test for the main effect of query length shows significant effect for the query length on the merging function performance (sig. = 0.04 < Alpha = 0.05). The test of interaction shows that the relative precision of the merging function does not depend on the number of query terms (sig. = 0.280 > Alpha = 0.05).

4.5.2 Merging Three Engines

Each merging function has been used for sorting the combined results for the three search engines then the average performance of the three merging function for the first 15 documents has been used for comparing the merging functions performance. Figure (18) and table (7) shows that none of the merging functions perform significantly different than the others in terms of ranking the search results. Each function performs better than the others in five positions. Table (7) shows that the Interleave function performs better in positions (1,5,6,13 and 14), the rank similarity function performs better in positions (3, 4, 8, 10, and 12) and the global similarity function perform better in positions (2, 7, 9, 11, 15).

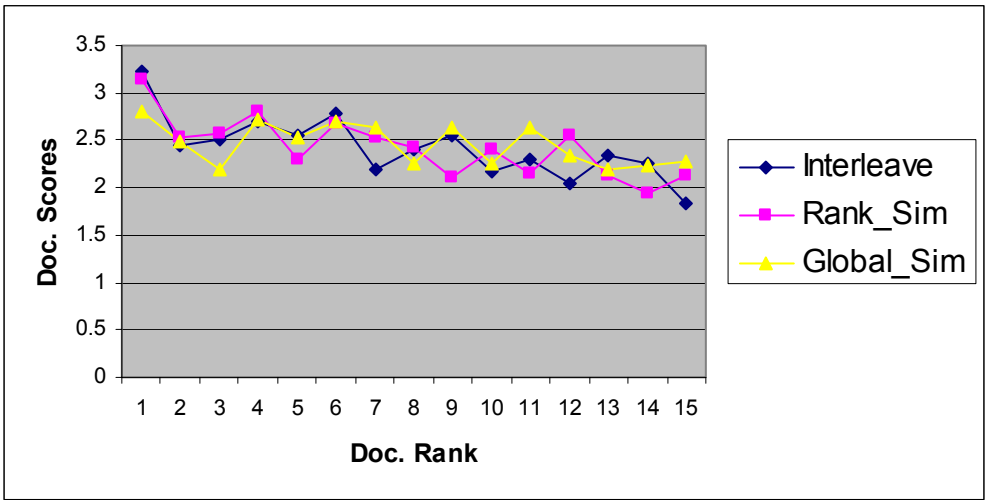

Figure 18: Merging Three Engines

Table 8: Rank Score for the three Functions

Rank Position	Interleave	Rank_Sim	Global_Sim
1	3.23	3.15	2.8
2	2.45	2.53	2.48
3	2.5	2.58	2.2
4	2.7	2.8	2.73
5	2.55	2.3	2.53
6	2.78	2.68	2.7
7	2.2	2.53	2.63
8	2.4	2.42	2.25
9	2.55	2.1	2.63
10	2.17	2.4	2.25
11	2.3	2.15	2.63
12	2.05	2.55	2.33
13	2.33	2.13	2.2
14	2.25	1.93	2.23
15	1.83	2.13	2.28

A test of 2-way ANOVA has been conducted to examine if there is a significant difference among the three merging methods.(see Appendix 18). The test for the main effect of the merging functions shows no significant difference among the three merging functions, since (sig. = .0.781 > Alpha = .05). The test for the main effect of query length shows significant effect for the query length on the merging function performance (sig. = 0.000 < Alpha = 0.05). The test of interaction shows that the relative precision of the merging function does not depend on the number of query terms (sig. = 0.937 > Alpha = 0.05).

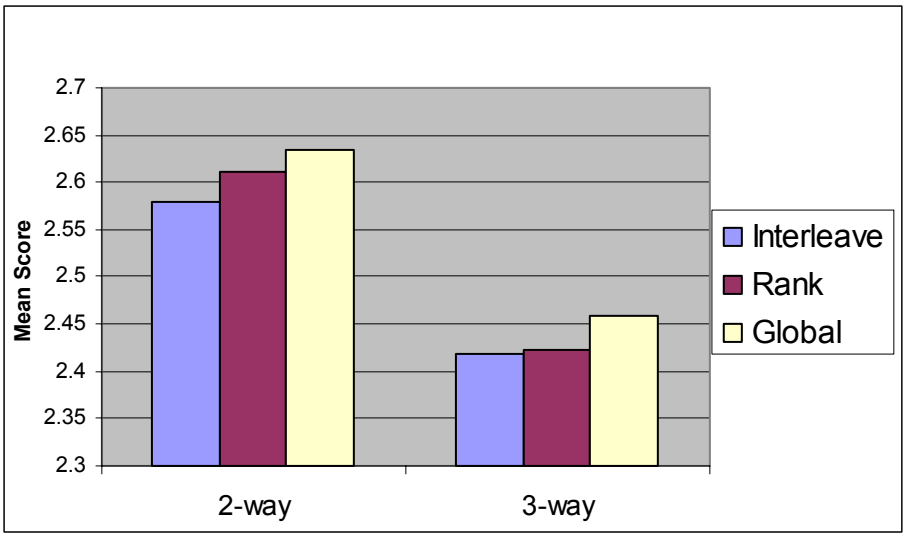

Figure 19 Average performance of the Merging Function

Figure (19) shows that on average global similarity function performs better than the other two function in terms of ranking the final list.

In conclusion, although the test of the main effect of the merging functions shows no significant difference, the mean of each merging function performance shows that the global similarity function is working slightly better than the other two functions in terms of the merging score for the two and the three ways combination.

CHAPTER FIVE: SUMMARY AND CONCLUSION

5. Introduction

This section first draws together the rest of the thesis, summarizes what has been achieved and states the conclusion. It then explores area where future work is required.

5.1 Context of the Study:

This study tries to build a framework that can be utilized by metasearch engines developers for merging multiple search results retrieved from distributed search engines. The study focuses on the three major steps in the data fusion process including: databases ranking, results combination, and results merging. Forty real queries have been utilized for ranking the selected databases, combining and merging the search results based on three heuristic solutions. Real users have been involved in the relevance judgment process using a scale of five points to evaluate and rank the retrieved web pages.

5.2 Discussion and Analysis

In chapter one, the study states that metasearch engines are one of the powerful and important tools for searching the web. In chapter two, the literature suggest that merging multiple search results for metasearch engines is an area need further investigation. In chapter three, three hypotheses were set up to see if the combination and merging functions improve the performance of the individual search engines. Hypothesis one tests if number of search engines with different databases size has been selected for metasearch engines does their performance differ significantly in terms of different query length and database size. The major goal of the this hypothesis is to identify the best rank order of the search engines as a major part in the merging process.

To test this hypothesis it was important to identify if the database size affect on the number of documents retrieved per query. The results shows that there is significant difference between search engines in terms of the number of documents retrieved per query, there is no significant difference between search engines in terms of the two level of query complexity. Google ranked first in terms of the database size and number of documents retrieved then Fast and AltaVista consecutively.

For hypothesis one, the study indicates that, although there is significant difference between search engines in terms of the their precision performance (FTP), there is no significant difference between search engines in terms of different level of query complexity. Descriptively, shorter queries perform better than lager query but statistically this different is not significant. The rank order of the search engines in terms of their precision performances is not totally coincident with their performance in terms of the database size and the number of documents retrieved per query because in terms of the databases size and number of documents retrieved the rank order of the engines is Google, Fast, and AltaVista, consecutively, while their rank order in terms of the precision performance and 11 point recall cutoff values is Google, AltaVista, Fast consecutively. This suggests that lager search engines are not always performing better than smaller engines. This indicates the importance of the search engines performance evaluation in the metasearch development process because the size of the databases may not be relevant factor for ranking the search engines. The database size should be compared to the engines performance in order to set up the best rank order for any combination and / or merging process.

Hypothesis two has been set up to test the best combination for the selected engines.

Three combination functions used to form the final lists of two ways and three ways combinations. The precision performance of these functions used for comparing and identifying the best combination. The comparison indicate that for the FTP of the 2-way combination, the global similarity function wins the run twice and rank similarity function win the run one time, while the Interleave function performs worse than them in the three runs. For the three way combination, although the 2-way ANOVA test shows no significant differences among the functions in terms of the mean differences, the query length, and the interaction, the global similarity had trivial advantage over the other two functions in terms of the marginal means The explanation for this is that the global similarity function performs better because it takes into consideration the overlapped documents. This function pops the overlapped documents up the entire unique document, so they appear in the first 10 of any combination which indicates that overlapped document play important role in the combination process. The test of overlapping document importance indicates that that the mean number of relevant documents is statistically different than the mean number of irrelevant documents. It also shows that while the mean number of relevant documents is descriptively larger than the mean number of partially relevant documents, it is not statistically different. This suggests that search engines tend to retrieve more relevant overlapped documents than partially relevant and not relevant overlapped document.

When the performance of the 11 point recall cutoff precision is used for comparing the best individual run with the best two and three way combinations the results shows that the performance does not change significantly when multiple sources are combined together. The other side effect of this observation is that when the best combination

method is used and the best search engines are selected for developing a metasearch engines the performance of the combination will not negatively be affected by the lower ranked engines as long as the combination method will pop the irrelevant document out of the first 10 documents.

Hypothesis three has tested the effect of the three merging methods in the final ranked lists. For the two way combination, the first 10 documents have been used for comparison and for the 3-way, the first 15 documents have been used for comparison. The test of the main effect of merging function shows no significant effect for the two ways and three ways merging. Each function performs better than the others in some position of the distribution of the first 10 or 15 documents. Although on average the global similarity function performs slightly better than the other two functions, the statistical test shows that the three functions perform approximately the same in terms of ranking the final list of documents with minor differences. One explanation for this observation is that the three merging functions are based on the same procedural logic which is that search engines should be ranked in terms of their search performance and in terms of their database size or number of users. Therefore, documents appeared in the highly ranked engines will always proceed documents appeared in lower ranked engines.

5.3 Summary and Conclusion

The WWW provides a convenient space for people to publish and disseminate information. Search engines are designed to capture, index and provides means for finding this information. It is often observed that search engines have a very limited coverage of Web documents in their collections. To overcome this shortcoming, solutions found in data fusion approaches which have been demonstrated as useful in the

traditional IR area. This approach aims to improve search performance through the combination of multiple sources results. Metasearch engines represent the principal application of this approach in terms of web retrieval.

This dissertation was designed to provide a framework for metasearch engines developers in their building process. The major three steps of the building process have been examined including: Search engines selection and ranking, multiple results combination, and merging the search results. Forty real general query samples have been organized into two groups, each group represent a level of query complexity. A five relevance scale have been used to define the document relevancy through real users have real information needs represent their academic interests. Three major hypotheses were set up to frame the major steps of the metasearch building process.

The first hypothesis assumed that larger search engines tend to retrieve more relevant documents than smaller search engines. The results show that larger search engines do not always retrieve more relevant documents. This provides an important hint for metasearch developers that they can not depend on database size in their selection process but they have to evaluate the performance of the search engines using rigorous measurements known in IR.

The second hypothesis was set up to test the effect of multiple combinations using different combination methods on the precision ratio. The results show that, for 2-way combination the global similarity function performs better than the other two methods of combinations. For the 3-way combination the global similarity function performs slightly better than the other two functions although the ANOVA test show neither significant effect for the function nor the query length on the final lists. The explanation for this

observation is that the three tested functions are based on the same major principle. This principle makes any merging process depends on the search engines rank.

The global similarity function provides higher rank in the final lists for the overlapped documents. This study proved also that search engines tend to retrieve more relevant overlapped documents than irrelevant documents.

Hypothesis three tested the effect of the three merging function on the final ranked list. Results show no significant effect for the merging function. The three functions perform approximately the same with trivial difference. On average the global similarity function performs slightly better than the other two functions in merging the final lists of two and three way combination but the statistical test shows no significant difference among the functions.

This study shows that metasearch engines developers should evaluate the performance of the search engines before adding them to the list of their databases. It also shows that overlapped documents play an important role on the combination process because they improve the precision ratio of the combined list. Any merging function based on simple solutions should give more rank for the overlapped documents than the unique documents. The global similarity function performs better than the rank similarity function which in turn performs better than the Interleave functions in terms of the combination and merging process.

5.4 Future Works

This section presents the areas in which future researches on the Web retrieval and merging multiple search results are required. These researches will help in improving the understanding of Web retrieval and result merging. Some of these

researches could provide more specific details about the merging issue which could be utilized by metasearch engines developers.

First, more could be done in the search engines selection stage, where search engines could represent specific domains and others could represent general collections.

Second, more could be done in query selection step by comparing more levels of complexity and different query structure to examine the effect of these complex queries on the performance of the search engines. This step is very important for metasearch engines developers because they have to map the user queries to each search engines searching capabilities.

Third, further works are needed to provide a baseline for measuring search engine recall. Since search engines tend to return huge numbers of hits for queries, web search engine evaluation studies have had to give up recall. However, the recall measurement is still important, especially for complex queries, which usually retrieve small number of hits. In this case, recall could be measured practically to test the completeness of coverage of web database by measuring to what extent search engines retrieved all the expected documents for this complex query. Another method for measuring recall depends on searching for specific document across the search engines, especially the most recent documents posted to the web.

Fourth, since metasearch engines combine results from different search engines and each search engine has special scheme for the document representations which mostly depends on the extraction from the text to the keyword which represent the query. This makes the same document has different representations and the metasearch has to chose one by default which might not be the best representation.

So some work need to be done to find methods for representing documents retrieved from different search engines.

Fifth, at least two improvements could be to improve the relevance evaluation. First two assessors could be asked to evaluate the same document and if they agree their score could be used as it is. If they don't agree a third person could be asked to evaluate the same document and his score could be used as the median score. Some of the documents may have been removed during the manipulation process though when assessors try to visit this document they find the link is dead, which would not be considered a search engine fault. So another run could be made especially for this document and if it is not available, it takes a score of zero because the link to it is not available any more on the web. The solution for this problem is to download the document for evaluation in the search time.

Sixth, more work could be done to compare the performance of the simple merging solution with the document fetching method (see chapter 3). These fetching methods depend on downloading the retrieved document on the local server and reanalyzing them for retrieval. This procedure could be compared with the procedure presented in this study for merging the search results in terms of time lag between query submission and result presentation beside the performance issues.

Developing methods for merging multiple web search results will lead to better usage of the information provided by web search engines and this will promote research and development of new tools that best serving user needs. Although, this study focused on academic queries, other type of information needs could be tested using the same procedure such as business and commercial needs.

References:

1- Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., & Raghavan, S. (2000). Searching the Web. Stanford University Technical Report 2000-37. [Online] Available at *http://dbpubs.stanford.edu/pub/2000-37*

2- Aslam, J & Montague, M. (2000) Bayes Optimal Metasearch: Probabilisitic Model for Combining the Results of Multiple Retrieval Systems (Poster Session). Annual ACM Conference on Research and Development in Information Retrieval. Proceeding of the 23rd Annual International Conference on Research and Development in Information Retrieval. pp. 397-381.

3- Aslam, J. & Montague, M. (2001). Relevance Score Normalization for Metasearch. . Proceedings of 10th Annual International ACM/SIGR Conference on Research and Development in Information Retrieval. Atlanta, Georgia: pp. 427-433.

4- Belkin, N., Kantor, P., Fox, E., & Show, J. (1995). Combining of Evidence of Multiple Query Representation for Information Retrieval. Information Processing & Management. 31(3): pp. 431-448.

5- Bernadette, M(ed.). (1998). Aggregation and Fusion of Imperfect Information. New York: Physica-Verlag: 278p.

6- Big Search Engine Index. *http://www.search-engine-index.co.uk* Available Online: September, 5, 2002.

7- Bokor, G. (1999). Terminology Search on the World-Wide Web. Translation Journal. 3(1). *http://accurapid.com/journal/07search.htm*

8- Bradley (1998). Multi-Search Engine: A Comparison. *http://www.philb.com/msengine.htm*. ERIC Abstract.

9- Brinkley. M. & Bure (1995). Information Retrieval from the Internet: An Evaluation of the Tools. Internet Research: *Electronic Networking Application and Policy*. 5(3), 3-10.

10- Callan. J & Connel, M. (2001). Query- based sampling of text databases. *ACM Transaction on information systems,* 19(2): pp.97-130.

11- Chignell, M, Gwizdka, Boder, C. (1999) Discriminating Meta-Search: AFramework for Evaluation. *Information Processing and Management*. 35(3): 337-62.

12- Chu, H., & Rosenthal, M. (1996). Search Engines for the World Wide Web: A Comparative Study and Evaluation Methodology. *In S. Hardin (Ed), Proceedings of the 59th Annual Meeting of the American Society for Information Science*. (pp.1127-135), Medford. NJ: American Society for Information Science.

13- Clarke, S.J., & Willett, P. (1997). Estimating the recall performance of Web search engines. *Aslib Proceedings*, 49(7), 184-189.

14- Cleverdon, Cyril W. (1991). The significance of the Cranfield tests on index languages. *Proceedings of 14th Annual International ACM/SIGR Conference on Research and Development in Information Retrieval*, ed Abraham Bookstein, Yves Chiaramella, Gerard Salton, Vijay V. Raghavan, Chicago, pp. 3-12

15- CommereceNet/ Nielsen (2003, March, 10) Worldwide Web Internet Population. Available Online. www.commerce.net/research/stats/wwstats.html

16- Courtois, M. (1996) Cool Tools for Web Searching: An Update, *Online*, May/June, 29-36.

17- Courtois. M., Baer,W. & Stark.M. (1995). Cool Tools for Searching the Web. *Online*, 19(6), 14-23.

18- Dennis, S., Bruza, P., & McArthur, R. (2002). Web Searching: A Process Oriented Experimental Study of Three Interactive Search Paradigms. *Journal of the American Society of Information Science*. 53(2): 120-133.

19- Ding, W., & Marchionini, G. (1996). A Comparative Study of Web Service Performance. *In S. Hardin (Ed), Proceedings of the 59th Annual Meeting of the American Society for Information Science* (pp.136-142), Medford. NJ: American Society for Information Science.

20- Dong, J. (2000). Combination of Multiple Web Based Search Results and Its Effect on the Search Performance. *Ph.D dissertation*. University of Illinois. Urbana.

21- Dwork, C. et al. (2001) Rank aggregation methods for the Web. Available online (1/15/2002).*http://citeseer.nj.nec.com/dwork01rank.html*

22- Dowk et al. (2002) Rank aggregation revisited. Citeseer. NEC Research. Available online (08/20/2002) *http://citeseer.nj.nec.com/478775.html*

23- Fox, E. & Show, J. (1994). Combination of Multiple Search. *In TREC 2*: pp: 243-249.

24- Gabe, B (2002). Search Engines Revisited. Online July, 3, 2002 *http://www.accurapid.com/journal/14search.htm*

25- Goodrum, A., & Spink, A. (2001). Image Searching on the Excite Web Search Engine. *Information Processing and Management*. 37(2), 295-312

26- Gordon, M & Pathak, P. (1999). Finding Information on the World Wide Web: The Retrieval Effectiveness of Search Engines. *Information Processing and Management*. 35(2): 141-80.

27- Gravano. N. & Garcia- Molina, H. (1995). Generalizing Gloss to vector-Space Database and broker Hierarchies. *Proceeding of the 21st international Conference on very large databases (VLDB).*

28- Griffithe, José-Marie & King, Donald W.(2000) US Information Retrieval System Evolution and Evaluation (1945-1975). *IEEE Annals of the History of Computing.*

29- Gudivada, V.N., Raghavan, V.V., Grosky, W.I., & Kasangottu, R. (1997) *Information Retrieval on the World Wide Web. IEEE Internet Computing*, 1(5), 58-68.

30- Harter, P. Variations in Relevance Assessments and the Measurement of Retrieval Effectiveness. *Journal of the American Society for Information Science*. 47(1): 37-49.

31- Hawking, D., Craswell, N., Thistlewaite, P., & Harman, D. (1999). Results and challenges in the Web Search Evaluation. In Proceedings of the 8th International World Wide Web Conference (WWW8) [On-line] Available at http://www8.org/w8-papers/2c-search-discover/results/results.html

32- Hawking, D., Craswell, N., Bailey, P. & Griffiths, K. (2001). Measuring search engine quality. *Information Retrieval*, 4: pp. 33-59.

33- Hawking, D., Voorhees, E., Craswell, N., and Bailey, P. (2002). Overview of the TREC-8 Web track. In E.M. Voorhees, and D. Harman (Eds.), *Proceedings of the Eighth Text REtrieval Conference (TREC-8).* (NIST Special Publication 500-246). [On-line]. Available at *http://trec.nist.gov/pubs/trec8/papers/web_overview.pdf*

34- Infopeople Search Engines Quick Guide. (2003, January,3) *http://www.infopeople.org/search/guide.html*

35- Internet Domain Survey (2002, June 19) Available Online. *http://www.isc.org/ds/www-200201/index.html*

36- Jansen, J. & Pooch, U. (2001. A Review of Web Searching Studies and a Framework for Future Research. *Journal of the American Society for Information Science*. 52(3): 235-246.

37- Jansen, J., Spink, A., Bateman, J., & Saracevic, T. (1998). Real Life Information Retrieval: A Study of the User Queries on the Web. *SIGIR Forum*, 32(1): 5-17.

38- Jansen, J., Spink, A., Saracevic.(2000). Real Life, Real Users, and Real Needs: A Study and Analysis of Users Queries on the Web. *Information Processing and Management.* 39(2): 207-227.

39- Jansen, B., Spink, A., Pfaff, A. (2000). Linguistic Aspects of Web Queries. *Proceeding of the 36rd. ASIS Annual Meeting* , Volume, 37: 169- 176.

40- Jarvenlin, K. & Kehalainen, J. (2000). IR Evaluation Methods for Highly Relevant Documents. *Proceedings of 23rd Annual International ACM/SIGR Conference on Research and Development in Information Retrieval*. Belkin, N.J.: pp. 41-48

41- Khan, K.. & Locatis, C. (1998). Searching through Cyperspace: The Effects of Link Dsipaly and Link Density on Information Retrieval from Hypertext on the World Wide Web. *Journal of the American Society for Information Science. 49(2): pp. 176-182.*

42- Katzer, J., McGill, M., Tessier, J., Frakes, W., & Dasgupta, P. (1982). A Study of Overlap among Document Representations. *Information Technology Research and Development.* 1(2): pp. 261-274.

43- Kobayashi, M., & Takeda, K, (2000). Information retrieval on the Web. *ACM Computing Surveys*, 32(2), 144-173.

44- Lancaster, F.W. (1998) Indexing and Abstracting in Theory and Practice. Champaign, Illinois : University of Illinois, Graduate School of Library and Information Science, 412 p.

45- Lawrence, S., & Giles, L. (1998). Inquirus, the NECI meta search engine. The Seventh International World Wide Web Conference, Brisbane, Australia, Elsevier Science, pp.95-105. *Accessed from the Web at February, 26, 2003.* *http://www.neci.nec.com/homepages/lawrence/papers/search-www7/*

46- Lawrence, S. & Giles, C.L. (1999). Accessibility of Information on the Web. *Nature,* v0l. 400, 107-109. Available online(June, 19, 2002) at http://wwwmetrics.com/

47- Lawrence, S. & Giles, C.L. (2002). New Study on the Accessibility and Distribution of Information on the Web. Available Online June, 19, 2002. *http://www.neci.nec.com/~lawrence/searchtips.html*

48- Lee, J. (1995) Combining Multiple Evidence from Different Properties of Weighting Schemes. Annual ACM Conference on Research and Development in Information Retrieval. *Proceeding of the 18th Annual International Conference on Research and Development in Information Retrieval.* pp. 180-188.

49- Lee, J. (1997). Analyses of Multiple evidence Combination. Annual ACM Conference on Research and Development in Information Retrieval. *Proceeding of the 23rd Annual International Conference on Research and Development in Information Retrieval.* pp. 267-276.

50- Leighton, H.V., Srivastava, J. (1999). First 20 Precision among World Wide Web Search Services)Search Engines). *Journal of the American Society for Library and Information Science*, 50, 870-881.

51- Lenth, Russ. (2002). Java Applets for Power and Sample Size. *http://www.stat.uiowa.edu/~rlenth/Power/index.html* Visited in November, 25, 2002.

52- Losee, R, & Paris, L. (1999) Measuring Search Engine Quality and Query Difficulty: Rankin with Target and Freestyle. *Journal of the American Society for Information Science*. 50 (10): 882-889.

53- Maze, S., Moxeley, D., & Smith, D.J. (1997) Authoritative Guide to Web Search Engines. New York, NY: Neal-Schuman.

54- Meng, W., Wu, Z., Yu, C. & Li, Z. (2001). A Highly Scalable and Effective Method for Metasearch. *ACM Transaction on Information Systems. 19(3): pp. 310-335.*

55- Meng, W., Yu, C., & Liu, K. (2002). Building Efficient and Effective Metasearch Engines. *In ACM Computing Survey, 34(1): pp. 48-89.*

56- Oppenheim, C., Morris, A., & McKnight, C. (2000). The evaluation of WWW search engines. *Journal of Documentation* 56, 190-211.

57- Rusmussen, E. (2003). Indexing and Retrieval for the Web. *Annual Review of Information Science and Technology*. Vol. 37. pp. 91-123.

58- Saracevic, T. & Kantor, P. (1988). A Study of Information Seeking and Retrieving. III. Searchers, Searches, Overlap." *Journal of the American Society for Information Science.* 39(3): pp. 197-216.

59- Savoy, J., Le-Clave, A. & Vrajitoru, D. (1996). Report on the TREC-5 Experiment: Data Fusion and Collection Fusion. *Proceedings TREC5. NIST Publication* 500-238, Gaithersburg (MD): pp. 489-502.

60- Savoy, J., & Picard, J. (2001). Retrieval effectiveness on the Web. *Information Processing & Management*, 37, 543-569.

61- Schatz, B. (1997). Information Retrieval in Digital Libraries: Bringing search to the Net. *Science*, 275 (2928), 327-334.

62- Schwartz, M. F. & Pu, C.(1998) Applying an Information Gathering architecture to Netfind: A White Pages tool for changing and Growing Internet. *IEEE/ACM Transaction on Networking*, 2(5), 426-439.

63- Schwartz, C. (1998)/ Web Search Engines. *Journal of the American Society for Library and Information Science*, 49, 973-982.

64- Search Engines.com. (August, 15, 2002). http://uk.searchengine.com/

65- Selberg, E. & Etzioni, O. (1995). Multi- Service Search and Comparison using the MetaCrawler. *In the Proceedings of the 4th International World Wide Web Conference, December.*

66- Selberg, E. & Etzioni, O. (1997). The MetaCrawler Architecture for Resource Aggregation on the Web. *IEEE Expert. 12(1): pp. 8-14.*

67- Shaw, J. & Fox, E. (1995). Combination of Multiple Search. TREC-3. Proceedings of the Third Text Retrival Conference. (January, 30, 2003) www.trec.com

68- Si, L. & Callan, J. (2002) Using Sampled Data and Regression to Merge Search Engine Results. *Annual ACM Conference on Research and Development in Information Retrieval. Proceeding of the Twenty Fifth Annual International Conference on Research and Development in Information Retrieval*: pp. 19-26.

69- Smeaton. A. F. & Crimmins. F. (1997). Using a Data Fusion Agent for Searching the WWW. http://decweb.ethz.ch/WWW6/Posters/752/FUSION-W.HTM

70- Spink, A. (2002). A user-centered approach to evaluating human interaction with Web search engines: an exploratory study. *Information Processing and Management.* 38(3): 401-426.

71- Spink, A., Bateman, J., & Jansen.B. J. (1998). Searching Heterogeneous Collection on the Web: Behavior of Excite Users. Information Research: An Electronic Journal , 5(2): http://www.shef.ac.uk/~is/publications/infers.

72- Spink, A., Bateman, J., & Jansen, B. J. (1999). Searching the Web: Survey of Excite users. *Internet Research: Electronic Networking Applications and Policy*, 9(2): 117-128.

73- Spink, A., Wolfram, D., Jansen, B., & Saracevic, T. (2001). Searching the Web: the public and their queries. *Journal of the American Society for Information Science*. 52(3): 226-234.

74- Spink, A., Jansen, B., & Ozmultu, C. (2000). Use of Query Reformation and Relevance Feedback by Excite Users. *Internet Research: Electronic Networking Application and Policy*. 10(4): 317-328.

75- Spink, A., & Ozmutlu, H. C. (2001) What do people ask for on the Web and how do they ask it: Ask Jeeves query analysis. *Information Today*, Inc. 545-554.

76- Sormunen, E. (2002). Liberal Relevance Criteria of TREC Counting on Negligible Documents? *Proceedings of 25th Annual International ACM/SIGR Conference on Research and Development in Information Retrieval*. Tampere, Finland: pp. 324-330.

77- Su, L. (1997). Developing a comprehensive and systemic model of user evaluation of Web-based search engines. *Proceedings of the 18th National Online Meeting* (pp. 335-344). Medford, NJ: Information Today.

78- Sullivan, D (2002). Search Engines Listings. Available Online, June, 25, 200 http://www.searchenginewatch.com/links/

79- Sullivan, D. (2001a, December 11). Search engine sizes. [On-Line] Available http://www.searchenginewatch.com/reports/sizes.html

80- Sullivan, D (2002). The Search Engines Report. Available Online, June, 24, 2002. http://www.searchenginewatch.com/reports/sizes.html

81- Tomiauolo, N.G., & Packer, J.G. (1996) An Analysis of Internet Search Engines: Assessment of Over 200 Search Queries. *Computer in Libraries*, 16(6), 58-62.

82- Tsikrika, T. & Lalmas, M. (2001). Merging Techniques for Performing Data Fusion on the Web. *Proceedings of 23rd Annual International ACM/SIGR Conference on Research and Development in Information Retrieval*. Atlanta, Georgia: pp. 127-134.

83- Turtle, H. & Croft, W. (1991). Evaluation of an Inference Network Based Retrieval Model. *ACM Transitions on Information Systems*. 9(3): pp. 187-222

84- Tzitzikas, Y. (2001). Democratic data fusion for information retrieval mediators. Citeseer. NEC Research. http://citeseer.nj.nec.com/tzitzikas01democratic.html

85- Voorhees, E. (2002). Evaluation of Highly Relevant Documents. Proceedings of 24th Annual International ACM/SIGR Conference on Research and Development in Information Retrieval. New Orleans.: pp. 74-82

86- Voorhees, E.; Gupta, N. & Laird, B. (1994). The Collection Fusion Problem. *NIST Special Publication 500-225: Overview of the Third Text REtrieval Conference (TREC-3)*.

87- Voorhees, E., Gupta, N., Larid, B. (1995). Learning Collection Fusion Strategies. *In the Proceedings of the ACM SIGIR Conference (Seattle, WA): pp. 172-179.*

88- Voorhees, E., Gupta, N. & Johnson, B.(1998). Collection Fusion Problem from Multiple Collection Information Retrieval Systems. *Journal of the American Society for Information Science*. 49(13): pp.1177-1184.

89- Wang, P., Hawk, W., & Tenopir, C. (2000). Users' Interaction with World Wide Web Resources: an Exploratory Study Using A Holistic Approach. *Information Processing and Management. 36. pp.229-252.*

90- Williams, M (1979) Automatic merging of monographic data bases-identification of duplicate records in multiple files: the IUCS scheme.*Journal-of-Library-Automation*. 12 (2: pp. 156-168.

91- Williams, M.; Maclaury, K; Preece, S; & Rouse, S.(1979) Data base mapping model and search scheme to facilitate resource sharing. *volume 1. mapping of chemical data bases and mapping of data base elements using a relational data base structure*. Coordinated Science Laboratory, University Of Illinois At Urbana-champaign. 342 P (Abstract).

92- Williams, M; Preece, S. (1977)Data Base Selector for network use: a feasibility study. Information management in the 1980s: *proceedings of the 40th ASIS Annual Meeting, volume 14*, edited by B.M. Fry. White Plains, New York, American Society for Information Science, Chicago, Illinois, September 26 October 1, 1977(Abstract).

93- Wishard, L (1998). Precision among Internet Search Engines: an Earth Science Case Study. *Issues in Science and Technology Librarianship*.

94- Yang, X. & Zhang, M. (2000) Necessary Constraints for Fusion Algorithms in Meta Search Engine Systems. *In Proceedings of International Conference on Intelligent Technologies, Bangkok. (Accessed through Citeseer Search Engines): pp. 409-416.*

95- Yu, C., Meng, W., Liu, K., Wu, W., & Rishe, N. (1999). Efficient and Effective Metasearch for a Large Number of Text Databases. *In Proceedings of the ACM SIGMOD Conference (Santa Barbara, CA., May): pp. 187-198.*

96- Yuwono, B., & Lee, D.L. (1996). Search and ranking algorithms for locating resources on the World Wide Web. In*: Proceedings of the 12th International Conference on Data Engineering* (pp. 164-171). [On-line]. Available at http://www.cs.ust.hk/~dlee/.

97- Yuwono, B. & Lee, D. (1997). Server Ranking for Distributed Text Retrieval Systems on Internet. *In Proceedings. Of the International Conference on Database systems for advanced Applications*: pp. 14-49, 1997.

Appendices

Appendix 1: Source Queries.

Two Term Queries

1 Teacher Socialization
2 self-preception profile
3 Cataloging Cost
4 Database Overlap
5 Multilingual OPACs
6 Liver Transplantation
7 Programming Algorithm
8 Reading Process
9 Experimental Methodology
10 Road-Map Plan
11 adolescent alcoholism
12 teacher attrition"
13 Journalism Online
14 MARC AND MODS
15 Obesity Treatment
16 health education
17 Libraries Management
18 stroke incidence
integrating technology AND teacher
19 education
20 Event retrieval

Three Term Queries

21 Web Based OPAC's
22 American Physical Activity
23 comparative education methodology
24 Arabic Information Retrieval
25 Non Roman Scripts
26 Liver Transplant Donars
27 Java Applet Programming
28 Individaul Word Recognition
29 Faculty Job Satisfaction
30 Search Engine Sizes
31 perceived social support
32 Social Capital Theory
33 arab identity representation
34 Indexing AND Digital Libraries
35 body massage technique
36 Public Library Mentor
37 Thioredoxin pKa computational
38 geographical stroke incidence
39 culturally responsive teaching
40 Multimedia Digital libraries retrieval

Appendix 2: Total Number of Documents Retrieved Per Query.

Two Terms	Google	AltaVista	Fast
1	669	351	357
2	947	486	459
3	121	78	75
4	150	46	72
5	8	9	10
6	81500	27026	83029
7	28400	9638	15139
8	47700	24440	59225
9	13200	5770	5453
10	11100	1350	1407
11	1720	181	155
12	2980	1705	1975
13	8410	3645	14510
14	9300	513	7491
15	21100	8763	35392
16	1540	1119	2098
17	1920	746	183
18	4380	1898	1780
19	62000	6053	4616
20	506	247	152
Average	14882.55	4703.2	11678.9
Three Terms	**1310**	**855**	**904**
21	33	12	10
22	31	15	21
23	185	116	63
24	2850	1402	1299
25	67	35	42
26	943	514	1354
27	47	24	18
28	432	230	231
29	71	37	63
30	3310	1566	1761
31	2400	1047	937
32	32600	14042	60493
33	143000	8746	10755
34	63	39	27
35	89	23	75
36	57	20	24
37	7160	3346	49
38	2790	1344	1304
39	710	314	345
40	**9907.4**	**1686.35**	**3988.75**
Whole Set Average	**12394.98**	**3194.775**	**7833.825**

Appendix 3: 2-way ANOVA test results for the Number of Doc. Retrieved Per Query

Means and Standard Deviations of the number of Document Retrieved Per Query by the Search.

	Google		AltaVista		Fast		
Number of terms	M	SD	M	SD	M	SD	Marginal
Two	14882.55	23029.6	4703.2	7776.7	11678.9	22345.2	10421.5
Three	9907.4	32147.6	1686.35	3527.9	3988.75	13508.4	15582.5
Marginal	**12394.98**		**3194.775**		**7833.825**		

Results of Analysis of Variance on Number of Doc. Retrieved.

Source	SS	df	MS	F	p
Degree of relevance	1692914057	2	846457028.4	3.842	.030
Error (relevance)	8372881783	38	220338994.3		
Number of terms	819766095.4	1	819766095.4	1.054	.317
Error (n. of terms)	14777269574	19	777751030.2		
Relevance X n terms	110152989.3	2	55076494.6	.251	.780
Error (rel. X n. terms)	8347649641	38	219674990.5		

Appendix 4: Number of Document Overlapped in Two and three Engine.

Query #	# of overlap	Overlap_in_2 engines	Overlap_in_3 engines	Google & AltaVista	Google & Fast	AltaVista & Fast
1	7	7	0	6	0	1
2	3	3	0	1	1	1
3	5	3	2	1	0	2
4	2	2	0	1	0	1
5	10	6	4	3	2	1
6	8	2	6	1	1	0
7	5	5	0	2	0	3
8	3	2	1	1	0	1
9	4	4	0	2	0	2
10	2	2	0	0	1	1
11	3	3	0	0	1	2
12	7	6	1	4	0	2
13	7	4	3	0	3	1
14	2	2	0	1	0	1
15	7	5	2	1	0	4
16	10	10	0	5	4	1
17	3	3	0	3	0	0
18	5	3	2	1	0	2
19	2	2	0	1	1	0
20	5	4	1	0	1	3
21	2	2	0	0	0	2
22	4	3	1	1	0	2
23	8	4	4	2	1	1
24	7	6	1	5	0	2
25	4	1	3	0	1	0
26	5	4	1	2	1	1
27	6	4	2	1	0	3
28	7	3	4	1	0	2
29	1	1	0	0	0	1
30	7	7	0	2	2	3
31	5	4	1	3	0	1
32	3	3	0	1	1	1
33	3	3	0	2	0	1
34	3	2	1	0	1	1
35	6	3	3	1	1	1
36	5	4	1	2	2	0
37	3	3	0	2	0	1
38	3	3	0	2	0	1
39	6	5	1	2	2	1
40	0	0	0	0	0	0
Sum = 820	188	143	45	63	27	
19.50%	12%	7.50%	44.10%	18.20%	37.70%	
Whole_set	5.85	3.6	1.125	0.65	0.675	1.35
Two_term_length	5	3.9	1.1	1.7	0.75	1.45
Three_term_length	4.4	3.25	1.15	1.45	0.6	1.25

Appendix 5: Individual Search Engines Performance.

Two Terms	Google	AltaVista	FAST
1	0.95	1	0.6
2	0.95	0.75	0.8
3	0.75	0.45	0.3
4	0.65	0.6	0.5
5	0.45	0.5	0.4
6	0.8	0.85	0.75
7	0.8	0.75	0.65
8	0.5	0.35	0.25
9	0.55	0.6	0.4
10	0.9	0.6	0.6
11	0.4	0.5	0.5
12	0.7	0.7	0.6
13	0.45	0.5	0.3
14	0.8	0.9	0.4
15	0.9	0.75	0.8
16	0.8	0.5	0.65
17	0.55	0.55	0.25
18	0.85	0.7	0.65
19	0.75	0.8	0.7
20	0.75	0.5	0.45
Mean	**0.7125**	**0.6425**	**0.5275**
Three Terms			
21	0.3	0.2	0.2
22	0.35	0.3	0.4
23	0.6	0.75	0.7
24	0.9	0.9	0.75
25	0.9	0.65	0.5
26	0.6	0.35	0.3
27	0.65	0.55	0.65
28	0.4	0.35	0.3
29	0.7	0.5	0.25
30	0.7	0.4	0.45
31	0.55	0.45	0.35
32	0.75	0.75	0.5
33	0.55	0.75	0.75
34	0.6	0.6	0.3
35	0.7	0.8	0.9
36	0.5	0.3	0.4
37	0.55	0.45	0.35
38	0.65	0.35	0.15
39	1	0.85	0.8
40	0.85	0.55	0.45
Mean	**0.64**	**0.54**	**0.4725**
Grand Mean	0.67625	0.59125	0.5

Appendix 6: 2-way ANOVA test results for Search Engines Precision Values.

Means and Standard Deviations of the Precision Values of the SE..

	Google		AltaVista		Fast		
Number of terms	M	SD	M	SD	M	SD	Marginal
Two	.71	.17	.64	.17	.53	.178	.63
Three	.64	.18	.54	.21	.47	.216	.55
Marginal	**.675**		**.59**		**.5**		

Results of Analysis of Variance on Precision Ratio

Source	SS	df	MS	F	p
Degree of relevance	.622	2	.311	23.264	.000
Error (relevance)	.508	38	1.3		
Number of terms	.176	1	.176	1.912	.183
Error (n. of terms)	1.75	19	9.2		
Relevance X n terms	1.15	2	5.77	.581	.564
Error (rel. X n. terms)	.378	38	9.9		

Appendix 7: Precision at 11 Point Recall Values for 2QL and 3QL.

Recall	Google_2	AltaVista_2	Fast_2	Google_3	AltaVista_3	Fast_3	Two_Term	Three_Terms
0	0.809524	0.73	0.715789	0.885714	0.725	0.738095	0.751771	0.782937
0.1	0.7627	0.735	0.655263	0.764286	0.675	0.654762	0.717654	0.698016
0.2	0.771419	0.69833	0.656137	0.761905	0.66666	0.634914	0.708629	0.687826
0.3	0.788095	0.6825	0.613158	0.741667	0.64375	0.577381	0.694584	0.654266
0.4	0.747619	0.67	0.626316	0.728571	0.635	0.552381	0.681312	0.638651
0.5	0.742062	0.67167	0.605268	0.711114	0.62084	0.547619	0.673	0.626524
0.6	0.732643	0.660005	0.578953	0.714981	0.610705	0.510186	0.6572	0.611957
0.7	0.72381	0.64	0.564474	0.685119	0.578125	0.511905	0.642761	0.591716
0.8	0.704767	0.63945	0.552047	0.687043	0.577775	0.502643	0.632088	0.589154
0.9	0.702381	0.6425	0.526316	0.683333	0.565	0.488095	0.623732	0.57881
1	0.707143	0.6425	0.526316	0.683333	0.565	0.488095	0.62532	0.57881

Appendix 8: FTP for Google – AltaVista Combination.

Query Length	Query Number	Interleave	Rank Sim.	Global Sim
Two Terms	1	1	1	1
	2	0.95	0.85	0.85
	3	0.6	0.6	0.6
	4	0.75	0.7	0.75
	5	0.45	0.5	0.45
	6	0.7	0.8	0.75
	7	0.85	0.85	0.9
	8	0.3	0.4	0.5
	9	0.5	0.55	0.6
	10	0.75	0.9	1
	11	0.4	0.4	0.35
	12	0.85	0.7	0.7
	13	0.4	0.45	0.5
	14	0.65	0.8	0.9
	15	0.7	0.85	0.75
	16	0.8	0.75	0.85
	17	0.65	0.8	0.75
	18	0.9	0.95	0.95
	19	0.85	0.78	0.8
	20	0.7	0.8	0.65
	Mean	0.6875	0.7215	0.73
Three Terms	21	0.3	0.3	0.3
	22	0.45	0.5	0.5
	23	0.75	0.7	0.75
	24	0.92	0.9	0.9
	25	0.7	0.9	0.85
	26	0.6	0.6	0.6
	27	0.65	0.6	0.55
	28	0.35	0.4	0.4
	29	0.35	0.7	0.6
	30	0.5	0.55	0.55
	31	0.5	0.65	0.65
	32	0.65	0.7	0.8
	33	0.75	0.65	0.7
	34	0.45	0.6	0.7
	35	0.8	0.75	0.75
	36	0.65	0.6	0.65

37	0.5	0.65	0.7
38	0.6	0.6	0.5
39	0.9	1	0.95
40	0.8	0.7	0.65
41	.	.	.
Mean	0.6085	0.6525	0.6525
Grand Mean	0.6085	0.6525	0.6525

Appendix 9: 2-way ANOVA test results for Search Engines Precision Values.

Means and Standard Deviations of the Precision Values of the SE.

	Interleave		Rank Sim.		Global Sim.		
Number of terms	M	SD	M	SD	M	SD	Marginal
Two	.6875	.196	.7215	.179	.73	.178	.713
Three	.6085	.18	.6525	.162	.6525	.216	.6378
Marginal	**.648**		**.687**		**.691**		

Results of Analysis of Variance on Precision Ratio.

Source	SS	df	MS	F	p
Degree of relevance	4.5	2	2.27	4.46	.018
Error (relevance)	.194	38	5.09		
Number of terms	.170	1	.170	1.69	.209
Error (n. of terms)	1.9	19	.100		
Relevance X n terms	5.187	2	2.9	.078	.925
Error (rel. X n. terms)	.142	38	3.724		

Appendix 10: FTP for Google – AltaVista Combination.

Query Length	Query Number	Interleave	Rank Sim	Global Sim
Two Terms	1	1	1	1
	2	0.95	0.95	0.95
	3	0.6	0.75	0.75
	4	0.75	0.75	0.75
	5	0.45	0.45	0.45
	6	0.7	0.7	0.8
	7	0.85	0.9	0.85
	8	0.3	0.3	0.4
	9	0.5	0.5	0.45
	10	0.75	0.9	0.1
	11	0.4	0.4	0.45
	12	0.85	0.8	0.8
	13	0.4	0.3	0.45
	14	0.65	0.65	0.7
	15	0.7	0.8	0.75
	16	0.8	0.7	0.7
	17	0.65	0.55	0.7
	18	0.9	0.95	0.9
	19	0.85	0.75	0.8
	20	0.7	0.7	0.75
	Mean	0.6875	0.7215	0.73
Three Terms	21	0.3	0.3	0.3
	22	0.45	0.35	0.4
	23	0.75	0.75	0.6
	24	0.95	0.9	0.95
	25	0.7	0.8	0.8
	26	0.6	0.65	0.6
	27	0.65	0.65	0.55
	28	0.35	0.4	0.35
	29	0.35	0.55	0.45
	30	0.5	0.5	0.7
	31	0.5	0.6	0.55
	32	0.65	0.55	0.65
	33	0.75	0.75	0.75
	34	0.45	0.6	0.4
	35	0.8	0.75	0.75
	36	0.65	0.65	0.65
	37	0.5	0.6	0.55
	38	0.6	0.65	0.6
	39	0.9	1	0.1
	40	0.8	0.85	0.8

Mean	0.6085	0.6525	0.6525
Grand Mean	0.648	0.687	0.69125

Appendix 11: 2-way ANOVA test results for FTP of Google – Fast Combination

Means and Standard Deviations of the combination

	Interleave		Rank Sim.		Global Sim.		
Number of terms	M	SD	M	SD	M	SD	Marginal
Two	.6875	.195	.6900	.21	.6750	.22	**.6842**
Three	.6100	.18	.6425	.177	.5750	.20	**.6092**
Marginal	**.649**		**.666**		**.625**		

Results of Analysis of Variance on FTP of the Combination

Source	SS	df	MS	F	p
Degree of relevance	3.4	2	1.7	1.74	.189
Error (relevance)	.374	38	9.8		
Number of terms	.169	1	.169	1.71	.207
Error (n. of terms)	1.87	19	9.8		
Relevance X n terms	1.38	2	6.9	.416	.663
Error (rel. X n. terms)	.634	38	1.67		

Appendix 12: FTP AltaVista – Fast Combination

Query Length	Query Number	Interleave	Rank Sim	Global Sim
TwoTerms	1	1	0.9	1
	2	0.75	0.75	0.75
	3	0.45	0.45	0.5
	4	0.8	0.7	0.7
	5	0.4	0.4	0.5
	6	0.85	0.75	0.9
	7	0.8	0.9	0.85
	8	0.3	0.25	0.3
	9	0.5	0.5	0.5
	10	0.7	0.75	0.5
	11	0.6	0.6	0.65
	12	0.6	0.7	0.75
	13	0.55	0.3	0.45
	14	0.7	0.4	0.7
	15	0.75	0.8	0.7
	16	0.6	0.6	0.9
	17	0.65	0.7	0.75
	18	0.75	0.75	0.75
	19	0.85	0.85	0.85
	20	0.55	0.55	0.55
	Mean	**0.6575**	**0.63**	**0.6775**
Three Terms	21	0.3	0.2	0.3
	22	0.45	0.4	0.35
	23	0.75	0.8	0.75
	24	1	1	0.9
	25	0.7	0.65	0.7
	26	0.45	0.45	0.45
	27	0.6	0.7	0.6
	28	0.3	0.3	0.3
	29	0.25	0.25	0.35
	30	0.3	0.4	0.4
	31	0.55	0.5	0.55
	32	0.75	0.75	0.7
	33	0.75	0.8	0.75
	34	0.5	0.5	0.45
	35	0.8	0.6	0.85
	36	0.6	0.45	0.6
	37	0.45	0.45	0.5
	38	0.35	0.35	0.35
	39	0.85	0.85	0.85
	40	0.55	0.55	0.6
	Mean	**0.5625**	**0.5475**	**0.565**
	Grand Mran	**0.61**	**0.58875**	**0.62125**

Appendix 13: 2-way ANOVA test results for FTP of AltaVista – Fast Combination

Means and Standard Deviations of the combination

	Interleave		Rank Sim.		Global Sim.		
Number of terms	M	SD	M	SD	M	SD	Marginal
Two	.6575	.171	.63	.193	.6775	.181	**.655**
Three	.5625	.21	.5475	.216	.5650	.194	**.5583**
Marginal	**.61**		**.589**		**.621**		

Results of Analysis of Variance on FTP of the Combination

Source	SS	df	MS	F	p
Degree of relevance	2.179	2	1.09	2.659	.083
Error (relevance)	.156	38	4.09		
Number of terms	.280	1	.280	2.679	.118
Error (n. of terms)	1.9	19	.105		
Relevance X n terms	4.5	2	2.7	.509	.605
Error (rel. X n. terms)	.170	38	4.46		

Appendix 14: Three Way Combinations

Query Length	Query Number	Interleave	Rank Sim.	Global Sim
TwoTerms	1	0.93	0.97	0.97
	2	0.83	0.87	0.83
	3	0.53	0.57	0.57
	4	0.7	0.73	0.7
	5	0.4	0.6	0.37
	6	0.77	0.73	0.77
	7	0.8	0.87	0.83
	8	0.3	0.33	0.37
	9	0.53	0.53	0.5
	10	0.73	0.83	0.8
	11	0.5	0.4	0.53
	12	0.7	0.63	0.73
	13	0.47	0.33	0.47
	14	0.73	0.67	0.8
	15	0.8	0.87	0.8
	16	0.67	0.67	0.7
	17	0.7	0.7	0.73
	18	0.83	0.83	0.8
	19	0.83	0.73	0.8
	20	0.67	0.63	0.63
	Mean	0.671	0.6745	0.685
Three Terms	21	0.3	0.3	0.27
	22	0.43	0.37	0.43
	23	0.8	0.73	0.73
	24	0.87	0.9	0.9
	25	0.73	0.8	0.77
	26	0.53	0.53	0.5
	27	0.6	0.73	0.57
	28	0.3	0.33	0.27
	29	0.37	0.5	0.5
	30	0.5	0.43	0.83
	31	0.53	0.57	0.53
	32	0.7	0.67	0.7
	33	0.7	0.67	0.67
	34	0.53	0.5	0.5
	35	0.77	0.73	0.77
	36	0.57	0.53	0.6
	37	0.53	0.5	0.6
	38	0.5	0.57	0.53
	39	0.9	1	0.97
	40	0.7	0.7	0.7
	Mean	0.593	0.603	0.617
	Grand Mean	0.671	0.6745	0.685

Appendix 15: 2-way ANOVA test results for the three way combinations

Means and Standard Deviations of the number FTP for each Function.

	Interleave		Rank Sim.		Global Sim.		
Number of terms	M	SD	M	SD	M	SD	Marginal
Two	.6710	.16	.6745	.179	.6850	.165	.677
Three	.5930	.17	.6030	.184	.6170	.186	.604
Marginal	**.632**		**.639**		**.651**		

Results of Analysis of Variance on FTP for the combination functions

Source	SS	df	MS	F	p
Degree of relevance	7.422	2	3.7	1.2	.309
Error (relevance)	.116	38	3.06		
Number of terms	.158	1	.158	1.78	.197
Error (n. of terms)	1.67	19	8.8		
Relevance X n terms	5.15	2	2.57	.083	.921
Error (rel. X n. terms)	.118	38	3.11		

Appendix 16: 2-way ANOVA test results for the Merging Two Engines Results

Means and Standard Deviations for the Three Merging Functions and the Different QL.

	Interleave		Rank Sim.		Global Sim.		
Number of terms	M	SD	M	SD	M	SD	Marginal
Two	2.54	. 0.26	2.6	0.23	2.6	0.22	**2.593**
Three	2.6	0.20	2.6	0.21	2.62	0.205	**2.625**
Marginal	**2.58**		**2.612**		**2.634**		

Results of Analysis of Variance for the three merging functions

Source	SS	df	MS	F	p
Degree of relevance	2.94	2	1.47	2.491	.111
Error (relevance)	.106	18	5.9		
Number of terms	1.51	1	1.5	5.784	.040
Error (n. of terms)	2.36	9	2.6		
Relevance X n terms	2.3	2	1.15	1.369	.280
Error (rel. X n. terms)	.151	18	8.39		

Appendix 17: 2-way ANOVA test results for Merging Three Engines Results

Means and Standard Deviations for the Three Merging Functions and the Different QL.

	Interleave		Rank Sim.		Global Sim.		
Number of terms	M	SD	M	SD	M	SD	Marginal
Two	2.5667	.34261	2.5500	.35956	2.5933	.36784	**2.570**
Three	2.2700	.36194	2.2967	.37487	2.3233	.36492	**2.297**
Marginal	**2.418**		**2.423**		**2.458**		

Results of Analysis of Variance for the three merging functions

Source	SS	df	MS	F	p
Degree of relevance	2.85	2	1.42	.249	.781
Error (relevance)	1.6	28	5.7		
Number of terms	1.68	1	1.681	42.544	.000
Error (n. of terms)	.553	14	3.95		
Relevance X n terms	7.167	2	3.58	.065	.937
Error (rel. X n. terms)	1.53	28	5.47		

Appendix 18: The code of the Perl scripts:

The code used for eliminating the overlapped documents and creating the index html file

```
# this script eliminate the overlapped documents and create an
# index html file for the documents retrieved from the three search engines
# the items are randomized to eliminate the biasness
# open spreadsheet for reading
open (DOC1, "doc1.txt") or die "Can't open file: $!";

# create a file called index.html for writing to
open (INDEX, ">index.html") or die "Can't open file: $!";

# loop over each line from spreadsheet
while ($line = <DOC1>) {

   # read each field into a variable
   ($no, $rand_no, $title, $url) = split("\t", $line);

   # skip this line if the url has already been seen
   next if $urls{$url};

   # now save the url so we know we've seen it
   $urls{$url} = 1;

   # create a HTML link using the url and the title variables
   $link = qq(<a href="$url">$title</a>);

   # create a hash where the random number is the key for the value
   $links{$rand_no} = $link;

}

# Put more HTML formatting here if you want to.
# put them in this form: print INDEX qq(<tags><tags>);
print INDEX qq(<html><body>);

# sort the links in the hash by their keys, which is the random number
foreach $result (sort numerically (keys (%links))) {
   print INDEX $links{$result};
   print INDEX qq(<br /> \n);
}
# print the end of the html document
print INDEX qq(</body></html>);
```

```
# subroutine that does numeric sort
sub numerically { $a <=> $b; }
```
This script is used for detecting the overlapping documents positions and the number of overlapped documents

```
# open spreadsheet for reading
open (DOC1, "doc1.txt") or die "Can't open file: $!";

# create a file called report.txt for writing to
open (REPORT, ">report.txt") or die "Can't open file: $!";

# create a file called index.html for writing to
open (INDEX, ">index.html") or die "Can't open file: $!";


# loop over each line from spreadsheet
while ($line = <DOC1>) {

   # read each field into a variable
   ($no, $rand_no, $title, $url) = split("\t", $line);


   # skip this line if the url has already been seen
  if ($urls{$url}){


         $matches{$url} .= ", " . $no;
         next;
         }


   # now save the url so we know we've seen it
   $urls{$url} = 1;

   $matches{$url}= $no;

}

print REPORT "The following numbers matched: \n\n";

foreach $match (sort keys %matches){

print REPORT $matches{$match}, "\n" if ($matches{$match} =~ /,/);

}
```

```
# subroutine that does numeric sort
sub numerically { $a <=> $b; }
# open spreadsheet for reading
open (DOC1, "doc1.txt") or die "Can't open file: $!";

# create a file called index.html for writing to
open (INDEX, ">index.html") or die "Can't open file: $!";

# loop over each line from spreadsheet
while ($line = <DOC1>) {

   # read each field into a variable
   ($no, $rand_no, $title, $url) = split("\t", $line);

   # skip this line if the url has already been seen
    next if $urls{$url};

   # now save the url so we know we've seen it
   $urls{$url} = 1;

   # create a HTML link using the url and the title variables
   $link = qq(<a href="$url">$no. $title </a>);

   # create a hash where the  number is the key for the value
   $links{$rand_no} = $link;

}


# Put more HTML formatting here if you want to.
# put them in this form: print INDEX qq(<tags><tags>);
print INDEX qq(<html><body>);

# sort the links in the hash by their keys, which is the random number
foreach $result (sort numerically (keys (%links))) {
   print INDEX $links{$result};
   print INDEX qq(<br /> \n);
}

# print the end of the html document
print INDEX qq(</body></html>);

# subroutine that does numeric sort
sub numerically { $a <=> $b; }
```

Appendix 20: The C code for calculating the global similarity and the rank similarity.

```
/* This program calculate the global similarity score for
* each document retreived from the three search engines
* Khaled Mohamed
* Created at May, 28, 2003
*/

#include <stdio.h>
void main()
{
#define MAX 10;

        /* float DBR1 = .25;  set rank for database one */
        /*float DBR2 = .50;   set rank for database two */
        /*float DBR3 = .75;   set rank for database three */

        double g_1, g_2, g_3;

        int rank[10] = {1,2,3,4,5,6,7,8,9,10};
        double f1, f2, f3;

        f1 = .033;
        f2 = 0.05;
        f3 = 0.1;



        for (int i = 0; i < 10; i++)
        {
                g_1 = 1 - (rank[i] * f1);
                g_2 = 1 - (rank[i] * f2);
                g_3 = 1 - (rank[i]  * f3);
                printf("the global similarity for %d = \t %8.5f \t %8.5f \t %8.5f\n\n",
                i+1, g_1, g_2, g_3);

        }
}
```

```
/* This program will calculate the rank similarity for each
*       rank document from 1 to 10
*       Khaled Mohamed
*       Fist issue May, 28, 2003
*/

#include <stdio.h>
void main()
{
        double ranksim;
        int rank[10] = {1,2,3,4,5,6,7,8,9,10};
        int numofDocRetrieved;
        double x;
        double y;
        printf("Enter the number of document of retrieved\t");
        scanf("%d", &numofDocRetrieved);

        for (int i = 0; i < 10; i++)
                {
                        x = rank[i];
                        y = x / numofDocRetrieved;
                   ranksim = 1 - y;


                        printf("\ndoc_num %d \t %7.5f\n", i+1, ranksim);
                }
                printf("Enter the number of document of retrieved\t");
                scanf("%d", &numofDocRetrieved);

        for (int i = 0; i < 10; i++)
                {
                        x = rank[i];
                        y = x / numofDocRetrieved;
                   ranksim = 1 - y;


                        printf("\ndoc_num %d \t %7.5f\n", i+1, ranksim);
                }
                printf("Enter the number of document of retrieved\t");
                scanf("%d", &numofDocRetrieved);

        for (int i = 0; i < 10; i++)
                {
```

```
                x = rank[i];
                y = x / numofDocRetrieved;
            ranksim = 1 - y;


                printf("\ndoc_num %d \t %7.5f\n", i+1, ranksim);
        }


}
```