Cross-media Meta-search Engine

CHENG Tung Yin

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Systems Engineering and Engineering Management

June 2005

摘要

在這個資訊發達的年代，網上有很多強大的搜尋器去幫助用戶找尋及瀏覽他們針對某事物想要知道的資料。可是，有些是使用單獨一個搜尋器 (單一搜尋器) 去作網上資料搜尋，其所得到的結果會遠比綜合幾個單一搜尋器的覆蓋範圍少及不全面。更有研究 [10] 指出大部份的單一搜尋器只能回傳少於 45%與用戶所查詢的有關連的資料結果，這對於用戶來說實在是極不方便，就算對於一個極具經驗的用戶亦然。

研究 [10] 指出了使用綜合搜尋器的好處，它能綜合幾個單一搜尋器的結果作爲它的合併結果，以及回傳更多有質素及與查詢有關連的結果。該搜尋器包括：先選擇以哪個單一搜尋器去作爲它的資料結果來源，分析結果的內容，合併資料及根據系統對結果與查詢事項的關聯程度以遞減形式重新排列結果，刪除重覆的結果，最後回傳結果給用戶。所以，目前有不少研究都致力改良這些程序。[1], [2], [3] 探討了綜合搜尋器如何選擇單一搜尋器；[4] 討論了結果內容的分析； [5], [6], [7], [8], [9] 提出一些合併及重新排序結果的方法。就這樣，現時發展了不少的綜合搜尋器，也有多項文獻討論它們的架構，有些工作更研究如何改良綜合搜尋器的搜尋所需時間，資料的儲存容量，及其查詢處理。

本研究旨在改良綜合搜尋器在重新排序及合併結果方面的表現，我們只集中發掘及嘗試不同的排列方法來改良綜合搜尋器的表現，我們提出利用搜尋結果之間的關係去達到這個目的。我們並進行實驗，針對各搜尋器回傳與查詢有關係的詳細資料的能力，而對該搜尋器的表現作出評價及比較。

本研究的首部份先會探討現有的綜合搜尋器，針對它們對搜尋結果的重新排序及合併的演算法加以分析。

隨後我們提出其它的排序演算法。我們所提出的綜合搜尋器將會用來搜尋多媒體的結果，包括純文字，圖像，聲音檔及視像檔。從那些參與該綜合搜尋的單一搜尋器之結果，經過

分析後我們發現到它們有些特點是可以加以利用，例如它們互相之間的關係 (當中有不同種類的關係，而不同的關係有不同的關聯程度)，我們發現擁有愈多與其它搜尋結果有關的網頁，就會和用戶所查詢的愈有關，而與其它搜尋結果沒有關係的網頁就多數是與查詢不相關的。 考慮了這個因素，我們提出這特點可用來改良綜合搜尋器的表現。而我們將會把它融入現有的重新排序演算法，再發展成新的演算法。

我們的實驗會針對：使用了現有排序演算法的綜合搜尋器，及使用我們所提出的排序演算法的綜合搜尋器，作出評價。

最後我們會針對各項綜合搜尋器的實驗結果加以分析，再與所參與的單一搜尋器一併比較，繼而找出一個最佳的排序演算法。

關鍵字: 單一搜尋器, 綜合搜尋器, 資料搜尋, 與查詢有關的結果, 結果內容分析, 合併資料, 重新排序演算法, 多媒體

Abstract

There are many powerful search engines on the Web in this era that assist data and information searching and mining. However, using individual search engines suffer from a web resource coverage problem. Empirical results show that many single search engines cannot return more than 45% of the relevant results [10]. Hence, even the most experienced users encounter challenges in working efficiently with the entire collection of search engines.

As a result of this limitation, much work had been performed to develop meta-search engines to deal with these problems. Meta-searching involves source selection [1, 2, 3], text and snippet information analysis [4], data fusion from different search engine results (re-ranking the results) [5, 6, 7, 8, 9], duplicate detection and removal, and finally presentation. Many meta-search engines have been explored using a variety of approaches, and many studies have discussed the framework of these meta-search engines [4, 10, 11]. Some work [12] has also contributed to enhancing the meta-search performance in other areas, such as runtime, storage, query processing, etc.

Our paper aims at improving the retrieval performance of a multimedia meta-search engine. It contributes to the literature in two areas. First, it develops a meta-search engine that retrieves multimedia objects, which is important, as there are not many meta-search engines that provide such a function. Second, it develops a new merging algorithm that improves the retrieval performance of a meta-search engine by ranking the results in descending order of desirability and relevancy with respect to a given query. We propose the utilization of different types of relationships of different strengths between the items in the search results, as we observe that most items in the same set of search results that have a strong or multiple relationships with other items in the results are more relevant to the query, whereas those that have a few and also weak, or even no relationships with the other retrieved items are usually irrelevant or cannot be accessed. We then incorporate this relationship feature into the current merging methods to investigate whether there is an improvement in the retrieval performance of the meta-search engine.

Our meta-search engine searches for items of various media in web, images, audio and video items. In these items, we discover that there are features that can be utilized to improve retrieval performance. We incorporate these features into existing well-suited re-ranking algorithms to develop new multimedia meta-searching algorithms, and carry out an experimental study to implement both the current and proposed re-ranking policies.

Finally, we analyze the experimental results to compare the retrieval performance of meta-search engines that employ different re-ranking policies, and also that of the involved single search engines, to see whether our proposed algorithm brings any improvement.

Keywords: single search engines, meta-search engines, information mining, relevant results, text and snippet information analysis, data fusion, re-ranking algorithm, multimedia

Acknowledgements

I would like to thank Professor Christopher C. Yang sincerely for his advice and comments on my research, and for encouraging me to publish the results. His work on information retrieval has inspired my interest in the topic of this research.

The advice and comments of Professor W. Lam and Professor Frank Y. H. Chen are highly appreciated, and their suggestions have greatly improved the quality of this research.

In addition, I would also like to extend my genuine gratitude to my family, who has given me great support and a warm nurturing environment. I also thank my friends, who have always been there to share happiness and sadness with me, and have been a constant source of support in my daily life. Some have even acted as teachers who have guided me toward a positive life and instilled in me great knowledge.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

1.1.1 Information Retrieval

Information retrieval is a subtopic of computer science that is concerned with presenting the information that is gathered from information resources to information users. It mainly consists of determining which documents in a collection contain the keywords in the user query in order to satisfy the *user information need* [56]. An information retrieval system must know how to interpret the contents of the information documents in a collection and *rank* them according to their *degree of relevance* to the *user's query*. The interpretation of a document's content involves the extraction of semantic and syntactic information from the text and matching this information to the user information need. The difficulty lies not only in extracting this information, but also in making the decision as to relevance. Thus, the notion of *relevance* is at the center of information retrieval, and in fact the primary goal of an information retrieval system is to retrieve all of the documents that are relevant to a user's query while retrieving as few non-relevant documents as possible.

Our work aims at devising algorithms that allow the integration of the results in multiple search engines that retrieve multimedia objects from the Web to improve the retrieval performance.

1.1.2 Search Engines

The World Wide Web contains a vast amount of useful up to date and archival information. In the very past, the display of query results from search engines was limited to a text format, for example MetaGer and Inquire, which was inconvenient for users who wanted to search for information in other types of media formats. Multimedia search engines such as Google, AltaVista and Lycos were later developed so that query results could be presented in image, audio, and video formats.

A search engine runs by taking a user's query, which is a statement of their information needs, and retrieving the results from its database. The database stores the internal representation of the documents (actually there is a *Web Crawler* for updating the database by keeping download pages from the Internet, processing them into its internal representation, and then storing them with indexing into the database), ranks them, and returns the result set to the user in the interface that it provides. Sometimes a score, title, or summary is returned along with the document.

In our research, we focus on the development of a better ranking algorithm that merges search results from different search engines that are involved in the meta-searching process, which we call a *re-ranking or merging algorithm* to distinguish it from the usual ranking algorithms that are used in search engines. *Data merging* techniques are therefore applied in this aspect.

1.1.3 Data Merging

Data merging is based on the concept of combining many answers to a query into a single answer. The benefits of using a meta-search engine rather than a single search engine are discussed later. Data merging techniques are useful in combining the result sets from different and unrelated search engines that is employed in meta-search engines. We would introduce the frameworks and techniques that are involved in meta-search engines in the following section.

1.2 Meta-search Engines

1.2.1 Framework and Techniques Employed

Framework

The underlying techniques that are used in meta-search engines draw ideas from a number of different areas of classical information retrieval studies, including query processing, source selection, re-ranking, and presentation. Once a query is submitted to a meta-search engine, it is processed and translated as appropriate, and the meta-search engine interface module connects to the selected subset of search engines by opening multiple connections via multi-threading. The processed query is then passed by the search engine on to the underlying search engines, from which results are obtained in the html format. The results are parsed, advertisements are removed, and the actual links are extracted and returned. If the number of links to be retrieved is larger than a given search engine's link increment value, then multiple html pages are retrieved until either the search engine's results are depleted or the requested number of links has been fetched.

The re-ranking module then merges the results from the participating search engines by using the properties of the documents, such as their scores, their ranking as returned by the underlying search engines, or their entire contents. The results from the multiple search engines are then ordered and merged, duplicates are removed, and the results are finally presented to the user.

Query Processing

Meta-searching begins with the submission of the user's query to the system user interface. Two users may submit queries for the same information using different terms or combinations. For example, user 1 types "Apple AND Computer", whereas user 2 may type "Computer + Apple". The attributes of the query parameters for search engines have been analyzed, and it has been found that by assembling the query with appropriate attribute values, queries can be sent in a uniform fashion to search engines. There is also other work contributed to query processing besides translation of Boolean Queries problem. Hector et al. [47] discussed techniques for rewriting predicates in Boolean queries into native subsuming forms, which is a basis for the translation of complex queries to enable query languages to be more uniform. Chen et al. [44] presented a quality-controlled query processing method for the Web by using some defined distance functions that could be used to evaluate the quality of the query parameters. Chidlovskii and Borghoff [45] studied the problem of the semantic caching of Web queries, and developed a caching mechanism for conjunctive Web queries that is based on signature files. Strzalkowski, Wang, and Bowden [46] investigated the role of automated document summarization in building effective search queries.

Source selection

Source selection, or collection selection, focuses on the identification of the right collections to be queried given a particular user query, which means selecting the search engines that are to be used as the sources for the meta-searching. Powell et al. [42] suggested that improvements in database selection could lead to broader improvements in retrieval performance. Daniel Dreilinger and Adele E. Howe [1, 15] evaluated and studied the efficacy of the incrementally acquired metaindex approach for the selection of search engines in SavvySearch, which is a meta-search engine that is designed to intelligently select and provide interfaces with multiple remote search engines, by analyzing the effect of time and experience on performance. Meng and Wu [3] proposed a highly scalable and accurate database selection method that operates by collecting and using metadata that reflects the contents of each search engine. Hawking et al. [38] suggested that meta-search engines could download a set of documents from each search engine to gather and learn statistics about each source. Garbe [39] devised a meta-search engine BINGOO that selects a subset of the search engines to make a query on the client side.

Re-ranking

The key component of a meta-search engine is the method that is used to merge and sort the individual lists of documents that are returned by different engines and present them to produce a ranked list to the user. Yager and Rybalov [31], Callan et al. [40], and Yu et al. [41] studied the merging of results from multiple search engines. Kumar et al. [8] developed a Mearf meta-search engine that employs four novel re-ranking methods. Our research focus is re-ranking, which is investigated more deeply in the next few chapters, and re-ranking algorithms are proposed for the improvement of system performance.

Presentation

The final stage of meta-search is presentation. This involves the displaying of the ranked list that results from a query to users, that can take different visualizing formats or be clustered or grouped using different aspects, such as topic. Zamir [48] and Mann [50] worked on the visualization of search results, and Zamir and Etzioni [49] suggested the idea of document clustering for visualization. Roy et al. [54] and Chen and Dumais [57] contributed to the organization of search results into a hierarchy of topics and sub-topics that facilitates the browsing of a collection and the location of results of interest. Cugini et al. [51] even proposed the visualization of search results in a 3-D design, and in another study [52] evaluated and compared the visualization interface of query results in text, 2-D, and 3-D designs.

The framework below depicts the procedures that are involved in the framework of a meta-search engine.

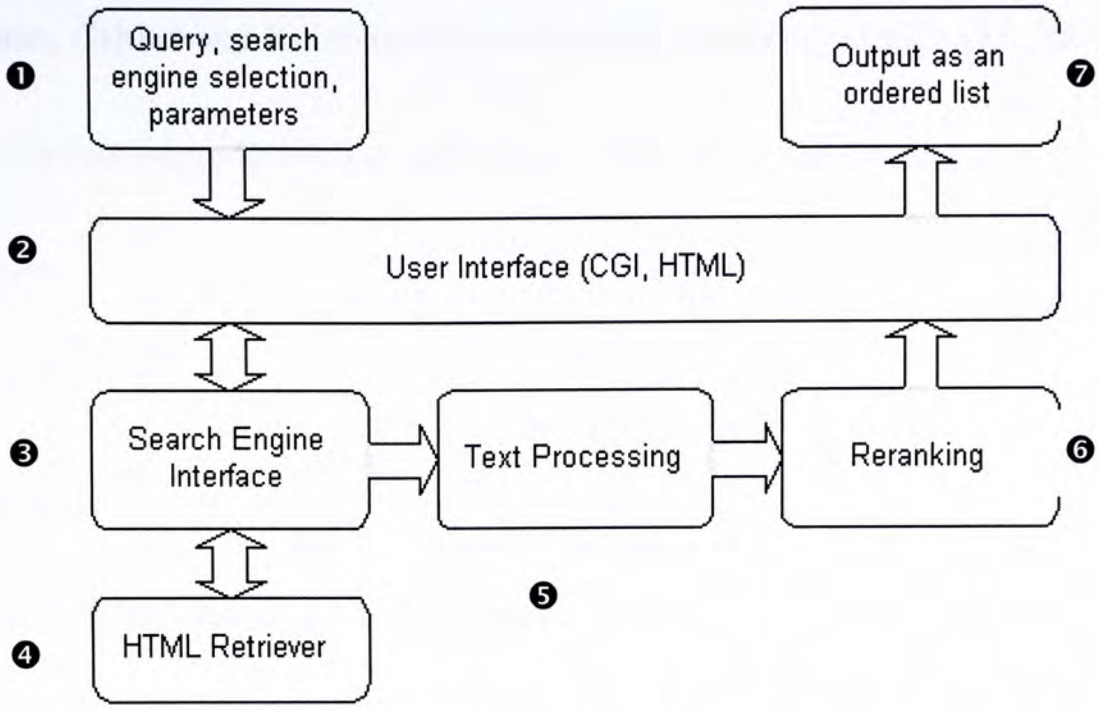

❶
Query, search engine selection, parameters
Output as an ordered list
❼
❷
User Interface (CGI, HTML)
❸
Search Engine Interface
Text Processing
Reranking
❻
❺
❹
HTML Retriever

Figure 1.1 Framework of a Meta-search Engine

❶ User's input (query by keywords or phrases, for example, "Hong Kong", "apple")

❷ Meta-search engine's user interface, which is run using techniques such as Java, ASP, HTML, and CGI.

❸ User's input is passed to the databases of search engines such as Altavista, AlltheWeb, Lycos, Google, Excite, Ditto, and AskJeeves through the interface of the engine.

❹ The search engine's HTML retriever works to retrieve the HTML pages of the returned documents for later text processing.

❺ HTML code extraction for the title, URL, property, summary, and even snippets (stop list, stemming, tf-idf normalization) for certain fusion methods, using the corpus statistics of the document such as Centroid and BestSim.

❻ The fusion methods that are used by the meta-search engine, such as Interleaving [8].

❼ The merged results returned to the user (client side), presented in an ordered list.

Currently there are many meta-search engines, such as John Wiley [72]'s MetaSpider, Steve and

Lee [73]'s Inquirus, Calmet and Kullmann [74]'s KOMET, Hawking et al.'s [75] PADRE, Gauch and Wang [6, 33]'s Profusion, Dreilinger and Howe [15]'s SavvySearch, and Selberg and Etzioni [11]'s MetaCrawler.

Merging Techniques

Having reviewed the framework of a meta-search engine, we briefly introduce the merging techniques that are involved, which is the area to which our work is restricted. As search engines return different sets of results and *result identifiers*, which may be a score, rank, title, or snippets, a meta-search engine can take all of these as inputs to its merging algorithm. Several studies [16, 17, 30, 31, 32] have discussed various data merging techniques that are performed by meta-search engines [1, 6, 10, 11, 33]. Savoy et al. [34] showed an example of data merging that made use of the scores of documents as computed by the participating search engines. Yager and Rybalov [31] and Voorhees et al. [32] proposed other merging algorithms by considering the original ranking positions of the documents in the search engines. We discuss this issue in more detail in Chapter 2.

1.2.2 Advantages of meta-searching

With the explosive growth and widespread accessibility of the Web nowadays, most single search engines are unable to index a large enough proportion of the available Web pages [23]. Furthermore, it is more and more difficult to keep up with the rate at which resources that have already been indexed are updated, which results in decreased coverage of Internet information.

Lawrence and Giles [24] found that the fraction of Internet information that is covered by the databases of search engines is shrinking.

The heuristics that are used in different search engines are often different, their qualities may not be the same across query types. This means that different search engine sources give different relevancy for queries, and some even return information that is irrelevant, outdated, or unavailable. Searching using just one engine thus gives a worse performance. Even the same search engine will often respond to the same query differently over time, as the Web Crawler may have captured different documents from the database that correspond to the same query, because the database changes. Even the performance of a fixed database varies, performing well for some queries and poorly for others.

In section 1.2.1, we can see that meta-search engines have the potential to address the problems that are inherent in single search engines by combining search results from multiple sources. They can provide better overall coverage of the Web than any individual search engine, and as they provide averaging procedures, the idiosyncrasies of any search engine can be smoothed out during the merging process, which creates a more reliable and consistent system. They can also offer potentially better overall rankings by taking advantage of the different heuristics that are used in different search engines. For example, the second link that is retrieved by a search engine may be more relevant to the query than the first link that is retrieved by another search engine. Selberg and

Etzioni [10] also highlighted the benefits of meta-search engines compared to individual search engines.

1.3 Contribution of the Thesis

Ranking is an integral component of any information retrieval system. In the case of a Web search, the role of ranking is critical because of the size of the Web and the special nature of Web users. It is common for a Web search query to have thousands or millions of results, but Web users do not have enough time and patience to go through them all to find out what they are interested in. It has actually been stated that most Web users do not look beyond the first page of results [25, 26, 27]. Therefore, it is important for the ranking function to give the desired results within the top few pages, otherwise the search engine is rendered useless. The focus of this thesis is therefore to develop a novel merging algorithm that improves the merging algorithm over the existing methods.Different merging methods that are used in existing meta-search engines are investigated, compared, and the most appropriate for modification purposes to develop the new merging algorithm are found, so that the retrieval effectiveness of the system can be increased after the proposed algorithm is applied.

There are many existing algorithms that utlize different properties of documents returned along with the search engines to merge the search results from the search engines participating in the meta-search. They use the score of the document, the rank of the document, the corpus statistics of the documents, or the entire content of them. Nevertheless, except the rank of the document, others are not always returned by the participating search engines. The algorithms proposed in our thesis are therefore motivated by the basic idea of one of the merging methods suggested in the Mearf system [8], which .merges the retrieved items based on their ranks and sameness. However, we found that the current meta-search engines such as the Mearf system still have limitations. They do not consider other types of relationships between multimedia items during the merging process except "duplicate" relationships, that is, the existing merging algorithms only utilize retrieved items that are duplicated to foster the relevance or importance of an item to the user query. This disregards the effect of other types of relationships on this support. This is because if other objects in other media formats appear in the search results, in which the media objects are related to an object whose relevancy to the query is to be determined, then the relevance of those related objects to the query and the similarity between them, together with the ranking of the object in question from its orginating search engine should help to support the relevance of the object under consideration and boost the rank of that object. Because of the limitations of the current mergning algorithm, our system is proposed with the aim of ranking objects that are either relevant to the query or can render more information about the query at higher ranks by considering the relationships other than "duplicate" between retrieved items. This is further discussed in Chapter 3, in which the different kinds of relationships are introduced.

By analyzing the search results returned by the participating search engines, we observe that there are relationships among the various items, and discover that an item's relationship with other items in the search results is related to its relevance. Items with more related objects in the search results under the same query are more relevant to that query than items with no or fewer related items. In addition, stronger relationships between the items (meaning that they are closer or more similar) supports the relevance of the item to the given query. This characteristic can be utilized to re-rank the results during the meta-searching process. By boosting the rank of an item that has more and closer related items in the search results at the expense of items that have fewer or no related items, the retrieval performance is improved, because the former are in most cases more relevant to the given query.

In this research, two multimedia meta-searching algorithms are built to implement our idea, which are then tested to see whether any improvement in the ranking performance of the system can be discerned.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 gives a review of the related work on meta-search engines and the different techniques that are employed in each process in such systems. Chapter 3 illustrates the derivation and design of the proposed algorithms used in our meta-search engine. Chapter 4 describes an experiment that implements several multimedia meta-search engines, details the evaluation methodology, and discusses the results using some widely used performance measures. Finally, we draw conclusions from our findings in Chapter 5.

Chapter 2

Literature Review

Nowadays, many meta-search engines operate using different information retrieval methods for source selection, merging results, and presentation. They include the Mearf system [8], Metacrawler, Profusion, Savvy Search [1, 15], Callan, and Inquirus. These engines use different re-ranking methods (in this thesis, the words fusion, re-ranking, merge are used interchangeably), which can be grouped into several types according to the document properties that are returned by the underlying search engines. These document properties are the document's *score, ranking position, titles and snippets* (stop list, stemming, tf-idf normalization), and *entire contents*, which are explained in the following.

2.1 Preliminaries

Before looking at those merging methods, we first introduce the symbols and terms that are used in the methods and their definitions.

Definition of symbols and terms

Let l_i^s be the hyperlink of a document that is ranked at the ith position in search engine s. Let *score*(l_i^s) be the system score that is assigned to the link l_i^s, which is the relevance measure of the corresponding document, where a higher value denotes greater relevance (according to the system) of the document to the query. Let *rank* (l_i^s, *s*) be the ranking position of the ith document in search engine s. Let w_s be the weighting of a search engine *s* among all of the participating search engines. This is further discussed later when the Linear Combination model is introduced. Let *raw score(l_is)* be the original score for a document l_is.

2.2 Fusion Methods

2.2.1 Fusion methods based on a document's score

Some search engines return a score along with each document as an input for a meta-search engine.

A document's score can be calculated using different statistics:

- Term frequency (tf)

  The number of occurrences of a term t_i in a query q or a document.

- Document frequency (df)

  The number of different documents containing the term across the selected databases, as sometimes there maybe overlapping of documents across the databases (participating search engines involved in the meta-searching process).

- tf weight of a term t_i in a query

  Weight factor based on the tf information in a query:

  $$\frac{\text{number of occurences of the term ti in a query}}{\text{total number of terms in a query}}$$

- tf weight of a term t_i in a document

  Weight factor based on the tf information in a document:

  $$\frac{\text{number of occurences of the term ti in a document}}{\text{total number of words in a document}}$$

- idf (inverted document frequency) weight of a term t_i

Weight factor based on the df information:

$$\log(N / DF(t_i))$$

where $DF(t_i)$ is the number of different documents that contain the term t_i across the selected databases, and N is the number of documents in the selected databases.

A document's score is the similarity between a query and a document, which can be measured by the dot-product of their respective vectors. A query vector is composed of the product of the tf weight of term t_i in that query and the idf weight of term t_i. A document vector is composed of the tf weight of t_i in that document. The dot-product is usually divided by the product of the lengths of the two vectors to normalize the similarity between 0 and 1.

A document's score (global similarity between query q and document d)

Let $\mathbf{q} = (q_1,\ldots, q_n)$ be a query, where q_i is the tf weight of term t_i in q (there are a total of n terms in q).

Let idf_i be the idf weight of t_i. The query vector is then $\mathbf{q'} = (q_1 * idf_1, \ldots, q_n * idf_n)$.

Let $\mathbf{d} = (d_1, \ldots, d_n)$ be a document vector, where d_i is the tf weight of t_i in d. $|d| = (d_1^2 + \cdots + d_n^2)^{1/2}$, which is the length of vector d. By the cosine similarity function, the global similarity between query q and document d is

$$sim(q, d) = (q' \cdot d) / |q'| \cdot |d| = (q_1 * idf_1 * \frac{d_1}{|d|} + \cdots + q_n * idf_n * \frac{d_n}{|d|}) / |q'|$$

The global similarity between query q and document d is used by many search engines to determine the document score.

Limitation

Sometimes a document's score would not be returned by the search engine, which results in its non-availability. The score can be calculated, but this is time-consuming and involves a large computational cost, which may affect the performance of the meta-search engine, as all the scores for all of the documents in all of the underlying search engines would need to be calculated. The document scores can be recomputed at the client side, but this would entail high communication costs.

Despite this limitation, there are several models that use document scores. Shaw and Fox [7] and Harman [64] tried the fusion method based on the un-weighted min, max, median, or sum of the normalized scores of documents across the participating search engines. They also tried using weight *n(d)*, which is the number of systems that return a given document d, by using the following formula.

$$s(l_i^s)' = (n\,(l_i^s))^{\gamma} \sum_i s_i(l_i^s) \qquad \gamma \in \{-1, 0, 1\},$$

where the score of document l_i^s , $s(l_i^s)$, which is not returned by a search engine i, $s(l_i^s)$ is assigned 0, and the sum is over the participating search engines.

When $\gamma = -1$, the result is equivalent to the average score over search engines that returned l_i^s, which is known as "CombANZ" (Average-of-Non-Zeros). When $\gamma = 0$, the result is simply the sum of the scores over all of the search engines, which is known as "CombSum", and when $\gamma = 1$, the result gives heavier weighting to a document that is returned by more search engines, which is known as "CombMNZ" (Multiply-by-Non-Zeroes).

"Comb" algorithms are the standard algorithms that are used for this application. Shaw and Fox [7] found that of these algorithms, "CombSum" in which the scores of documents are simply summed, provides the best retrieval performance. Ng et al. [65] and Voorhees and Harman [66] found that there is no improvement when two different routing task algorithms are merged by averaging their normalized relevance scores ("CombANZ"). We show some of the fusion methods emerged from "CombSum" in the following.

Fusion methods from "CombSum"

The fusion methods that make use of the scores of documents use the following general algorithm.

Algorithm

```
let results be an empty array of links
for each link $l_i^s$
    ** calculate the score, score($l_i^s$) for each document $l_i^s$
while there are duplicate links across search engines
```

```
        merge the links by adding up their scores
        add all links to results
        sort links in results according to their scores
        return results
```

However, there are different methods to calculate the score for a document (** in the algorithm).

(i) Confidence score fusion [9]

This method first distributes the confidence scores of the documents as returned by each engine within a range of 0 to 1000. The top ranked document from each engine has a confidence score of 1000. Duplicates are eliminated, and the scores of the removed references are added to the sum of the confidence scores of the duplicated references. The overall confidence score for each document is then used to determine how closely the document matches a query in the re-ranking process. Meta-search engines such as MetaCrawler [10] use this method if the document scores are available from the participating search engines.

(ii) Raw score fusion [9]

If the relevancy scores of documents are available and the scores from different collections are comparable, then multiple results can be merged based directly on the document scores. Meta-search engines such as Callan et al. [16] use this method if the document scores are available and comparable.

(iii) Normalized score fusion [9]

If the document scores are incomparable, then they can be normalized by standardization, so that the best ranked is given a score of 1 and the worst ranked is given a score of 0. The original relevancy scores *score* (l_i^s) are mapped into the range [0, 1] after normalization. Let s be *score* (l_i^s) and s'' be score of l_i^s after standard normalization.

The standard normalized score of a link l_i^s is

$$s'' = \frac{s - \min(s)}{\max(s) - \min(s)}.$$

The sum of all of the normalized scores of the duplicates is used to rank the retrieved items in descending order of score. SavvySearch [1, 15] uses this method to merge the results from multiple search engines.

(iv) Weighted score fusion/linear combination of scores scheme (LC) [9, 20, 43]

This method ranks documents against the product of the document scores and the weights of the collections to find the real-valued relevance of the documents. This method benefits documents from search engines with high scores, but also allows a good document from a search engine with a low score to be ranked high if the document has a sufficiently high score. If there are duplicated documents, then the maximum of all of the scores of the duplicates will be the final relevancy score of each different document, which is used to decide the final ranking of the document.

Algorithm

If w' = (w_1, w_2, …, w_n) is the weight assigned to each individual search engine s_1, s_2, …, s_n, then the real-valued relevance ρ of a document d to a query q, $\rho\,(w', d, q)$ is given by:

$$\rho\,(w', d, q) = \sum_{search\_engine} w_i * \rho_i(d,q),$$

where $\rho\,(d, q)$ is the relevancy score of d returned by a certain search engine.

Instead of taking the sum of the scores (CombSum), some meta-search engines, such as ProFusion [6, 33], take the maximum of the scores amongst the duplicates. A hybrid of the normalized score and weighted score fusion is used that maps the original relevancy scores into values of [0, 1]. The normalized scores are then multiplied by the estimated accuracy of search engines w_i. As a result, the maximum of all of the weighted normalized scores $w_i*s_i(d)$ of the duplicates of a document d will be the document's final relevancy score $s_{ProFusion}(d)$ that determines its ranking.

$$s_{ProFusion}(d) = max_i\,(w_i*s_i(d))$$

The weighting of a search engine among all participating search engines can be evaluated using the following methods [16, 20].

The following is an example of weighting a search engine s.

$$w_s = 1 + |C| * (s-s_{mean})/s_{mean}$$

$|C|$: number of search engines involved

s: score of search engine s

s_{mean}: the mean of all of the scores of the search engine.

A search engine's score can be found by using the *user relevance judgments* as training data for the system. Relevance judgments are human evaluations of a document's relevance with respect to a certain query. With enough of such information, the overall performance of a search engine can be determined. Each individual search engine is then weighted according to its performance using the aforementioned formula. In other words, the training data is tested on each search engine so that the performance is evaluated using standard information retrieval metrics. However, it is generally expensive to make such relevance judgments, as it involves human analysis, and thus the use of training data to obtain the weighting of a search engine in this way is usually not possible.

Nevertheless, none of these methods is applicable if the document scores are unavailable, that is, when they are not returned by the participating search engine during the meta-search process.

Scores that are calculated using raw score fusion and weighted score fusion are incomparable (because there is no normalization of the original document scores) in the participating search engine, which means that a document with a higher score in one search engine may be less relevant to one that has a lower score in another search engine. This is because some search engines calculate document scores using different corpus statistics (such as idf or average document length), which can lead to much variance.

2.2.2 Fusion methods based on a document's ranking position

Some search engines, in fact many of them, do not return document scores. The following table shows a short summary of this.

Search Engines	Does **not** return document score	Return document score
Yahoo (www.yahoo.com)	√	
Google (www.google.com	√	
MSN (http://search.msn.com)	√	
AltaVista (www.altavista.com)	√	
AlltheWeb (www.alltheweb.com)	√	
Lycos (www.lycos.com)	√	
C4 (www.c4.com)	√	
Inquirus (www.inquirus.com)	√	
YiSou (http://www.yisou.com)	√	
So 163.com (http://page.so.163.com)	√	
Sina (http://search.sina.com)	√	
Baidu (www.baidu.com)	√	
Mamma (www.mamma.com)	√	

Teoma (www.teoma.com)	√	
WiseNut (www.wisenut.com)	√	
Overture (www.overture.com)	√	
Ask Jeeves (www.ask.com)	√	
LookSmart (www.looksmart.com)	√	
AOL (http://search.aol.com)	√	
Netscape (http://search.netscape.com)	√	
DMOZ (http://dmoz.org/)	√	
Infoseek (www.infoseek.com)	√	
WebCrawler (www.webcrawler.com)	√	
Search (www.search.com)	√	
Ultraseek (www.ultraseek.com)		√

When the document scores are unavailable, it is said to be in a "rank-only" situation, in which one can only simulate a document's score using the rank of the document. For example, Interleaving and Agreement fusion methods are used in the Mearf meta-search system that was proposed by Oztekin, Karypis, and Kumar et al. [8]. Voorhees et al. [32] also suggested that results from the participating search engines can be interleaved.

Fusion methods

(i) Interleaving [8, 32]

In this method, the results from different search engines are interleaved, and the result sets of search engines are visited one by one to fetch each rank, that is, the first result is taken from all of the search engines, then the second, then the third, and so on. If the current link from a search engine is a duplicate of a previously visited link (the link has occurred before when visiting the result sets), then this link is skipped and the next search engine is analyzed. This method corresponds to the linear combination of scores scheme [20] with equal search engine weights, which takes the best score if there are duplicates, except that interleaving makes use of the document's ranking, rather than its score.

Algorithm

```
let n be # links to be retrieved from each engine
let results be an empty array of links
for i=1 to n
        for s=1 to # search engines
                if $l_i^s$ exists and is not a duplicate of links in results
                        insert $l_i^s$ at the end of results
return results
```

Meta-search engines such as the Mearf system [8] and Callan et al. [16] use this method if no document scores are available.

Limitation

This method produces the best retrieval performance only if the individual ranking of each search engine is perfect and each search engine is equally suited to the query. The rankings of the search engines are perfect if the rankings of all of the documents are consistent, that is, if A is ranked higher than B in one search engine, then it should also be ranked higher than B in another search engine. However, such ranking does not always occur. It is also unlikely that all of the sources will have equal numbers or proportions of relevant documents, and if each search engine performs differently with respect to a query, then the merged results will also perform differently depending on the order in which the search engines are inserted during the interleaving.

(ii) Agreement [8]

Unlike Interleaving, in which the best rank of a link is always selected even if it occurs in multiple search engines (duplicates), Agreement does not ignore the duplicated links, as it is suggested that a link that appears in multiple search engines is more important than a link that appears in just one engine at a similar rank. For example, a link that is ranked second, third, and second in three different search engines would be a better link (more relevant to the query) than a link that is ranked first in one search engine only. Agreement is introduced to boost the rank of documents that appear in multiple search engines that participate in the meta-search process.

Algorithm

```
let results be an empty array of links
for each link $l_i^s$
    score $(l_i^s) = [\ 1/\text{rank}(l_i^s, s)\ ]^c$
while there are duplicate links across search engines
merge the links by adding up their scores
    add all links to results
sort links in results according to their scores
return results
```

Note that c is a constant that usually takes 1 for convenience. The emphasis on agreement is increased if c is small.

Limitation

Although this method does consider the relevance of duplicate links, it suggests equal weights for the search engines, and as a result the effect of the search engines that participate in the meta-searching process is ignored.

(iii) Democratic data fusion [19]

This method is similar to the Agreement fusion method but differs in that it views the fusion problem as an election, in which the documents are the candidates, the systems represents the electors, and each ordering corresponds to a voting ticket [19]. A document's ranking in an individual search engine is determined by the number of voting tickets, and the higher the ranking, the less the number of voting tickets. The number of voting tickets is summed for each document, and the document with the least number of tickets is the most relevant to the query.

Mathematical model

Let D = {d_1, .., d_n} be a collection of n documents. Let S = {S_1, …, S_k} be a set of k information retrieval systems (search engines) over the collection D. Let S_0 be a mediator (meta-searching engine) over the set of systems S. When a query q is received, the mediator S_0 forwards q to each system in S. When a system S_i receives the query, a linear ordering of the set D is returned. This ordering is denoted by $(D)_i$. After retrieving the orderings from the underlying systems, the system fuses them to derive a single ordering $(D)_0$.

To fuse the result sets, assume a mediator [S_1, …, S_k] that has forwarded a query q to each of the underlying systems, and let {$(D)_1$, .., $(D)_k$} be the returned orderings of D. $r_i(d)$ is used to denote the position, from left to right, of d in $(D)_i$, which is also the ranking position of d in $(D)_i$. For example, if $(D)_i$ = <d_1, d_2>, then $r_i(d_1) = 1$ and $r_i(d_2)$ =

2. Let S be a mediator of D and let d be a document in D. The vote of d over S, $V_S(d)$, is the sum of the ranking position of d in each S_i

$$V_S(d) = \sum_{Si \in S} r_i(d).$$

This is similar to the Agreement fusion method, in that both methods consider the document's ranking and accumulate data on the relevance of duplicates to determine the overall relevance of the document in the merging result. This method ranks the document with the least number of voting tickets as the most relevant, whereas the Agreement method takes the reciprocal of the document's ranking as its score, and thus the higher the document score, the higher the ranking of the document in the merged result.

(iv). Hybrid of the score-based and rank-based methods

Some systems, such as MetaCrawler (http://www.metacrawler.com) that is described by Selberg [69], use the Normalize-Distribute-Sum algorithm to merge the results from the search engines. During phase 1, the document scores are normalized in the standard mode, and in phase 2 the algorithm distributes them using the following distribution formula.

$$s(l_i^s)' = s(l_i^s) * \frac{\max(r_i) - r_i(l_i^s) + 1}{\max(r_i)}$$

Let $s(l_i^s)'$ and $s(l_i^s)$ be the new and old scores, respectively, of document l_i^s, which are normalized in the standard mode (phase 1). $max(r_i)$ is the worst rank of the document in search engine i and $r_i(l_i^s)$ is the rank of document l_i^s in search engine i. The formula uses a combination of the score and rank of the document. Finally, the algorithm assigns a final score to the document by adding up the scores that it obtains from all of the participating search engines.

2.2.3 Fusion methods based on a document's URL title and snippets

Some search engines return the title and snippets of a document, that is, a short text summary. Bo Shu and Subhash Kak [28] proposed an Amvish system with a meta-search combination that makes use of the textual summaries of documents. In the fusion methods of Centroid, WCentroid, BestSim, and BestMSim [8], a sparse vector is formed for each link using the URL's title and snippets (stop list, stemming, tf-idf normalization).

There is a dictionary that consists of about 50,000 stemmed words and an augmented stop list, both of which are geared for html and snippet domains. If a term does not appear in the dictionary but appears in the query, then it is assumed to be a rare term, and is assigned a predetermined important idf value. If a term is neither in the query nor in the dictionary, or if it is in the stop list but not in the query, then it is ignored. Each vector is normalized using 2-norm approach. The Mearf system [8] implemented sparse vector, html and text processing modules to handle all html to text conversions, word stemming (a variation of Porter's stemming algorithm), text to sparse vector conversions, and operations on sparse vectors.

A set of relevant documents, say, the first k links, are found first, and all documents in the search result are then re-ranked based on their cosine similarities to a vector obtained from the relevant set. Therefore, the original rankings of the documents in each search engine can affect the selection of relevant set. However, the set produced for each fusion method is different: Centroid and WCentroid use all of the first k links, whereas BestSim and BestMSim consider the first k links from each search engine, but do not include all of them in the relevant set, instead using a subset of the links that is selected according to their content.

Before discussing the fusion methods, we first introduce some notation.

Recall that l_i^s is the i^th link of search engine s and *score*(l_i^s) is the relevance measure of link l_i^s, where a higher value means greater relevance.

Furthermore,

vector(l_i^s) is the sparse vector of link l_i^s formed by processing the URL title and the snippet of a link (or a document), which is the augmentation of the link's triplet (URL, URL title, snippet). A link thus has a quadruple (URL, URL title, snippet, sparse vector).

Permutation $p(r_1, r_2,..., r_s)$ is an n-tuple of positive integers, and an entry r_i denotes the position of a link in search engine i, where s is the number of search engines that are used in the query. For example, *p(1, 21)* states that the first link from search engine 1 and the twenty-first link from search engine 2 have been selected.

Range selection $rs(set_1, set_2,... ,set_s)$ of size s is applied to permutations of size n, the purpose of which is to put a limit on the permitted permutations of size n for a given context. Each set_i is a set of positive integers and a permutation $p(r_1, r_2,..., r_s)$ that is restricted with a range selection $rs(set_1, set_2,... ,set_s)$ is valid only if $\forall i,(i \in [1,n] \wedge i \in N) \Rightarrow r_i \in set_i$, where N is the set of positive integers.

It can be seen that the number of valid permutations for a given range selection $rs(set_1, set_2,... ,set_s)$ is $| set_1|$ x $| set_2|$ x $| set_3|$ x…x $| set_n|$, where $| set_i|$ is the cardinality of set_i.

$|V|_2$ is the 2nd normalization of vector V = √ |V · **V**| = √ (v_1^2+....+v_n^2)

Fusion methods

(i) Centroid [8]

The first k links from each search engine can be deemed to be relevant to the query. The average (centroid) of the vectors of the first k links returned by each search engine is thus found.

The links are ranked using the cosine measure of their sparse vectors to the centroid vector by taking the maximum of the scores in case of duplicates.

Algorithm

```
let k be the number of top links to be considered in ranking
let centroid be an empty sparse vector
let results be an empty array of links
for s=1 to # search engines
    for i=1 to k
        if $l_i^s$ exists
            centroid = centroid + vector($l_i^s$)
centroid = centroid / $|centroid|_2$
for each link $l_i^s$
    score ($l_i^s$) = vector ($l_i^s$) · centroid
while there are duplicate links across search engines
```

```
        merge duplicates by taking the maximum of the scores
    add all links to results
    sort links in results according to their scores
    return results
```

Meta-search engines such as Mearf apply this method.

This method does not weight the links based on criteria such as the rank of a link in its original search engine.

(ii) WCentroid [8]

This method is like Centroid, but weights each link according to certain criteria, for example, the rank of the link in its original search engine, instead of treating the links equally when finding the centroid of the vectors of the first k links in each search engine that are believed to be relevant to the query. The average (centroid) of the vectors of the first k links returned by each search engine is found by using a linearly decaying weighting function starting with 1 at the first rank, and min_val at the k^{th} rank, where min_val is a value between 0 and 1. If k is small (about 5), then it is better that min_val be between 0.25 and 0.5, but if k is larger, then it should be between 0 and 0.25. The links are ranked using the cosine measure of their sparse vectors to the centroid vector by taking the maximum of the scores in case of duplicates.

Algorithm

```
let k be the number of top links to be considered in ranking
let centroid be an empty sparse vector
let results be an empty array of links
for s=1 to # search engines
    for i=1 to k
        if $l_i^s$ exists
            centroid += vector($l_i^s$ )·[1-[((i-1)·(1-min_val))/k]]
centroid = centroid / $|centroid|_2$
for each link $l_i^s$
    score ($l_i^s$ )= vector($l_i^s$ ) · centroid
while there are duplicate links across search engines
    merge duplicates by taking the maximum of the scores
add all links to results
sort links in results according to their scores
return results
```

Meta-search engines such as Mearf apply this method. As with Centroid, the rankings of the search engines are used to select the relevant set that is used in the re-ranking process.

(iii) BestSim [8]

Unlike the two centroid methods, the first k results from each search engine are considered, but the relevant set does not consist of all of the first k links. Instead, a subset of them is selected using the content of the links. A link from each search engine is found such that the tuple of links selected has a maximum self-similarity over all of the permutations in the range set.

All permutations $p(pos_{s1}, pos_{s2}, ..., pos_{sn})$ that are restricted with a range selection r_s $(\{1, 2, ..., k\}, \{1, 2, ..., k\},..., \{1, 2, ..., k\})$ are considered, and the best permutation is $bp(r_1, r_2, ..., r_s)$, for which the self-similarity of the vectors of the links $l_{r1}^{1}, l_{r2}^{2}, ... l_{rs}^{s}$ is the highest over all possible permutations. $(l_{r1}^{1}, l_{r2}^{2}, ... l_{rs}^{s})$ is the tuple of link that has the maximum self-similarity, where l_{ri}^{i} is the r_i th link in search engine i. The links are ranked using the cosine measure of their sparse vectors to the best similarity vector by taking the maximum of the scores in case of duplicates.

Algorithm

```
let current_best = -1

for each search engine i
```

```
    $set_i = \{1,2,...,min(k, number\_of\_links\_returned(i)\}$
if all $set_i$ are empty
    return nil
for each valid permutation $p(r_1, r_2, ..., r_s)$
under rs $(set_1, set_2, ..., set_s)$
    centroid=\underseti=1\oversetsΣvector($l_{ri}^{i}$)
    if $|centroid|_2$ > current_best
        current_best=$|centroid|_2$
        best_centroid=centroid
best_centroid=best_centroid/$|best\_centroid|_2$
for each link $l_i^s$
    score($l_i^s$)= vector($l_i^s$)·best_centroid
while there are duplicate links across search engines
    merge duplicates by taking the maximum of the scores
add all links to results
sort links in results according to their scores
return results
```

Meta-search engines such as Mearf employ this method.

Unlike the centroid-based schemes, the content of the link is used to select the relevant set that is used in the re-ranking. A single permutation with the best similarity is found by making use of the first k links, and thus has the potential to seize the main theme that is presented in the first k links from each search engine, which is better for specific queries.

(iv) BestMSim [8]

Similar to BestSim, the first k results from each search engine are considered, but the relevant set does not consist of all of the first k links, but a subset of them that is selected using the content of the links. However, in this method the first m best permutations are found, instead of just a single permutation with the best similarity. Initially, the first k links are considered from each search engine, and a permutation is found that has the highest similarity. This permutation is marked, the links that are selected from the range sets are removed, and then the sets are augmented by the next available links (k+1). After repeating this m times, the relevance set is obtained. It should be note that by the removal, a link from each search engine can only appear in one of the permutations. M tuples of link are found that have the maximum self-similarity over all of the permutations in m range sets, that is, m best_centroid in the following algorithm are found and summed m times. Second normalization is then carried out to achieve the overall best similarity (ranking_vector in the algorithm). Finally, the links are ranked using the cosine measure of their sparse vectors to the centroid vector by taking the maximum of the scores in case of duplicates.

Algorithm

```
let current_best = -1

let ranking_vector be an empty sparse vector

for i=1 to s

   $set_i$ = {1,2,...,min(k, number_of_links_returned(i)}

for i=0 to m-1

    for each valid permutation p($r_1$, $r_2$, ..., $r_s$)

    under rs($set_1$, $set_2$, ..., $set_s$)

        centroid=\underseti=1\oversets∑vector($l_{ri}^{i}$)

        if $|centroid|_2$ > current_best

            current_best = $|centroid|_2$

            best_centroid = centoid

            for j = 1 to s

                index[j] = $r_j$

        for j = 1 to s

            $set_j$ = $set_j$-{index[j]}

           $set_j$ = $set_j$ +{(k+i)}

        ranking_vector += best_centroid/$|best\_centroid|_2$
ranking_vector = ranking_vector/$|$ranking_vector$|_2$
for each link $li^s$
    score($l_i^s$) = vector($l_i^s$)$\cdot$ranking_vector
while there are duplicate links across search engines
    merge duplicates by taking the maximum of the scores
add all links to results
sort links in results according to their scores
return results
```

Meta-search engines such as Mearf utilize this method. As with the BestSim scheme, the content of the links is used to select the relevant set to be used in the re-ranking. However, m best permutations with the best similarity are taken by making use of the first k+m links, and thus there is the potential to acquire more than one theme in the first k+m links, which is better for multi-modal or general queries.

2.2.4 Fusion methods based on a document's entire content

Some meta-search engines take the time to look up each document (*full-text analysis*) that is returned by the participating search engines, and use the entire content of documents in their fusion algorithm. This fusion method fetches selected documents and orders them according to the relevancy scores that are calculated during the meta-searching. Unlike the document scores that are mentioned in subsection 2.2.1, which have already been calculated by the search engines and returned with the documents for merging in a meta-search engine, the scores in this approach are obtained by using the cosine similarity between the document and the query. This cosine similarity is derived according to the number of query items presented in documents, the proximity between the query terms, the term frequencies, and other factors.

Vogt's ACE model [13] investigated this approach in an experimental framework, and it is also executed on the Internet by the Inquery system [14] and the NECI meta-search engine [29]. MetaGer [http://meta.rrzn.uni-hannover.de], which is a meta-search engine that was designed especially for Germans [18] also applies full-text analysis to retrieve the entire contents of the documents returned, and merges them using information retrieval techniques that are applicable to full documents only. However, this scheme involves a huge amount of communication between the server and the client at great computational cost.

A number of meta-search engines are also available on the Web, for example, C4 {http://www.c4.com}, Ixquick {http://www.ixquick.com}, and Mamma {http://www.mamma.com}. However, due to their commercial nature, limited information is available on the underlying approaches to combining the results.

2.3 Comparison of the Fusion Methods

Callan et al. [16] conducted tests on the score-based merging methods that are detailed in section 2.2.1 and concluded that the weighted score method is the best. However, as aforesaid, fusion methods based on document scores are not always applicable and appropriate, as document scores and search engine scores may not be available if the sources only provide a ranked list of documents but no numerical scores, which is actually the case for most search engines. Although document scores can be calculated, a high computational cost is incurred.

Therefore, fusion methods that are based on a document's ranking may be more suitable for merging results from most search engines. Interleaving uses the highest rank of a document among its duplicates to help decide its final ranking, which ignores the relevance of the same document in other search engines. However, duplicates are removed and are not involved in the merging process, and thus Interleaving overestimates the relevance of a document which has a high rank in one search engine but no rank in the targeted set of retrieved results in other search engines, and underestimates the relevance of a document that has a relatively lower rank but appears in many search engines. Moreover, Interleaving produces excellent rankings only if the individual ranking of each search engine is perfect and each search engine is equally suited to the query.

Agreement overcomes the problem of Interleaving, but they both have insufficiencies. Both methods use a document's ranking only to decide its relevance to the given query, but it is possible that a document's ranking does not reflect its overall relevance among all of the documents in the search results. For example, a document that is ranked lower by one source may be more relevant to the query than another document that is ranked higher by another source, but the former document would be improperly ranked lower in the final combined result due to its low original rank.

Suppose that there are two search engines, SE 1 and SE 2

SE 1	SE 2
A	C
B	D

Using Interleaving and Agreement, one of the merged results is A->C->B->D. A and C would be ranked higher than B and D whatever, but it is possible that B is more relevant to the user's query than C, so B should be ranked higher than C. This problem is similar to the un-normalized score fusion problem, in which document scores are incomparable.

It is found that fusion methods that are based on a document's corpus statistics require no calculation and can overcome the aforementioned problems. The Centroid, WCentroid, BestSim, and BestMSim methods can raise the ranks of documents that are similar in content to the top ranked documents that are deemed relevant by using expert agreement about the content to merge and re-rank the documents. In this way, the contents of a document can be taken into account to improve the accuracy in deciding a document's relevance to the original user query.

The Centroid re-ranking method is similar to the Weighted centroid (WCentroid) method, but differs in the way in determining the centroid only. The Centroid method does not consider the ranks of the links that are given by the original source search engines, whereas WCentroid weights the links according to their placing in the search engines. Such weighting is calculated by [1- [(i-1) · (1-min_val)/k]], where i is the rank of the link. Thus, the first few links are given higher weights, the function decays the weights of the links according to their place in the top k. The WCentroid method uses a relevance set in which each link is weighted differently based on certain criteria (such as the ranks of the links from the source search engines in Mearf's method), instead of being treated equally.

The BestSim and BestMSim methods use somewhat different approaches from the two centroid-based schemes. The centroid methods utilize the rankings of the search engines in selecting the relevant set to be used in the re-ranking, whereas BestSim and BestMSim consider the first k links from each participating search engine, but the relevant set does not include all of the first k links, but rather a subset of them that is selected using the contents of the links. In the BestSim method, a link is found from each source so that the tuple of links selected has the maximum self-similarity. The best permutation bp(r_1, r_2, …, r_s) is searched for so that the self-similarity of the vectors of the links l_{r1}^{1}, l_{r2}^{2}, l_{r3}^{3}, …, l_{rs}^{s} is highest of all the possible permutations. The BestMSim method is like the BestSim method, but the first m best permutations (if k+m does not reach the total number of links exceeded) are found (m best_centroid are found and summed m times and second normalization is carried out to achieve the overall best similarity), rather than a single permutation with the best self-similarity. The BestSim method can be viewed as a method that seizes the main theme that exists in the first k results from each search engine, and is thus be more appropriate for specific queries. BestMSim is likely to seize more than one theme in the first k+m links, and so is preferable for cases of multi-modal or general queries.

However, merge methods that are based on the corpus statistics of documents are unsuitable for meta-searching for other media formats, such as image and video, as there are parts of the corpus statistics that are not returned with these items by the participating search engines.

It is in fact difficult to say which re-ranking method is always the best, as each performs differently depending on the parameters. We can only ascertain which fusion method generally yields a good result in that the documents are ranked reasonably according to their relevance to the query.

To effectively evaluate the new proposed method, we need to compare the modified and unmodified re-ranking methods, and the ranking methods of the participating search engines.

User experimental evaluation should also be carried out to evaluate and compare the performance of these re-ranking methods using metrics. The general idea is that the greater the number of links that is presented to users that are relevant to the query, the better the re-ranking method.

2.4 Relevance Feedback

Relevance feedback is a process by which user feedback preference on the contents of each article is obtained. This information can be obtained from users explicitly using human judgment, or implicitly by applying implicit measures.

Explicit rating is defined as the consciously expressed preference of a user on a discrete numerical scale. Users explicitly assign a rating on a numeric scale based on how relevant they think the articles are to their queries and how much information they obtained from the results. For instance, the ProFusion meta-search engine [6, 33] asked users to evaluate the performance of each search engine under comparison based on their explicit relevance judgments.

Implicit rating is the interpretation of user behavior or selection to impute a vote of preference. Dumais et al. [62], Kelly and Teevan [63] presented several implicit measures. Corin and Eric [61] recorded the time and date of each page that was visited by users, its URL, and the topic of the page contents. Semantic ratings from users have also been used by some systems [58, 59]. The Siteseer system [60] utilized personal bookmark lists. However, conducting relevance feedback based on implicit ratings is very costly, and is beset by problems such as inaccuracy in the evaluation that is caused by feedback from first-time users. These problems are explained more in Chapter 4 in the discussion of our evaluation methodology.

Chapter 3

Research Methodology

3.1 Investigation of the features of the retrieved results from the search engines

There are different types of relationships between Internet items, in textual, image, audio, and video formats. Hence, some items retrieved by the search engines that participate in a meta-search process also have relationships between them. In theory, an item that has more relationships with other items in the results for the same query is more relevant to the query, whereas items that have fewer or no relationships with other items are less relevant, or even irrelevant. Furthermore, if an item has a stronger relationship with other items, then it means that the items are similar to one another, and are likely to be more relevant to the query than an item that has weaker relationships with other items. This is because the relevance of the item that has stronger relationship with other items is further supported by the related items that are also retrieved by the participating search engines for the same query. For example, suppose that we want to obtain information about "David Beckham," and there is a page (W) in the search results for webs that describes his profile that has a link to his photo (P), and such photo is also one of the search results for images. Suppose that we do not know what the photo is before we investigate it, but have grounds to believe that the photo is most probably concerned with "David Beckham", as it is linked to the Web object (W). We can thus assume that the photo is more relevant to the query because of its relationship with the Web object compared to another retrieved image object that has no relationships with any of the search results for any type of media. We should thus boost the rank of the image object (P) that is linked to the Web object, and do nothing for the image object that has no relationship with other items in the search results to reflect the fact that the image object (P) is more relevant to "David Beckham." In addition, we should also boost the rank of the Web object (W) that contains the photo (P) over Web objects that have no relationship with other objects, as (W) contains more details about the query than the latter objects, although they are both relevant to the query.

We find that the existing re-ranking mechanisms that are discussed in the literature review do not include this relationship factor in determining the relevance of an item when merging the results from the participating search engines. This thesis therefore suggests that by utilizing and incorporating the relationships between the items in the search results into the existing merge mechanisms and developing new merge mechanisms by modifying the existing methods, the retrieval performance of meta-search engines can be improved. The main research focus here is therefore to incorporate the relationship features between retrieved items into the existing merge methods. Hence, there is no need to conduct experiments to compare the performance of existing algorithms, as this has already been undertaken in other studies, such as that of the Mearf system [8]. Here, we restrict our work to ascertaining whether incorporating the relationship factor into current merge algorithms can improve a system's retrieval performance.

However, we still need to choose a current re-ranking algorithm to be modified to test our proposed algorithm. At this stage, we are not concerned with which of the existing ranking algorithms is the best. Even if could find the best, we would still need to re-rank the sorted results again by considering the relationships between the items to ascertain whether this further improves a system's performance. In the following, we elaborate our idea. In the introduction to this thesis, we highlighted that in addition to using the existing re-ranking methods, we should also consider the relationships between the objects in different media formats (web, image, audio, and video). There is also a benefit in considering many kinds of relationships that we would illustrate later. Agreement merge method introduced before takes relationships into account but for "duplicate" relationship only, so it just boosts the ranks of items which have duplicates in other underlying search engines.

The idea that a system's retrieval performance can be improved by using the Agreement merge method is further supported by Lee [68] and Ng and Kantor [70]. Lee [68] suggested that an "unequal overlap property" (relevant items have more overlap than irrelevant items) held, that is, different retrieval techniques used in different search engines retrieve many of the same relevant documents but different irrelevant documents. Ng and Kantor [70] therefore concluded from the work of Lee [68] that if this suggestion is true, then any merging technique that weights common documents more heavily should be able to improve the precision of a meta-search system. This is the so-called "chorus effect" that was proposed by Vogt [13], which posits that meta-search system lie in improving precision in this way.

Nevertheless, it must be questioned whether it is only "duplicate" relationships that can be utilized to improve performance, or whether there are other relationships that can be incorporated into the merging process. It has been found that weighting common (duplicated) documents more heavily can improve the retrieval performance of a meta-search system when the underlying search engines retrieve many of the same relevant documents but different irrelevant documents. However, search engines that use different retrieval algorithms also retrieve many documents that are inter-related and relevant to the query, especially if they are in the same class. We therefore investigate the effect on the merging performance by considering more types of relationships. The proposed methodology that is used in our algorithms does not only consider the "duplicate" relationship, but also other kinds of relationships, which are introduced later. Actually considering other kinds of relationships gives a better performance than merely considering "duplicate" relationships. According to Ng and Kantor [70], if the participating search engines have similar or even the same output, where the same documents are ranked in a similar or even the same way for each search engine (there are duplicates for both relevant and irrelevant items), then placing a heavier weighting on common documents would not yield a significantly better performance, as the ranks of the irrelevant items would also be boosted along with the rankings of the relevant items. This does occur for some groups of search engines, which produce similar output and so there are not much distinct irrelevant items. However, relevant documents are in most cases also *inter-related*, whereas most irrelevant or less relevant documents are *unrelated*. Thus, if types of relationships other than duplication are taken into account and incorporated into merging algorithms that are derived from score-based "CombSum" or rank-based Agreement concepts, then documents that have multiple and strong relationships with other documents (most relevant) will gain higher weighting and a boosted rank. The performance of the meta-search system would thus ameliorated further, and the improvement would be significant. Considering various kinds of relationships allows a more all-round and fair way to reflect the relevance of items.

It maybe argued that some irrelevant items also have relationships, but we observe that in many meta-search engines irrelevant items are given low ranks by their original search engines, and thus their ranks would not be boosted much and would be kept at a low position after merging.

If most relevant items from the constituent search engines are inter-related while most irrelevant items are unrelated, which is proved in section 3.4.1, then assigning higher ranks to objects that have more related objects in any media format in the search results will improve a system's retrieval performance after merging. This is because more relevant items are retrieved and placed in high positions. For example, assume that we want to search for multimedia items about fruit, and in the search result pool we have three items that are in similar positions in the participating search engines: an image object with the hyperlink http://www.14ushop.com/grace/fruit/Apple.jpg and an image object with the hyperlink http://www.14ushop.com/grace/fruit/Peach.jpg that both come from the same category http://www.14ushop.com/grace/fruit/. After employing the proposed algorithm, they would be ranked higher than another web object with the hyperlink http://www.orange.com, which has no other objects linked to it or residing in classes that are related to it.

We now investigate the different types of relationships between the items that are retrieved by the search engines that participate in a meta-search. These may be in terms of their sites [21] and linkages [22], both of which are explained in the following section.

3.2 Types of relationships

There are two main types of relationships: (i) distance between two objects (the Web hierarchy) [21] and (ii) linkages between two objects [22], which can be further divided into several types: same class, sibling, cousin, parent, unidirectional and directional, and inlink and outlink. We can also take duplication as being a kind of relationship, in which the URLs of two objects are the same.

1. Distance between the two objects (Web hierarchy)

In this hierarchy, the directory is collapsed below a fixed depth of three and ignores (relatively few) documents above that depth. Therefore objects that are more than two levels apart are regarded to be unrelated.

(i) Same class [21] – Distance 0

Two objects are said to belong to the same classes if the distance between them is 0, that is, if they are in the same category. Figure 3.1 illustrates this relationship.

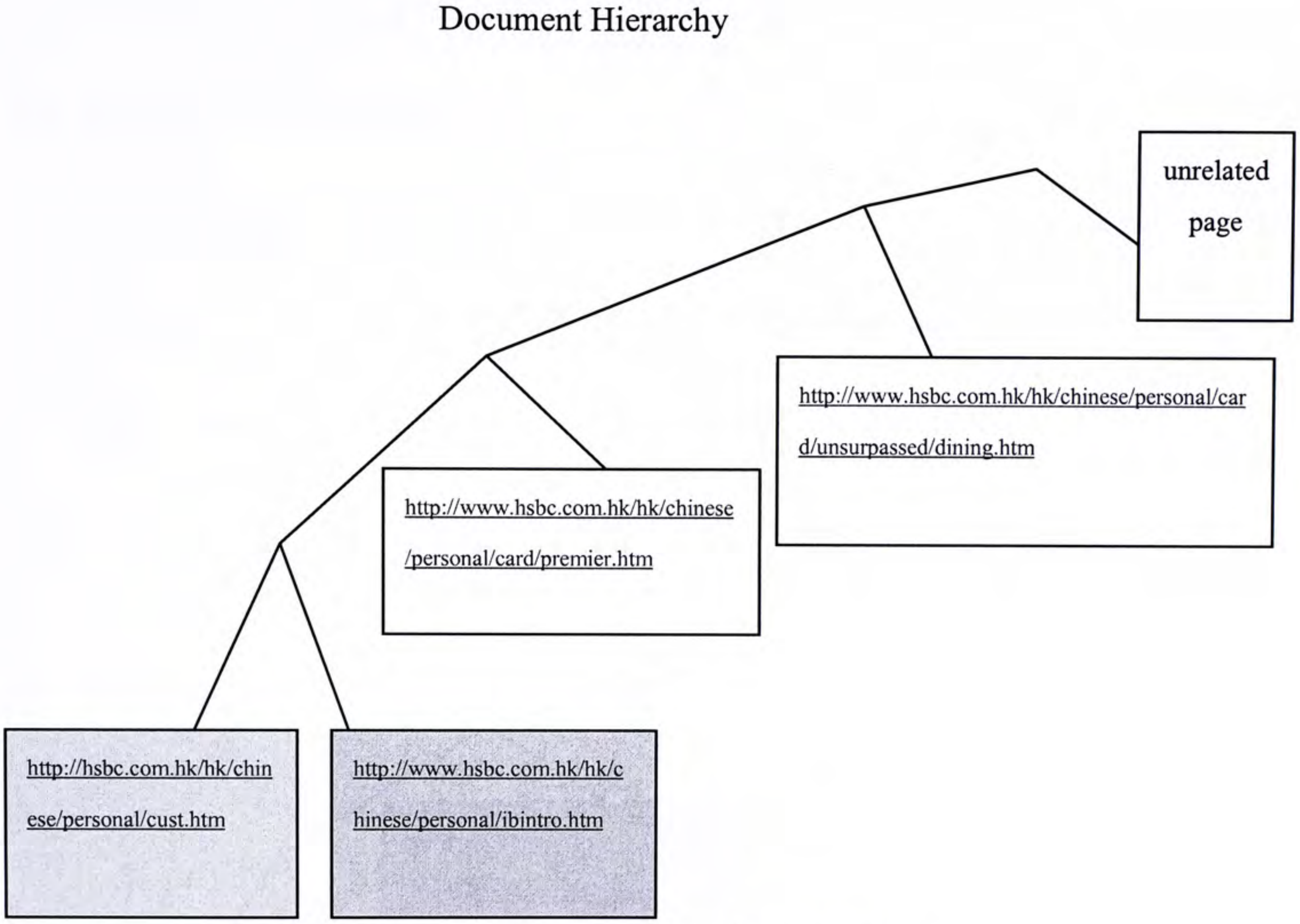

Document Hierarchy
unrelated page
http://www.hsbc.com.hk/hk/chinese/personal/card/unsurpassed/dining.htm
http://www.hsbc.com.hk/hk/chinese/personal/card/premier.htm
http://hsbc.com.hk/hk/chinese/personal/cust.htm
http://www.hsbc.com.hk/hk/chinese/personal/ibintro.htm

Figure 3.1 "Same Class" relationship

The highlighted boxes show two documents. The documents that are represented by the shaded box are in the "**Same Class**". For example, an image with the hyperlink http://hsbc.com.hk/hk/chinese/personal/cust.htm and another image object that resides at http://www.hsbc.com.hk/hk/chinese/personal/ibintro.htm, both of which come from the same category http://hsbc.com.hk/hk/chinese/personal/, are in the same class. Note that the two documents that reside just below the domain should not be regarded as belonging to the same class, for example, http://www.apple.com/fruit.html and http://www.apple.com/computer.html, as they may not belong to the same category.

(ii) Sibling class [21] – Distance 1

A medium object is said to be the sibling of another object if the distance between them is 1. That is, if they differ in their URL paths by one level. Figure 3.2 illustrates this relationship.

Document Hierarchy

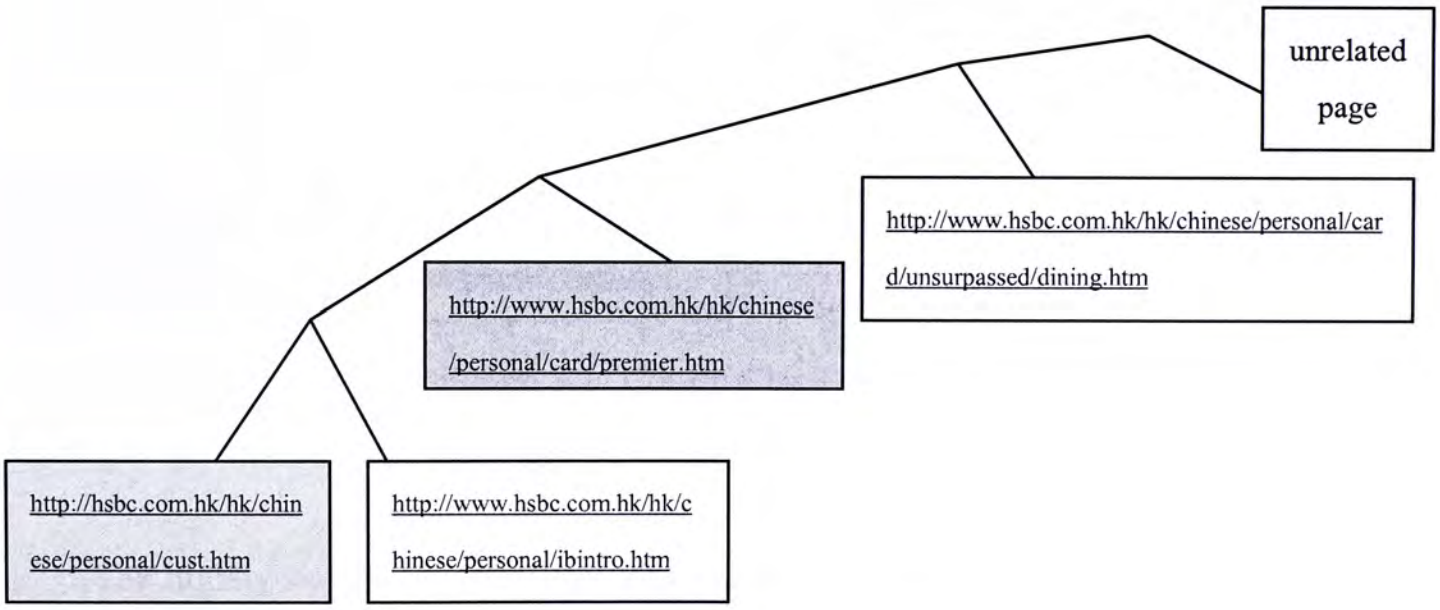

unrelated page
http://www.hsbc.com.hk/hk/chinese/personal/card/unsurpassed/dining.htm
http://www.hsbc.com.hk/hk/chinese/personal/card/premier.htm
http://hsbc.com.hk/hk/chinese/personal/cust.htm
http://www.hsbc.com.hk/hk/chinese/personal/ibintro.htm

Figure 3.2 “Sibling Class” relationship

The highlighted boxes show two documents. The documents that are represented by the shaded box are in the “**Sibling Class**”. For example, a web object with the hyperlink http://www.hsbc.com.hk/hk/chinese/personal/cust.htm and an image object that resides at http://www.hsbc.com.hk/hk/chinese/personal/card/premier.htm differ in their URL paths by one level, and are thus siblings of each other.

(iii) Cousin class [21] – Distance 2

A medium object is said to be the cousin of another object if the distance between them is 2, which means that they differ in their URL paths by two levels. Figure 3.3 illustrates this relationship.

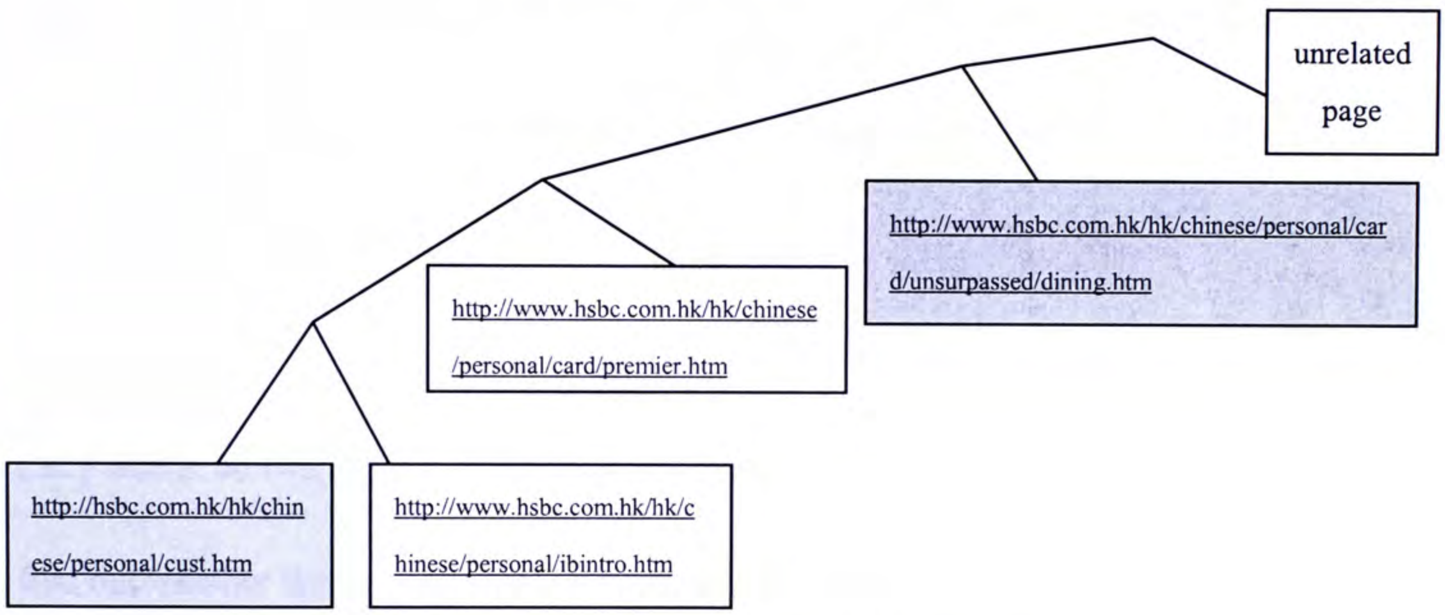

Document Hierarchy
unrelated
page
http://www.hsbc.com.hk/hk/chinese/personal/car
d/unsurpassed/dining.htm
http://www.hsbc.com.hk/hk/chinese
/personal/card/premier.htm
http://hsbc.com.hk/hk/chin
ese/personal/cust.htm
http://www.hsbc.com.hk/hk/c
hinese/personal/ibintro.htm

Figure 3.3 “Cousin Class” relationship

The highlighted boxes show two documents. The documents that are represented by the shaded box are in the “**Cousin Class**”. For example, an web object with the hyperlink http://www.hsbc.com.hk/hk/chinese/personal/cust.htm and another object with the hyperlink http://www.hsbc.com.hk/hk/chinese/personal/card/unsurpassed/dining.htm differ in their URL paths by two levels, and are thus cousins of each other.

It is, of course, possible that the location hierarchy does not accurately reflect document similarity. For example, documents that are in the subdirectory “recreation/autos” are almost certainly more similar to those in “shopping/autos” than to those in “recreation/smoking”. However, this effect is very small, and such cases are rare given that we average the statistics of many documents.

2. By linkages between the two document objects [22]

There are two main kinds of relationships for linkages between two objects, (I) sibling and (II) parent relationships, in which sibling can be further classified by the direction and the cardinality of the direction.

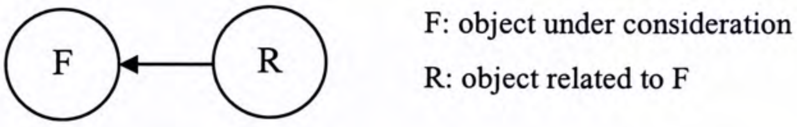
Let F and R be two objects of any media, where F is the object under consideration. We aim to find out whether the object has any related objects and decide its relevance to the query. R is the object that is related to F, and we classify and depict the relationship between them as follows.

(I) Sibling

Direct Link

a) Unidirectional

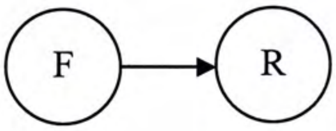
Inlink (direct inlink)



As is shown by the diagram, R is related to F through a direct inlink in a single direction.

For example, http://www.peu.cuhk.edu.hk/ is pointed by http://www.cuhk.edu.hk/en/minor.htm, as the latter page has a direct link to the former.

Outlink (direct outlink)

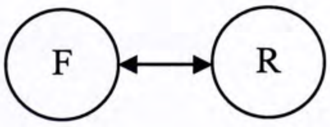


F: object under consideration
R: object related to F

As is shown by the diagram, R is related to F through a direct outlink from F to R in a single direction.

For example, http://www.peu.cuhk.edu.hk/ points to https://www.peu.cuhk.edu.hk/peu/schedule_c/, as the former has a direct link to the latter.

b) Bidirectional



F: object under consideration
R: object related to F

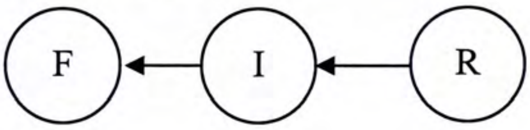
As is shown by the diagram, R is related to F through a bidirectional link.

For example, https://www.peu.cuhk.edu.hk/ and https://www.peu.cuhk.edu.hk/peu/ are related in that the two documents point to each other.

Indirect Link

a) Unidirectional

Inlink (indirect inlink)

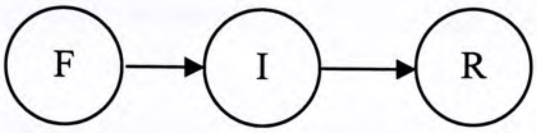


F: object under consideration
R: object related to F
I: intermediate object

As is shown by the diagram, R is related to F through an indirect inlink via an intermediate object I in a single direction.

For example, http://www.cuhk.edu.hk/en/minor.htm indirectly points to http://www.cuhk.edu.hk/itsc/onlineapp/facility/index.html through the intermediate http://www.cuhk.edu.hk/wbt/.

Outlink (indirect outlink)



F: object under consideration

R: object related to F

I: intermediate object

As is shown by the diagram, R is related to F through an indirect outlink from F to R via an intermediate object I in a single direction.

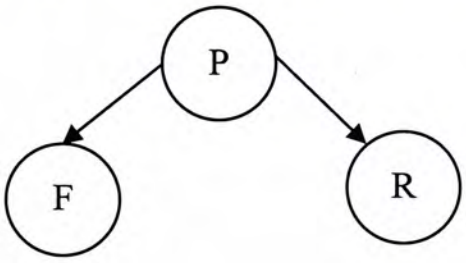
For example, http://www.cuhk.edu.hk/en/minor.htm indirectly points to http://www.cuhk.edu.hk/itsc/onlineapp/facility/index.html through the intermediate http://www.cuhk.edu.hk/wbt/.

b) Bidirectional

We regard this as a type of "Parent" relationship, which is discussed in the following.

(II) Parent

a) Both are pointed by the same object (parent)



F: object under consideration

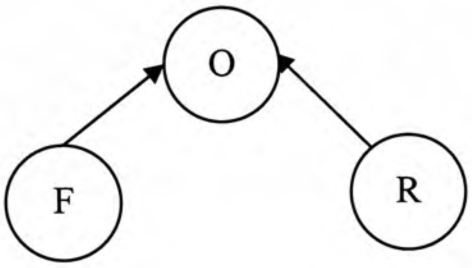
R: object related to F

P: parent object

F and R are related by an object (their parent) that points to them both.

For example, http://www.cuhk.edu.hk/en/news.htm and http://www.cuhk.edu.hk/en/research.htm are related via the parent http://www.cuhk.edu.hk/en/.

b) Both point to the same object



F: object under consideration
R: object related to F
O: an object

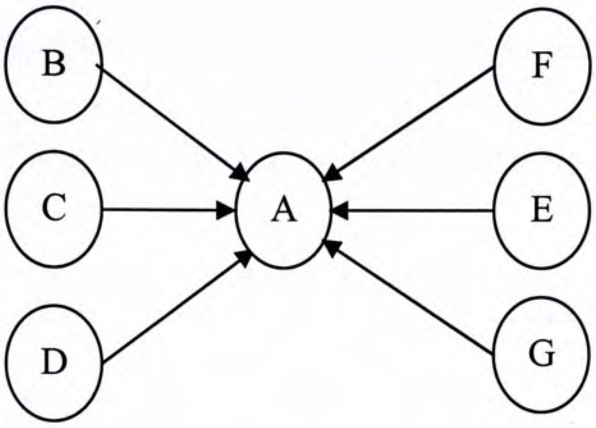
F and R are related by an object to which they both point.

For example, http://www.cuhk.edu.hk/en/ and http://www.cuhk.edu.hk/en/news.htm are related in that they both point to http://www.cuhk.edu.hk/en/map.htm.

There are no direct linkages between image, audio, and video objects, for example, a video object http://www.bbc.co.uk/london/realmedia/sport/davidbeckham.ram would not have a direct link with the image object http://www.bbc.net.uk/devon/fun/images/david_beckham/david_beckham_270.jpg, and thus the link of the page in which the real URL location of this object is embedded should both be considered. For example, in considering whether an image object is pointed by an audio object, we must find out whether the page in which the audio object is embedded has link to the image object. Similarly, in considering whether a video object points to an audio object, we must ascertain whether the page in which the video object is embedded has a link to the audio object.

However, we need not consider the page in which the object is embedded to find the relationships between objects of the "Web hierarchy" type, as we can use the real URL location of the object to compare its level with the object in the Web hierarchy that is being considered.

Thus, the linkage between two objects makes one object related to another. An object is said to be related to another object if it has a linkage) with that object. In the Internet environment, documents are linked by pointers, and these linkages can indicate the degree of importance of documents. It is argued that an important document is pointed by many documents. By this token, the official homepage of the computer company DELL, for example, is an important document, as there are many documents on the Web that point to it. Consider a query that consists of the single term "DELL". There may be thousands of documents on the Internet that contain this term, but it is likely that the user is interested only in the official homepage of DELL. Of all of the documents that contain the term "DELL," the official homepage of DELL is the most important due to the numerous links that point to it, and thus it can be presented to the user with a high rank if the merging algorithm boosts the rank of the homepage by considering its numerous relationships with other items.

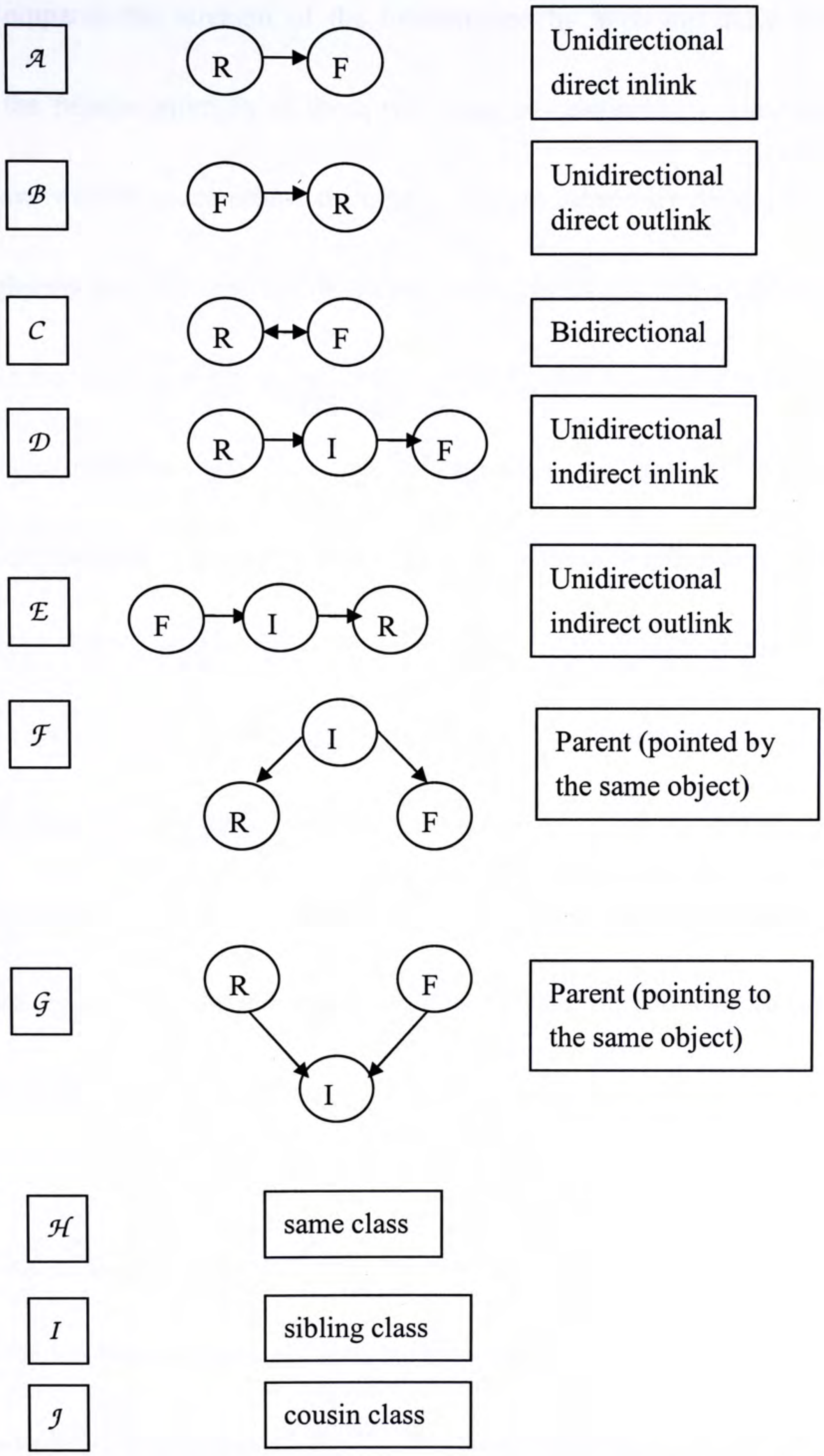


This diagram shows seven web objects. A is the official homepage of DELL, and B to G are other homepages that describe DELL in some way and have links that point to A. When a user searches for "DELL," it is supposed that A to G are retrieved in the result set, as they contain the term "DELL", but that A should be ranked the highest as it is the official homepage of DELL and will deliver the most information that is most relevant to the query. This can be achieved by assigning scores to the items according to the relationships of each retrieved item. Obviously, A has the largest number of relationships (six), whereas the others have just one relationship. This example demonstrates why we suggest that items that are more relevant to the query have more relationships with other items in the search results, and that there are some grounds for our relationship hypothesis, but further investigation and analysis are needed, which are given in section 3.4.1. In the following section, we discuss the strengths of the various relationships and their order.

3.3 Order of Strength of the Relationships

Different relationships have different strengths by which they may be ordered. A source document is on average more similar to a same-class document than to a sibling-class document, and is on average more similar to a sibling-class document than to a cousin-class document, and so on [21]. Thus, in general, if two objects are in the same class, then they are most likely to be in the same category in terms of their content. Such objects have the closest relationships of all of the relationships that have been mentioned, excluding the duplicate relationship, and thus the highest weighting would be given to such objects to boost their rank if the proposed algorithm includes the strength of relationships. By the principle that true similarity decreases with increased distance between two documents, in which distance is determined by the number of levels of difference in the document URLs, same-class objects have weaker relationships than duplicated objects, those with "sibling class" relationship would be less close, and much weaker for those with "cousin class" relationship. It is because the more similar the two objects are, the stronger is the relationship between them. However, it is possible that the principle, does not always hold, but it is reasonable to expect that a merging algorithm that accords with the aforementioned ordering of similarity will perform better than one that does not.

Furthermore, it is obvious that a bi-directional direct link relationship is closer than a unidirectional direct inlink relationship, a unidirectional direct inlink relationship is as strong as a unidirectional direct outlink relationship. A unidirectional indirect inlink relationship is as strong as a unidirectional indirect outlink relationship, but both of them are weaker than a direct link relationship. Parent relationship is the weakest among the linkage relationships, as the focused object has neither a direct nor indirect linkage with the related object, as the two are related by pointing to or being pointed by the same third object. However, the relationship between two objects with the same parent is weaker than that between two objects that point to the same object, as objects that emerge from the same parent may not be similar in content and topic, but those that point to the same object may be more similar and as they have a greater chance of presenting a similar topic. The following simplifies the comparison and lists the order of strength of the relationships.



𝒜
R
F
Unidirectional direct inlink
ℬ
F
R
Unidirectional direct outlink
𝒞
R
F
Bidirectional
𝒟
R
I
F
Unidirectional indirect inlink
ℰ
F
I
R
Unidirectional indirect outlink
ℱ
I
R
F
Parent (pointed by the same object)
𝒢
R
F
I
Parent (pointing to the same object)
ℋ
same class
ℐ
sibling class
𝒥
cousin class

According to the strength of relationship between objects, we have the following order.

$\mathcal{C} > \mathcal{A} = \mathcal{B} > \mathcal{D} = \mathcal{E} > \mathcal{G} > \mathcal{F}$

$\mathcal{H} > \mathcal{I} > \mathcal{J}$

F: focused object (object of which its relationships with other objects are to be determined)

R: related object of the focused object (F)

>: closer/stronger relationship = : equally strong relationship

Having compared the strength of the relationships by Web hierarchy and linkages, we must compare the relative strength of these two types of relationships. Generally, linkages between objects have weaker relationships than those that are related by the class in the Web hierarchy, because objects that differ in URL paths by two levels or less in the Web hierarchy belong to the same main category, and are more likely to describe the same or similar topic than objects that have linkages between them. However, this assertion may not always hold, as objects with a hierarchical relationship can have totally different content. For example, a news Web site might have links to finance and sports pages, which have different content but reside in the same class under the "news" category, whereas two other pages on the news site with direct linkages between them may deal with the same topic. In this case, the objects that are related by linkages (even indirectly) are more similar than those that are in the same class in the Web hierarchy. However, these cases are rare, and should not be significant in our algorithm as we average the statistics of many documents.

To differentiate the importance of these relationships, weighting should thus be assigned in order of the strengths of the relationships between objects. We use scoring models that assign a weight to each relationship, or criterion, to signify the relative importance of one candidate to another. A weighted score is then computed for each candidate.

3.3.1 Derivation of the weight for each kind of relationship (criterion)

There are various methods for deriving the weighting for each kind of relationship.

1. All of the criteria are listed in descending order of importance, and the least important (last-listed) criterion is assigned a value of 10. A numerical weight is then assigned to each criterion based on how important it is relative to the least important criterion. A criterion that is considered to be twice as important as the least important criterion is assigned a weight of 20. If the values cannot be agreed upon, then a sensitivity analysis should be performed.
2. Uniform or equal weights. Given N criteria, the weight for each criterion is $w_i = 1/N$. However, this method is not appropriate if the criteria have different relative importance, as in our case.
3. Rank sum weights. If R_i is the rank position of criterion i (where 1 is the highest rank) and there are N criteria, then the rank sum weights for each criterion can be calculated by $w_i = (N - R_i + 1) / (\sum_{k=1}^{N} N - R_k + 1)$.
4. Rank reciprocal weights. These weights may be calculated by

   $w_i = (1/ R_i) / (\sum_{k=1}^{N} 1 / R_k)$.

In this thesis, we use the rank sum weight method with the following formula to derive the weight for each relationship.

$$w_i = (N - R_i + 1) / (\sum_{k=1}^{N} N - R_k + 1)$$

Symbol	Relationship	Rank (R_i) Value	Weight assigned to the relationship
R_1	Same class	1	$w_1 = 0.158730$
R_2	Sibling class	2	$w_2 = 0.142857$
R_3	Cousin class	3	$w_3 = 0.126984$
R_4	Bidirectional	4	$w_4 = 0.111111$
R_5	Unidirectional direct inlink	5	$w_5 = 0.095238$
R_6	Unidirectional direct outlink	5	$w_5 = 0.095238$
R_7	Unidirectional indirect inlink	6	$w_6 = 0.079365$
R_8	Unidirectional indirect outlink	6	$w_6 = 0.079365$
R_9	Parent (pointed to the same object)	7	$w_7 = 0.063492$
R_{10}	Parent (pointing by the same object)	8	$w_8 = 0.047619$

N = 10

Table 3.1 Labeling of the relationships and their associated weights

We exclude "duplicate" relationship from this list, and instead assign weighting of 1.00 to such relationship, as the duplicate relationship is a very strong relationship that must have a much higher rating than other types of relationships. Therefore, by assigning rankings to the relationships according to their strengths, we can obtain their weights using the aforementioned method.

3.4 Observation of the relationships between retrieved objects and the effects of these relationships on the relevance of objects

3.4.1 Observation on the relationships existed in items that are irrelevant and relevant to the query

We have suggested that the number of relationships that an item has with other items in the search results from the participating search engines, together with the strength of these relationships, has a positive effect on the relevance of the item. However, we need more observation and proof to support this assertion, and thus investigate using ten queries.

We look at whether the most irrelevant items have more "no relationship", or "a few weak relationships" than relevant items, and whether most relevant items have more "many relationships" (including weak and strong), or "a few strong relationships", compared to irrelevant items.

We introduce the following notation.

Classified by the relationships the item has with another item	Classified by the relevance of the item
N – No relationship	R – Page is relevant to the query
FW – A few weak relationships	IR – Page is irrelevant to the query
M – Many relationships (including weak and strong)	NA – Page is not accessible
FS – A few strong relationships	

F – Item under focus
R – Item that is related to F in the retrieved results from the participating search engines
E – Items in the retrieved results from the participating search engines

To state how we regard F as bearing no, many, few, weak, or strong relationships with E, the following notation applies.

No – F has no relationship with E.

Few – F has ten or less relationships, as most items in the retrieved results bear up to twenty relationships, some even bear more than fifty, so items that bear ten or less relationships are regarded to bear a few relationships with E.

Many – Oppositely to the above, F has more than ten relationships and are regarded to bear many relationships with E.

In defining "strong" and "weak" relationships, we need to consider both the strength of the original relationship and the original rank of the related item's original rank from its underlying search engine, which together yield the ***overall*** relationship between F and R. We use this overall relationship to evaluate the importance of the item under consideration (F) to the given query. A strong overall relationship supports the importance of F to the given query. The following definitions of "strong" and "weak" give a clearer picture of this idea.

Strong – F has strong overall relationships with E, such as duplicate or same-class relationships. Note that this definition also depends on the original ranking of R, for example, if item X has a duplicate item Y, but Y was ranked extremely low originally, then X cannot be regarded to have strong overall relationship with Y because bearing a duplicate that has a low original rank provides little support for the importance of X to the given query. Alternatively, if item X has a is related to item Y as a cousin, even though the cousin class is not a very strong type of relationship, if Y has a very high original rank, then X can be regarded as having a strong overall relationship with Y. Some relationships are so weak that only when the original rank of R is high enough will the overall relationship be regarded as strong, for example, if item X has a related item Y that has a very high original rank and both X and Y point to the same third object, then X can be regarded as having a strong overall relationship with Y. In fact, the weaker the relationship between two items, the higher the original rank of R must be to support the importance of F to the given query, and thus the overall relationship can only be regarded as strong after consideration of the original rank R.

Weak – F has a weak overall relationship with E, such as the "indirect link", "point to", and "pointed by" relationships that are mentioned in Section 3.2. If the original rank of R is also low, then the overall relationship is considered to be weak.

Thus, the original rank R is required to determine whether an overall relationship is strong or weak. We have adopted the following policy to classify the overall strength of a relationship. The left column sorts the relationship according to its original strength (the closeness or similarity between F and R) in descending order, which is discussed in section 3.3.1. Note that the weaker the relationship between F and R, the higher the original rank of R must be for the overall relationship to be deemed strong.

Type of relationship between F and R	**R's original rank in its underlying search engine**
Duplicate	1st – 10th (relationship is strong if the rank is 10th or

	higher, otherwise it is weak)
Same class	1st – 8th (relationship is strong if the rank is 8th or higher, otherwise it is weak)
Sibling class	1st – 7th
Cousin class	1st – 6th
Bidirectional	1st – 5th
Unidirectional direct inlink	1st – 4th
Unidirectional direct outlink	1st - 4th
Unidirectional indirect inlink	1st - 3rd
Unidirectional indirect outlink	1st - 3rd
Parent (pointed to the same object)	1st - 2nd
Parent (pointing by the same object)	1st only

Using these metrics, we analyze each item's relationships with E in the categories N, M, FW, and FS. We then ascertain how many items have no relationship with other items if the items are irrelevant, that is, the number of items that have no relationships divided by number of irrelevant items. This is equivalent to the probability of obtaining an item that has no relationships given that the items selected are irrelevant, P (N | IR). We also investigate how many items have no relationship with other items if the items are relevant, which is equivalent to the probability of obtaining an item that has no relationships given that the items selected are relevant, P (N | R).

If P (N | IR) > P (N | R), then the proportion of irrelevant items without any relationship is larger than the proportion of relevant items without any relationship. If the algorithm gives a zero score to items without relationships, then the ranking performance will be improved, as more irrelevant items will be given no score and lower ranks, compared to the relevant items.

We then find out how many items have a few weak relationships given the items are irrelevant, that is, the number of items that have a few weak relationships divided by the number of irrelevant items. This is equivalent to the probability of obtaining an item that has a few weak relationships given the items selected are irrelevant, P (FW | IR). We also investigate how many items have a few relationships with other items if the items are relevant, which is equivalent to the probability of obtaining an item that has a few weak relationships given that the items selected are relevant, P (FW | R).

If P (FW | IR) > P (FW | R), then the proportion of irrelevant items with a few weak relationships is larger than the proportion of relevant items with a few weak relationships. If the algorithm gives a smaller score to items with a few weak relationships, then the ranking performance will be improved, as more irrelevant items will be given a lower score and ranks, compared to the relevant items.

We next determine how many items have many relationships, including weak and strong, if the items are *irrelevant*, that is, number of items that have many relationships divided by the number of irrelevant items. This is equivalent to the probability of obtaining an item that has *many* relationships given that the items selected are irrelevant, P (M | IR). We also determine how many items have many relationships with other items if the items are relevant. This is equivalent to the probability of obtaining an item that has many relationships given that the items selected are relevant, P $(M \mid R)$.

If $P\ (M \mid IR) < P\ (M \mid R)$, then the proportion of irrelevant items with many relationships is smaller than the proportion of relevant items with many relationships. If the algorithm gives higher scores to items with many relationships, then the ranking performance will be improved, as more relevant items are given higher scores and ranks, compared to the irrelevant items.

We further determine how many items that have a few strong relationships if the items are irrelevant, that is, the number of items that have a few strong relationships divided by the number of irrelevant items. This is equivalent to the probability of obtaining an item that has a few strong relationships given that the items selected are irrelevant, $\mathrm{P\ (FS \mid IR)}$. We also determine how many items have a few strong relationships with other items if the items are relevant, which is equivalent to the probability of obtaining an item that has a few strong relationships given that the items selected are relevant, $\mathrm{P\ (FS \mid R)}$.

If $\mathrm{P\ (FS \mid IR) < P\ (FS \mid R)}$, then the proportion of irrelevant items with strong relationships is smaller than the proportion of relevant items with strong relationships. If the algorithm gives higher scores to items with strong relationships, then the ranking performance will be improved, as more relevant items are given higher scores and ranks, compared to the irrelevant items.

However, comparing $P(N \mid IR)$ and $P(N \mid R)$ and $P(FW \mid IR)$ and $P(FW \mid R)$ alone is not representative and meaningful enough if we want to study whether irrelevant items are more likely than relevant items to have no relationships or just a few weak relationships with other items. Similarly, comparing $P(M \mid IR)$ and $P(M \mid R)$ and $P(FS \mid IR)$ and $P(FS \mid R)$ alone is not representative and meaningful enough if we want to study whether relevant items are more likely than irrelevant items to have many relationships or a few strong relationships with other items. Therefore, we need to find the sum of all of these relationship sets and conduct the comparison at the end. That is, we check whether $[P(N \mid IR) + P(FW \mid IR)] > [P(N \mid R) + P(FW \mid R)]$ and $[P(M \mid IR) + P(FS \mid IR)] < [P(M \mid R) + P(FS \mid R)]$, and if so, then by assigning a score to an item based on its relationships will lead to the result that an item with no or fewer relationships is ranked lower than an item with more relationships, and so is an irrelevant item.

To investigate whether our algorithm is able to improve the ranking performance of a meta-search engine, we need to ascertain whether **(1)** $P(N \mid IR) > P(N \mid R)$; **(2)** $P(FW \mid IR) > P(FW \mid R)$; **(3)** $P(M \mid IR) < P(M \mid R)$; (4) $P(FS \mid IR) < P(FS \mid R)$.

Note that $X \gg Y$ means that X is significantly larger than Y at $\alpha = 0.1$ significance level, and $X \ll Y$ means that X is significantly less than Y at $\alpha = 0.1$ significance level.

(1). Compare P (N | IR) and P (N | R)

We first find out the number of irrelevant items that have no relationships with other items of the total number of irrelevant items and the number of relevant items that have no relationships with other items of the total number of relevant items.

Hence,

$$P(N \mid IR) = \frac{\text{number of irrelevant items that bear no relationships}}{\text{number of irrelevant items}}$$

$$P(N \mid R) = \frac{\text{number of relevant items that bear no relationships}}{\text{number of relevant items}}$$

We analyze the ten keywords that will be used in the experiment. These keywords were chosen as they are more specific query terms that facilitate the evaluation of the relevance of the retrieved items. That is, users find it convenient and easy to judge the relevance of the items, as there are fewer ambiguities. For example, if users are searching for "Lord of the Rings", then most of the results will be about the story written by J. R. R. Tolkien. This helps to narrow down the targeted scope and set a standard for users to judge whether the items are relevant to the query term. However, if "orange" is searched and users are asked to evaluate the relevance of the retrieved items, then the judgment would be more difficult, as results about the category such as a fruit, color, the mobile service provider in Hong Kong and Macau are returned.

The results of the observation are summarized in the following table.

Keyword	Media	P (N | IR)	P (N | R)	Comparison
David Beckham	Web	32/53=0.603774	68/226=0.300885	P (N | IR) >> P (N | R)
	Image	8/24=0.333333	83/229=0.362445	***P (N | IR) < P (N | R)***
	Audio	5/14=0.357143	9/71=0.126761	P (N | IR) >> P (N | R)
	Video	2/22=0.090909	4/112=0.035714	P (N | IR) >> P (N | R)
Faye Wong	Web	18/40=0.45	74/236=0.313559	P (N | IR) >> P (N | R)
	Image	30/52=0.576923	72/211=0.341232	P (N | IR) >> P (N | R)
	Audio	9/13=0.692308	17/130=0.130769	P (N | IR) >> P (N | R)
	Video	Not Applicable	Not Applicable	Not Applicable
Janet Jackson	Web	35/51=0.686275	92/219=0.420091	P (N | IR) >> P (N | R)
	Image	61/139=0.438849	31/124=0.25	P (N | IR) >> P (N | R)
	Audio	37/145=0.255172	2/48=0.041667	P (N | IR) >> P (N | R)
	Video	1/47=0.021277	0/23=0	P (N | IR) >> P (N | R)
Moulin Rouge	Web	57/87=0.655172	51/198=0.257576	P (N | IR) >> P (N | R)
	Image	67/143=0.468531	28/109=0.256881	P (N | IR) >> P (N | R)
	Audio	24/73=0.328767	31/175=0.177143	P (N | IR) >> P (N | R)
	Video	0/12=0	7/113=0.061947	***P (N | IR) << P (N | R)***
Lord of the Rings	Web	26/78=0.333333	46/212=0.216981	P (N | IR) >> P (N | R)
	Image	21/28=0.75	39/228=0.171053	P (N | IR) >> P (N | R)
	Audio	11/25=0.44	19/190=0.1	P (N | IR) >> P (N | R)
	Video	0/29=0	1/153=0.006536	***P (N | IR) < P (N | R)***
Liv Tyler	Web	14/24=0.583333	74/259=0.285714	P (N | IR) >> P (N | R)
	Image	8/12=0.666667	49/259=0.189189	P (N | IR) >> P (N | R)
	Audio	2/8=0.25	1/54=0.018519	P (N | IR) >> P (N | R)
	Video	6/13=0.461538	2/82=0.02439	P (N | IR) >> P (N | R)
Keanu Reeves	Web	26/43=0.604651	89/248=0.358871	P (N | IR) >> P (N | R)
	Image	42/76=0.552632	31/172=0.180233	P (N | IR) >> P (N | R)
	Audio	15/67=0.223881	8/165=0.048485	P (N | IR) >> P (N | R)
	Video	8/64=0.125	3/61=0.04918	P (N | IR) >> P (N | R)
Norah Jones	Web	15/33=0.454545	83/230=0.36087	P (N | IR) >> P (N | R)
	Image	9/13=0.692308	54/236=0.228814	P (N | IR) >> P (N | R)
	Audio	10/17=0.588235	35/110=0.318182	P (N | IR) >> P (N | R)
	Video	3/5=0.6	3/21=0.142857	P (N | IR) >> P (N | R)
Ricky Martin	Web	21/34=0.617647	96/225=0.426667	P (N | IR) >> P (N | R)
	Image	15/20=0.75	57/176=0.323864	P (N | IR) >> P (N | R)
	Audio	19/43=0.44186	61/125=0.488	***P (N | IR) < P (N | R)***
	Video	8/23=0.347826	10/60=0.166667	P (N | IR) >> P (N | R)
Orlando Bloom	Web	27/40=0.675	71/216=0.328704	P (N | IR) >> P (N | R)
	Image	14/23=0.608696	52/222=0.234234	P (N | IR) >> P (N | R)

	Audio	0/2=0	3/153=0.019608	***P (N | IR) << P (N | R)***
	Video	0/6=0	2/33=0.060606	***P (N | IR) << P (N | R)***

Table 3.2 Results of P (N | IR) and P (N | R) and their comparisons across four media formats for the ten queries

* **Not Applicable** means there are no irrelevant or relevant items in the search results, and thus we cannot find the corresponding proportion or draw a comparison.

From Table 3.2, we can see that for most of the media of different queries (33/39=11/13), there are many more items that are irrelevant to the query that have no relationships with other items than relevant items with no relationships [P (N | IR) >> P (N | R)]. For some queries for certain media, there are more relevant items that have no relationships with other items than irrelevant items [P (N | R) > P (N | IR)], but the number is not significant.

(2) Compare P (FW | IR) and P (FW | R)

We first find out the number of irrelevant items that have a few weak relationships with other items of the total number of irrelevant items, and the number of relevant items that have a few weak relationships with other items of the total number of relevant items.

Hence,

$$P\ (FW \mid IR) = \frac{\text{number of irrelevant items that bear a few weak relationships}}{\text{number of irrelevant items}}$$

$$P\ (FW \mid R) = \frac{\text{number of relevant items that bear a few weak relationships}}{\text{number of relevant items}}$$

We analyze the 10 keywords that will be used in the experiment and summarize the results in the following table.

Keyword	**Media**	**P (FW | IR)**	**P (FW | R)**	**Comparison**
David Beckham	Web	19/53=0.358491	93/226=0.411504	***P(FW | IR) < P(FW | R)***
	Image	14/24=0.583333	81/229=0.353712	P(FW | IR) >> P(FW | R)
	Audio	6/14=0.428571	25/71=0.352113	P(FW | IR) > P(FW | R)
	Video	18/22=0.818182	58/112=0.517857	P(FW | IR) >> P(FW | R)
Faye Wong	Web	18/40=0.45	78/236=0.330508	P(FW | IR) >> P(FW | R)
	Image	18/52=0.346154	34/211=0.161137	P(FW | IR) >> P(FW | R)
	Audio	4/13=0.307692	41/130=0.315385	***P(FW | IR) < P(FW | R)***
	Video	Not Applicable	Not Applicable	Not Applicable
Janet Jackson	Web	10/51=0.196078	64/219=0.292237	***P(FW | IR) << P(FW | R)***
	Image	50/139=0.359712	14/124=0.137097	P(FW | IR) >> P(FW | R)
	Audio	81/145=0.558621	26/48=0.541667	P(FW | IR) > P(FW | R)
	Video	28/47=0.595745	7/23=0.304348	P(FW | IR) >> P(FW | R)
Moulin Rouge	Web	26/87=0.298851	66/198=0.333333	***P(FW | IR) < P(FW | R)***
	Image	66/143=0.461538	27/109=0.247706	P(FW | IR) >> P(FW | R)
	Audio	37/73=0.506849	44/175=0.251429	P(FW | IR) >> P(FW | R)
	Video	6/12=0.5	23/113=0.20354	P(FW | IR) >> P(FW | R)
Lord of the Rings	Web	37/78=0.474359	57/212=0.268868	P(FW | IR) >> P(FW | R)
	Image	5/28=0.178571	8/228=0.035088	P(FW | IR) >> P(FW | R)
	Audio	12/25=0.48	25/190=0.131579	P(FW | IR) >> P(FW | R)
	Video	8/29=0.275862	19/153=0.124183	P(FW | IR) >> P(FW | R)
Liv Tyler	Web	7/24=0.291667	69/259=0.266409	P(FW | IR) > P(FW | R)
	Image	0/12=0	23/259=0.088803	***P(FW | IR) << P(FW | R)***
	Audio	2/8=0.25	1/54=0.018519	P(FW | IR) >> P(FW | R)
	Video	6/13=0.461538	1/82=0.012195	P(FW | IR) >> P(FW | R)
Keanu Reeves	Web	13/43=0.302326	60/248=0.241935	P(FW | IR) > P(FW | R)
	Image	26/76=0.342105	32/172=0.186047	P(FW | IR) >> P(FW | R)
	Audio	15/67=0.223881	22/165=0.133333	P(FW | IR) >> P(FW | R)
	Video	19/64=0.296875	0/61=0	P(FW | IR) >> P(FW | R)
Norah Jones	Web	15/33=0.454545	71/230=0.308696	P(FW | IR) >> P(FW | R)
	Image	4/13=0.307692	54/236=0.228814	P(FW | IR) > P(FW | R)
	Audio	5/17=0.294118	33/110=0.3	***P(FW | IR) < P(FW | R)***
	Video	2/5=0.4	6/21=0.285714	P(FW | IR) >> P(FW | R)

Ricky Martin	Web	8/34=0.235294	47/225=0.208889	P(FW | IR) > P(FW | R)
	Image	5/20=0.25	39/176=0.221591	P(FW | IR) > P(FW | R)
	Audio	12/43=0.27907	10/125=0.08	P(FW | IR) >> P(FW | R)
	Video	11/23=0.478261	16/60=0.266667	P(FW | IR) >> P(FW | R)
Orlando Bloom	Web	9/40=0.225	36/216=0.166667	P(FW | IR) > P(FW | R)
	Image	6/23=0.26087	30/222=0.135135	P(FW | IR) >> P(FW | R)
	Audio	2/2=1	5/153=0.03268	P(FW | IR) >> P(FW | R)
	Video	6/6=1	12/33=0.363636	P(FW | IR) >> P(FW | R)

Table 3.3 Results of P (FW | IR) and P (FW | R) and their comparisons across four media formats for the ten queries

From Table 3.3, we can see that for most of the media formats for the different queries (33/39=11/13), there are more items that are irrelevant to the query that have a few weak relationships” with other items than relevant items, and that 25 of the 33 results show a significant difference [P(FW | IR) >> P(FW | R)]. For some of the queries for certain media, more relevant items have a few relationships with other items than irrelevant items [P(FW | R) > P(FW | IR)], but the number is not significant.

(3) Compare P (N or FW | IR) and P (N or FW | R)

We first ascertain the number of irrelevant items that have no or a few relationships with other items of the total number of irrelevant items, and the number of relevant items that have no or a few relationships with other items of the total number of relevant items.

Hence,

$$P\ (N\ \text{or}\ FW \mid IR) = \frac{\text{number of irrelevant items that bear no or a few weak relationships}}{\text{number of irrelevant items}}$$

$$P\ (N\ \text{or}\ FW \mid R) = \frac{\text{number of relevant items that bear no or a few weak relationships}}{\text{number of relevant items}}$$

* We use A to represent N or FW relationships between items.

We analyze the 10 keywords that will be used in the experiment and summarize the results in the following table.

Keyword	**Media**	**P (A | IR)**	**P (A | R)**	**Comparison**
David Beckham	Web	51/53=0.962264	161/226=0.712389	* P(A | IR) >> P(A | R)
	Image	22/24=0.916667	164/229=0.716157	P(A | IR) >> P(A | R)
	Audio	11/14=0.785714	34/71=0.478873	P(A | IR) >> P(A | R)
	Video	20/22=0.909091	62/112=0.553571	P(A | IR) >> P(A | R)
Faye Wong	Web	36/40=0.9	152/236=0.644068	P(A | IR) >> P(A | R)
	Image	48/52=0.923077	106/211=0.50237	P(A | IR) >> P(A | R)
	Audio	13/13=1	58/130=0.446154	P(A | IR) >> P(A | R)
	Video	Not Applicable	Not Applicable	Not Applicable
Janet Jackson	Web	45/51=0.882353	146/219=0.666667	P(A | IR) >> P(A | R)
	Image	111/139=0.798561	48/124=0.387097	P(A | IR) >> P(A | R)
	Audio	118/145=0.813793	28/48=0.583333	P(A | IR) >> P(A | R)
	Video	29/47=0.617021	7/23=0.304348	P(A | IR) >> P(A | R)
Moulin Rouge	Web	83/87=0.954023	117/198=0.590909	P(A | IR) >> P(A | R)
	Image	133/143=0.93007	55/109=0.504587	P(A | IR) >> P(A | R)
	Audio	61/73=0.835616	75/175=0.428571	P(A | IR) >> P(A | R)
	Video	6/12=0.5	30/113=0.265487	P(A | IR) >> P(A | R)
Lord of the Rings	Web	63/78=0.807692	103/212=0.485849	P(A | IR) >> P(A | R)
	Image	26/28=0.928571	47/228=0.20614	P(A | IR) >> P(A | R)
	Audio	23/25=0.92	44/190=0.231579	P(A | IR) >> P(A | R)
	Video	8/29=0.275862	20/153=0.130719	P(A | IR) >> P(A | R)
Liv Tyler	Web	21/24=0.875	143/259=0.552124	P(A | IR) >> P(A | R)
	Image	8/12=0.666667	72/259=0.277992	P(A | IR) >> P(A | R)
	Audio	4/8=0.5	2/54=0.037037	P(A | IR) >> P(A | R)
	Video	12/13=0.923077	3/82=0.036585	P(A | IR) >> P(A | R)
Keanu Reeves	Web	39/43=0.906977	149/248=0.600806	P(A | IR) >> P(A | R)
	Image	68/76=0.894737	63/172=0.366279	P(A | IR) >> P(A | R)

	Audio	30/67=0.447761	30/165=0.181818	P(A | IR) >> P(A | R)
	Video	27/64=0.421875	3/61=0.04918	P(A | IR) >> P(A | R)
Norah Jones	Web	30/33=0.909091	154/230=0.669565	P(A | IR) >> P(A | R)
	Image	13/13=1	108/236=0.457627	P(A | IR) >> P(A | R)
	Audio	15/17=0.882353	68/110=0.618182	P(A | IR) >> P(A | R)
	Video	5/5=1	9/21=0.428571	P(A | IR) >> P(A | R)
Ricky Martin	Web	29/34=0.852941	143/225=0.635556	P(A | IR) >> P(A | R)
	Image	20/20=1	96/176=0.545455	P(A | IR) >> P(A | R)
	Audio	31/43=0.72093	71/125=0.568	P(A | IR) >> P(A | R)
	Video	19/23=0.826087	26/60=0.433333	P(A | IR) >> P(A | R)
Orlando Bloom	Web	36/40=0.9	107/216=0.49537	P(A | IR) >> P(A | R)
	Image	20/23=0.869565	82/222=0.369369	P(A | IR) >> P(A | R)
	Audio	2/2=1	8/153=0.052288	P(A | IR) >> P(A | R)
	Video	6/6=1	14/33=0.424242	P(A | IR) >> P(A | R)

Table 3.4 Results of P (N or FW | IR) and P (N or FW | R) and their comparisons across four media formats for the ten queries

From Table 3.4, we can see that for all of the media formats for different queries (39/39=1), there are more irrelevant items that have no relationships or a few weak relationships with other items than relevant items [P(N or FW | IR) >> P(N or FW | R)].

(4) Compare P (M | IR) and P (M | R)

We first find out the number of irrelevant items that have many relationships (including weak and strong) with other items of the total number of irrelevant items, and the number of relevant items that have many relationships (including weak and strong) with other items of the total number of relevant items. Usually, items that have ten or more relationships are said to have many relationships with other items.

Hence,

$$P(M \mid IR) = \frac{\text{number of irrelevant items that bear many relationships}}{\text{number of irrelevant items}}$$

$$P(M \mid R) = \frac{\text{number of relevant items that bear many relationships}}{\text{number of relevant items}}$$

We analyze the 10 keywords that will be used in the experiment and summarize the results in the following table.

Keyword	**Media**	**$P(M \mid IR)$**	**$P(M \mid R)$**	**Comparison**
David Beckham	Web	0/53=0	36/226=0.15929	$P(M \mid IR) \ll P(M \mid R)$
	Image	0/24=0	48/229=0.20961	$P(M \mid IR) \ll P(M \mid R)$
	Audio	3/14=0.21429	18/71=0.25352	$P(M \mid IR) < P(M \mid R)$
	Video	0/22=0	37/112=0.33036	$P(M \mid IR) \ll P(M \mid R)$
Faye Wong	Web	0/40=0	62/236=0.26271	$P(M \mid IR) \ll P(M \mid R)$
	Image	4/52=0.07692	88/211=0.41706	$P(M \mid IR) \ll P(M \mid R)$
	Audio	0/13=0	56/130=0.43077	$P(M \mid IR) \ll P(M \mid R)$
	Video	Not Applicable	Not Applicable	Not Applicable
Janet Jackson	Web	1/51=0.01961	34/219=0.15525	$P(M \mid IR) \ll P(M \mid R)$
	Image	19/139=0.13669	61/124=0.49194	$P(M \mid IR) \ll P(M \mid R)$
	Audio	17/145=0.11724	13/48=0.27083	$P(M \mid IR) \ll P(M \mid R)$
	Video	6/47=0.12766	6/23=0.26087	$P(M \mid IR) \ll P(M \mid R)$
Moulin Rouge	Web	1/87=0.01149	40/198=0.20202	$P(M \mid IR) \ll P(M \mid R)$
	Image	2/143=0.01399	31/109=0.2844	$P(M \mid IR) \ll P(M \mid R)$
	Audio	5/73=0.06849	74/175=0.42286	$P(M \mid IR) \ll P(M \mid R)$
	Video	6/12=0.5	59/113=0.52212	$P(M \mid IR) < P(M \mid R)$
Lord of the Rings	Web	7/78=0.08974	65/212=0.3066	$P(M \mid IR) \ll P(M \mid R)$
	Image	2/28=0.07143	171/228=0.75	$P(M \mid IR) \ll P(M \mid R)$
	Audio	0/25=0	135/190=0.71053	$P(M \mid IR) \ll P(M \mid R)$
	Video	18/29=0.62069	110/153=0.71895	$P(M \mid IR) \ll P(M \mid R)$
Liv Tyler	Web	1/24=0.04167	93/259=0.35907	$P(M \mid IR) \ll P(M \mid R)$
	Image	0/12=0	181/259=0.69884	$P(M \mid IR) \ll P(M \mid R)$
	Audio	0/8=0	37/54=0.68519	$P(M \mid IR) \ll P(M \mid R)$
	Video	0/13=0	76/82=0.92683	$P(M \mid IR) \ll P(M \mid R)$
Keanu Reeves	Web	1/43=0.02326	35/248=0.14113	$P(M \mid IR) \ll P(M \mid R)$
	Image	4/76=0.05263	99/172=0.57558	$P(M \mid IR) \ll P(M \mid R)$
	Audio	30/67=0.44776	126/165=0.76364	$P(M \mid IR) \ll P(M \mid R)$

	Video	31/64=0.48438	54/61=0.88525	$P(M \mid IR) \ll P(M \mid R)$
Norah Jones	Web	1/33=0.0303	40/230=0.17391	$P(M \mid IR) \ll P(M \mid R)$
	Image	0/13=0	101/236=0.42797	$P(M \mid IR) \ll P(M \mid R)$
	Audio	2/17=0.11765	29/110=0.26364	$P(M \mid IR) \ll P(M \mid R)$
	Video	0/5=0	7/21=0.33333	$P(M \mid IR) \ll P(M \mid R)$
Ricky Martin	Web	5/34=0.14706	71/225=0.31556	$P(M \mid IR) \ll P(M \mid R)$
	Image	0/20=0	69/176=0.39205	$P(M \mid IR) \ll P(M \mid R)$
	Audio	12/43=0.27907	52/125=0.416	$P(M \mid IR) \ll P(M \mid R)$
	Video	0/23=0	27/60=0.45	$P(M \mid IR) \ll P(M \mid R)$
Orlando Bloom	Web	4/40=0.1	102/216=0.47222	$P(M \mid IR) \ll P(M \mid R)$
	Image	3/23=0.13043	129/222=0.58108	$P(M \mid IR) \ll P(M \mid R)$
	Audio	0/2=0	140/153=0.91503	$P(M \mid IR) \ll P(M \mid R)$
	Video	0/6=0	17/33=0.51515	$P(M \mid IR) \ll P(M \mid R)$

Table 3.5 Results of P (M | IR) and P (M | R) and their comparisons across four media formats for the ten queries

From Table 3.5, we can see that for all of the media formats for different queries (39/39=1), there are more items that are relevant to the query that have many relationships including both weak and strong with other items than irrelevant items [P(M | R) >> P(M | IR)]. The difference is significantly large.

(5) Compare P (FS | IR) and P (FS | R)

We first ascertain the number of irrelevant items that have a few strong relationships with other items of the total number of irrelevant items, and the number of relevant items that have a few strong relationships with other items of the total number of relevant items.

Hence,

$$P\ (FS \mid IR) = \frac{\text{number of irrelevant items that bear a few strong relationships}}{\text{number of irrelevant items}}$$

$$P\ (FS \mid R) = \frac{\text{number of relevant items that bear a few strong relationships}}{\text{number of relevant items}}$$

We analyze the 10 keywords that will be used in the experiment and summarize the results in the following table.

Keyword	Media	P (FS | IR)	P (FS | R)	Comparison
David Beckham	Web	2/53=0.03774	29/226=0.12832	P(FS | IR) << P(FS | R)
	Image	2/24=0.08333	17/229=0.07424	***P(FS | IR) > P(FS | R)***
	Audio	0/14=0	19/71=0.26761	P(FS | IR) < P(FS | R)
	Video	2/22=0.09091	13/112=0.11607	P(FS | IR) < P(FS | R)
Faye Wong	Web	4/40=0.1	22/236=0.09322	***P(FS | IR) > P(FS | R)***
	Image	0/52=0	17/211=0.08057	P(FS | IR) << P(FS | R)
	Audio	0/13=0	16/130=0.12308	P(FS | IR) << P(FS | R)
	Video	Not Applicable	Not Applicable	Not Applicable
Janet Jackson	Web	5/51=0.09804	39/219=0.17808	P(FS | IR) << P(FS | R)
	Image	9/139=0.06475	15/124=0.12097	P(FS | IR) << P(FS | R)
	Audio	10/145=0.06897	7/48=0.14583	P(FS | IR) << P(FS | R)
	Video	12/47=0.25532	10/23=0.43478	P(FS | IR) << P(FS | R)
Moulin Rouge	Web	3/87=0.03448	41/198=0.20707	P(FS | IR) << P(FS | R)
	Image	8/143=0.05594	23/109=0.21101	P(FS | IR) << P(FS | R)
	Audio	7/73=0.09589	26/175=0.14857	P(FS | IR) < P(FS | R)
	Video	0/12=0	24/113=0.21239	P(FS | IR) << P(FS | R)
Lord of the Rings	Web	8/78=0.10256	44/212=0.20755	P(FS | IR) << P(FS | R)
	Image	0/28=0	10/228=0.04386	P(FS | IR) << P(FS | R)
	Audio	2/25=0.08	11/190=0.05789	***P(FS | IR) > P(FS | R)***
	Video	3/29=0.10345	23/153=0.15033	P(FS | IR) < P(FS | R)
Liv Tyler	Web	2/24=0.08333	23/259=0.0888	P(FS | IR) < P(FS | R)
	Image	4/12=0.33333	6/259=0.02317	***P(FS | IR) >> P(FS | R)***
	Audio	4/8=0.5	15/54=0.27778	***P(FS | IR) >> P(FS | R)***
	Video	1/13=0.07692	3/82=0.03659	***P(FS | IR) > P(FS | R)***
Keanu Reeves	Web	3/43=0.06977	64/248=0.25806	P(FS | IR) << P(FS | R)
	Image	4/76=0.05263	10/172=0.05814	P(FS | IR) < P(FS | R)
	Audio	7/67=0.10448	9/165=0.05455	***P(FS | IR) >> P(FS | R)***
	Video	6/64=0.09375	4/61=0.06557	***P(FS | IR) > P(FS | R)***
Norah Jones	Web	2/33=0.06061	36/230=0.15652	P(FS | IR) << P(FS | R)
	Image	0/13=0	27/236=0.11441	P(FS | IR) << P(FS | R)
	Audio	0/17=0	13/110=0.11818	P(FS | IR) << P(FS | R)

	Video	0/5=0	5/21=0.2381	P(FS | IR) << P(FS | R)
Ricky Martin	Web	0/34=0	11/225=0.04889	P(FS | IR) << P(FS | R)
	Image	0/20=0	11/176=0.0625	P(FS | IR) << P(FS | R)
	Audio	0/43=0	2/125=0.016	P(FS | IR) < P(FS | R)
	Video	4/23=0.17391	7/60=0.11667	***P(FS | IR) > P(FS | R)***
Orlando Bloom	Web	0/40=0	7/216=0.03241	P(FS | IR) << P(FS | R)
	Image	0/23=0	11/222=0.04955	P(FS | IR) << P(FS | R)
	Audio	0/2=0	5/153=0.03268	P(FS | IR) << P(FS | R)
	Video	0/6=0	2/33=0.06061	P(FS | IR) << P(FS | R)

Table 3.6 Results of P (FS | IR) and P (FS | R) and their comparisons across four media formats for the ten queries

From Table 3.6, we can see that for most of the media formats for the different queries (30/39=10/13), there are more relevant items that have a few strong relationships with other items than irrelevant items, and 23 of the 30 results show a significant difference [P(FS | R) >> P(FS | IR)]. For some queries for certain media, there are more irrelevant items that have a few strong relationships with other items than relevant items, but the number is not significant [P(FS | IR) > P(FS | R)].

(6) Compare P (M or FS | IR) and P (M or FS | R)

We determine the number of irrelevant items that have many or a few strong relationships with other items of the total number of irrelevant items, and the number of relevant items that have many or a few strong relationships with other items of the total number of relevant items.

Hence,

$$\text{P (M or FS | IR)} = \frac{\text{number of irrelevant items that bear many or a few strong relationships}}{\text{number of irrelevant items}}$$

$$\text{P (M or FS | R)} = \frac{\text{number of relevant items that bear many or a few strong relationships}}{\text{number of relevant items}}$$

* We use B to represent M or FS relationships between items.

We analyze the 10 keywords that will be used in the experiment and summarize the results in the following table.

Keyword	Media	P (M or FS | IR)	P (M or FS | R)	Comparison
David Beckham	Web	2/53=0.03774	65/226=0.28761	P(B | IR) << P(B | R)
	Image	2/24=0.08333	65/229=0.28384	P(B | IR) << P(B | R)
	Audio	3/14=0.21429	37/71=0.52113	P(B | IR) << P(B | R)
	Video	2/22=0.09091	50/112=0.44643	P(B | IR) << P(B | R)
Faye Wong	Web	4/40=0.1	84/236=0.35593	P(B | IR) << P(B | R)
	Image	4/52=0.07692	105/211=0.49763	P(B | IR) << P(B | R)
	Audio	0/13=0	72/130=0.55385	P(B | IR) << P(B | R)
	Video	Not Applicable	Not Applicable	Not Applicable
Janet Jackson	Web	6/51=0.11765	73/219=0.33333	P(B | IR) << P(B | R)
	Image	28/139=0.20144	76/124=0.6129	P(B | IR) << P(B | R)
	Audio	27/145=0.18621	20/48=0.41667	P(B | IR) << P(B | R)
	Video	18/47=0.38298	16/23=0.69565	P(B | IR) << P(B | R)
Moulin Rouge	Web	4/87=0.04598	81/198=0.40909	P(B | IR) << P(B | R)
	Image	10/143=0.06993	54/109=0.49541	P(B | IR) << P(B | R)
	Audio	12/73=0.16438	100/175=0.57143	P(B | IR) << P(B | R)
	Video	6/12=0.5	83/113=0.73451	P(B | IR) << P(B | R)
Lord of the Rings	Web	15/78=0.19231	109/212=0.51415	P(B | IR) << P(B | R)
	Image	2/28=0.07143	181/228=0.79386	P(B | IR) << P(B | R)
	Audio	2/25=0.08	146/190=0.76842	P(B | IR) << P(B | R)
	Video	21/29=0.72414	133/153=0.86928	P(B | IR) << P(B | R)
Liv Tyler	Web	3/24=0.125	116/259=0.44788	P(B | IR) << P(B | R)
	Image	4/12=0.33333	187/259=0.72201	P(B | IR) << P(B | R)
	Audio	4/8=0.5	52/54=0.96296	P(B | IR) << P(B | R)
	Video	1/13=0.07692	79/82=0.96341	P(B | IR) << P(B | R)
Keanu Reeves	Web	4/43=0.09302	99/248=0.39919	P(B | IR) << P(B | R)
	Image	8/76=0.10526	109/172=0.63372	P(B | IR) << P(B | R)

	Audio	37/67=0.55224	135/165=0.81818	P(B | IR) << P(B | R)
	Video	37/64=0.57813	58/61=0.95082	P(B | IR) << P(B | R)
Norah Jones	Web	3/33=0.09091	76/230=0.33043	P(B | IR) << P(B | R)
	Image	0/13=0	128/236=0.54237	P(B | IR) << P(B | R)
	Audio	2/17=0.11765	42/110=0.38182	P(B | IR) << P(B | R)
	Video	0/5=0	12/21=0.57143	P(B | IR) << P(B | R)
Ricky Martin	Web	5/34=0.14706	82/225=0.36444	P(B | IR) << P(B | R)
	Image	0/20=0	80/176=0.45455	P(B | IR) << P(B | R)
	Audio	12/43=0.27907	54/125=0.432	P(B | IR) << P(B | R)
	Video	4/23=0.17391	34/60=0.56667	P(B | IR) << P(B | R)
Orlando Bloom	Web	4/40=0.1	109/216=0.50463	P(B | IR) << P(B | R)
	Image	3/23=0.13043	140/222=0.63063	P(B | IR) << P(B | R)
	Audio	0/2=0	145/153=0.94771	P(B | IR) << P(B | R)
	Video	0/6=0	19/33=0.57576	P(B | IR) << P(B | R)

Table 3.7 Results of P (M or FS | IR) and P (M or FS | R) and their comparisons across four media formats for the ten queries

From Table 3.7, we can see that for all of the media formats for different queries (39/39=1), there are more relevant items that have many relationships including both weak and strong or a few strong relationships with other items than irrelevant items [P(M or FS | R) >> P(M or FS | IR)].

To conclude, we observe that items that are relevant to the query are more likely to have many relationships or a few strong relationships with other items than items that are irrelevant to the query, whereas items that are irrelevant to the query are more likely to have no relationships or a few weak relationships with other items than items that are relevant to the query. This supports our claim that items that are more relevant to the query have either more relationships or stronger relationships with the other items in the search results.

Therefore, if the proposed algorithm gives a score to an item according to the number and strengths of its relationships with other items, that is, if higher scores are given to the items that have many or a few strong relationships with other items and lower scores are given to the items that have no or a few weak relationships with others, then higher scores will be given to items that are more relevant to the query, which will boost the ranking of the relevant items.

Using this scoring scheme, an item with no relationships would be given a zero relationship score, but would be assigned a score for the original rank that was given by its underlying search engine, as this reflects its relevance to a certain extent. If an item with no relationships was ranked high originally, then we would still give it a high score to take its original ranking into account during the merging process.

Hence, our algorithm assigns a score to an item according to two criteria:

1). *original ranking* - the item's original ranking from its source search engine, and 2). *the new add-in, relationships between items* - the numbers and strength of the relationships that the item has with other retrieved items in the participating search engines. The relevancy scores of the related items, which are rank-based, are also included. If an item has a high original rank, then it will be assigned a high score for the first part, the original ranking score, but if it has no relationships with other items in any media, then it will be assigned a score of zero for the second part, the relationship score.

Finally, the algorithm gives a total score to an item by summing these two parts to obtain the *system score* of the item. The algorithm merges and ranks the items using this system score, and the higher the system score, the higher the ranking of the item after fusion.

3.5 Proposed re-ranking algorithms

In the following, we explain the development of our proposed re-ranking methods.

After comparing the re-ranking methods that are discussed in Section 2.3, we selected a well-suited re-ranking method to modify to generate two multimedia meta-searching algorithms that will rank documents with relatively more related objects higher after merging. As document scores are not always available and snippets are not returned by image, audio or video objects, we found the Interleaving, Agreement, and Democratic data fusion rank-based merge algorithms to be best suited for modification. It may be queried which of these methods results in better ranking performance after modification, but this is not our research focus. Rather, our work is to test whether there is an improvement in the ranking performance of a meta-search engine after the incorporation of the relationship factor. We could not test the suitability of all of the merging methods for modification, and thus we employed certain metrics to choose one merging algorithm to be modified. Interleaving has more bias than Agreement, as it overestimates the relevance of items that have no duplicates in other results of the participating search engines and underestimates the relevance of items that have duplicates. Furthermore, the retrieval performance of meta-search engines that use this method depends on the interleaving order. Fox et al. [67] claimed that in recent testing of the system [68],´much emphasis had been placed on the number of times the file was identified by multiple search engines locating the same file (duplicate). The Agreement and Democratic data fusion methods are similar in their idea but items that have lower original ranks will be given higher votes in the latter method. As our algorithm uses the total system score of an item to represent its relevance by adding together the original ranking score and the relationship score and allows an item with a high original rank to obtain a higher score, the Agreement method was chosen for modification.

Therefore our proposed merging method evolves from the concept of Agreement, and we would develop a new re-ranking algorithm by modifying Agreement merge algorithm.

In the following, we introduce several merging algorithms, two original (unmodified) and two modified, which are later compared to ascertain whether the modified algorithm is better. We also compare the two modified algorithms to see which is better.

3.5.1 Original re-ranking algorithm (before modification)

Algorithm 1 (namely Agreement 1)

The unmodified Agreement algorithm, which is addressed in other literature [8], is equivalent to a merging mechanism that uses a non-linear scoring method but does not consider other kinds of relationships (except duplicates).´ However, here we simulate the ranking of a document as the score that is obtained by that document, and normalize it first using standard normalization to make the simulated score from each system comparable, which solves the problem discussed in Section 2.3.

This unmodified algorithm is named Agreement 1, and the formula for this algorithm is as follows.

$$\text{score}(l_i^s)\text{''} = \text{score}(l_i^s) + \sum_{duplicate} [1/\text{rank}\ (l_j^e, e)]$$

where

- rank (l_j^e, e) is the ranking of an item that is a duplicate of the item in questionable relevancy, and is located at the j^{th} position in the search engine e and $[1/\text{rank}\ (l_j^e, e)]$ is normalized;
- score(l_i^s) is the original score of the item in question l_i^s, which is located at the i^{th} position in search engine s, is initialized to be $[1 / \text{rank}(l_i^s)]$, and is normalized, where rank(l_i^s) is the rank of the item in question;
- score(l_i^s)" is the new score of l_i^s that is calculated by summing its original score and the scores of its duplicates each time a duplicate of l_i^s is found until all items in the retrieved results in the participating search engines are visited.

This scoring method is based on the rank of an item, and is a non-linear scoring method because it takes the reciprocal of the item's rank. The relationship between the score and the item's rank is non-linear, which thus gives an unfair scoring scheme for different rankings.

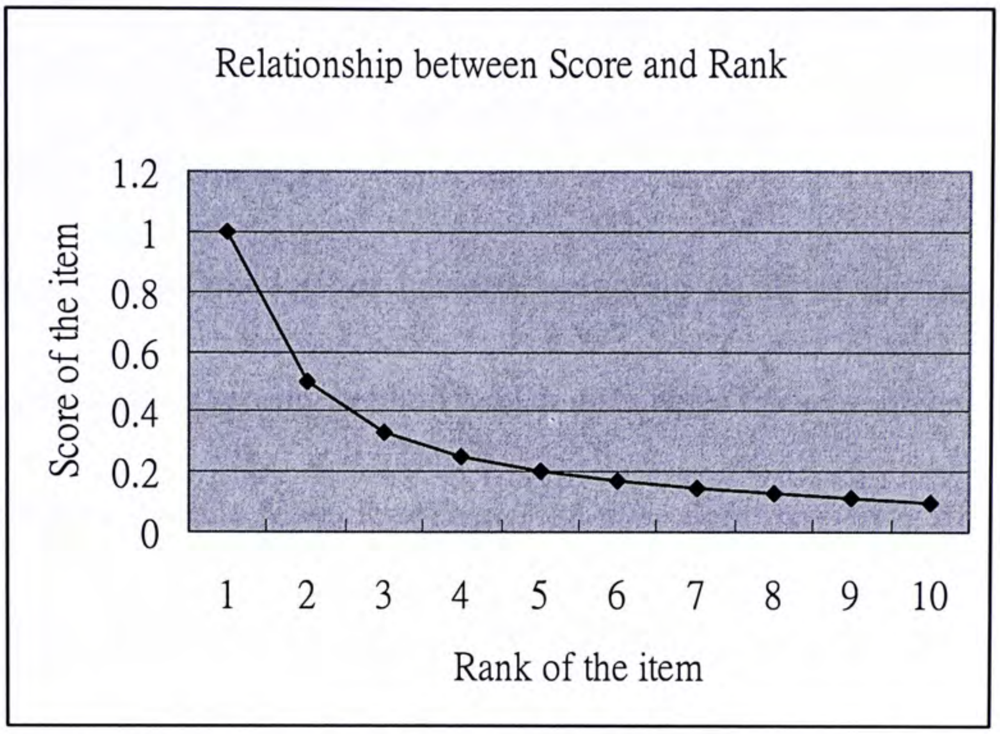


Figure 3.4 Relationship between Score and Rank for the Agreement 1 Re-ranking Method

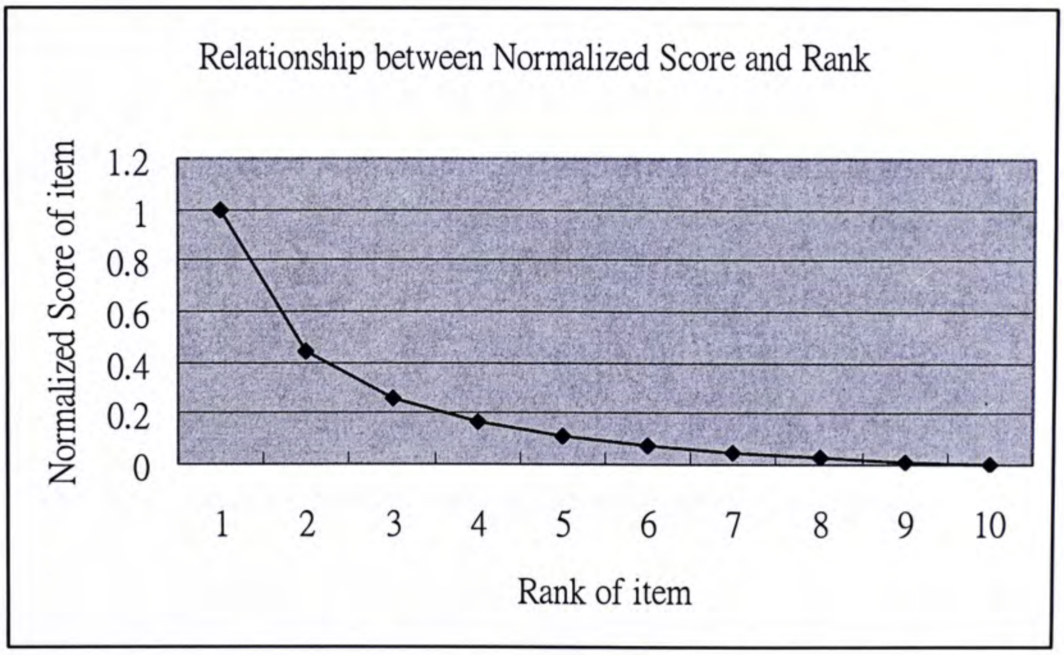


Figure 3.5 Relationship between Normalized Score and Rank for the Agreement 1 Re-ranking Method

It can be seen that both the curves with and without standard normalization become less steep when an item's rank is lower, which shows that the lower the rank of an item, the less the decrease in its score, which makes the scoring method non-linear.

Algorithm 2 (namely Agreement 2)

As the algorithm just mentioned gives unequal weighting to all of the ranks for their scores obtained, we suggest another scoring method, which distributes the scores evenly (as advocated by the Phase 2 *"Distribute"* of the *Normalize-Distribute-Sum* algorithm [69]). This is an Agreement re-ranking mechanism that does not consider other kinds of relationships (except duplicates), and uses a linear scoring method. We want to see whether giving a proportional score to an item according to its rank will cause much difference in a system's retrieval performance, compared to the algorithm that uses a non-linear scoring method.

This unmodified algorithm is named Agreement 2. The formula for this algorithm is as follows.

$$\text{score}(l_i{}^s)\text{"} = \text{score}(l_i{}^s) + \sum_{\textit{duplicate}} [1 - (\text{rank}\,(l_j{}^e, e) - 1) / N],$$

where

- $\text{rank}\,(l_j{}^e, e)$ is the ranking of the item that is a duplicate of the item in question and is located at the j^{th} position in search engine e, where $[1 - (\text{rank}\,(l_j{}^e, e) - 1) / N]$ is normalized;
- $\text{score}(l_i{}^s)$ is the original score of the item in question l_i^s that is located at the i^{th} position in search engine s, which is initialized to be $[1 - (\text{rank}\,(l_i{}^s, e) - 1) / N]$ and is normalized, where $\text{rank}(l_i{}^s)$ is the rank of the item in question;

- score(l_i^s)" is the new score of l_i^s that is calculated by summing its original score and the scores of its duplicates each time a duplicate of l_i^s is found until all items in the retrieved results in the participating search engines are visited;

- N is the total number of items retrieved by search engine s

This scoring method is based on the rank of an item, and is a linear scoring method because it gives mark based on the item's rank on a scale, and the relationship between the score and the item's rank is linear, which thus makes it a fair scoring scheme for different rankings.

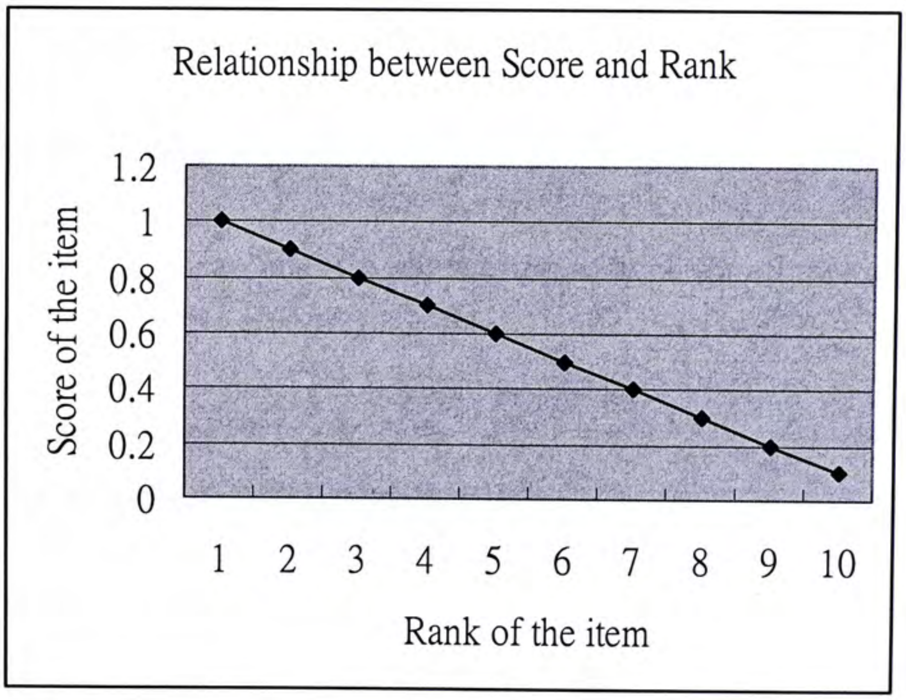


Figure 3.6 Relationship between Score and Rank for Agreement 2 Re-ranking Method

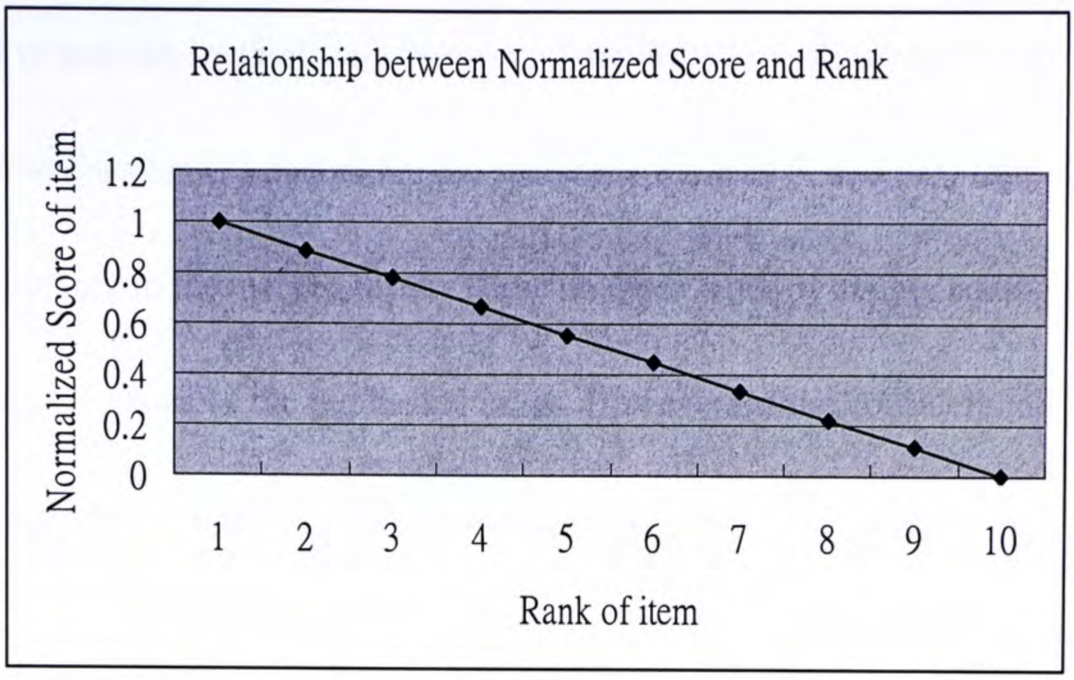


Figure 3.7 Relationship between Normalized Score and Rank for the Agreement 2 Re-ranking Method

It can be seen that both the curves with and without standard normalization are equally steep at each ranking point, which shows that regardless of the rank of an item, the decrease in score is even. This scoring method is thus linear, and is a fair scoring scheme for all rankings.

3.5.2 Modified re-ranking algorithm (after modification)

Algorithm 3 (namely Modified Agreement 1)

We now present one of the proposed algorithms, which is an Agreement re-ranking mechanism that considers the relationships between objects using a non-linear scoring method. That is, one of the components in the calculation of the score of an item is taken as the reciprocal of the item's rank. This algorithm is named Modified Agreement 1.

The unmodified re-ranking methods consider only "duplicate" relationships, or items that are the same as the item in question (F) that add to the relevancy score of F, and only they can give marks to F to boost its ranking in the merged result. These methods work by simply adding the scores that are contributed by the ranks of the duplicated items. However, as the consideration of other types of relationships between items is our main research focus, the scores that are contributed by all of the relationships in which the item in question has (the total relationship score) are taken into account when calculating the item's final score for ranking. This score has three components: ***i).*** *Rank-based relevancy score of the related item* - a related item that is ranked higher in its original search engine should have a positive effect on the relevance of the focused item, because the presence of related documents makes the document in question more important to the query, and if the related document is more relevant to the query, then the object in question has a greater chance of being more relevant to the query. Thus, the relevance of related documents should be counted when evaluating the relevance of a document in question to the query; ***ii). Strength of the relationship*** - as discussed in section 3.3, not all of the types of relationships have the same importance and strengths. For example, items in the same class have stronger relationships than items in the sibling class. Our algorithm needs to reflect this by assigning different weightings (refer to section 3.3.1) to different types of relationships and multiplying the weighting by the rank-based score of the related item; iii). Rank-based relevancy score of the item in question – an item in question is an item whose relevance to the query is determined. The algorithm should also assign higher scores to higher ranked items in question.

The total relationship score S_j for the item in question is obtained by using a "scoring model" that multiplies the relative weights, w_i for criterion i (relationship in our work), by the relevancy score of the corresponding related object, L_{ij}, and then adds them together. That is,

$$S_j = \sum_i w_i \, L_{ij}$$

Therefore, the total score of the item in question is the sum of the product of the relevancy scores of its related items and the corresponding weightings of the strengths of the relationships, plus the relevancy score of the item itself, which is the reciprocal of its rank. The score components are normalized each time. We use the sum of the weighted normalized scores as a document's total score, instead of the min, max, median, and average of the weighted normalized scores, because summing them produces the most effective retrieval system performance, as is shown in Section 2.2.1 in the discussion of the "Comb" algorithm. Hence, our proposed algorithms are based on the "CombSum" method, but differ in that an item's score is rank based.

Hence an item's total score is contributed by (i) the ranks of its related items in their originating search engines; (ii) the strengths of these relationships; (iii) the rank of the item in question in its originating search engine; and (iv) the number of items that are related to the item in question. The formula for this algorithm is shown as follows.

$$\text{score}(l_i^s)\text{''} \;\; = \text{score}(l_i^s) + \sum_{\mathit{relationships}} [1 \,/\, \text{rank}\,(\text{l}_j^e, \text{e})] * \mathit{weighting}\; (l_i^s,\, l_j^e)$$

where

- score(l_is) is the original score of the item in question l_i^s that is located at the ith position in search engine s, which is initialized to be [1 / rank(l_i^s)] and is normalized, where rank(l_i^s) is the rank of the item in question in search engine s;
- rank (l_j^e, e) is the rank of the jth item in search engine e, which is a related object of the item in question, and [1/rank (l_j^e, e)] is normalized;
- weighting (l_is, l_j^e) is the weighting of relationships between the jth item in search engine e that is related to the item in question; and
- score(l_is)" is the new score of l_i^s that is calculated by summing the original score and the total relationship scores until all of the related items in the retrieved results in the involved search engines are found

Algorithm 4 (namely Modified Agreement 2)

This proposed algorithm is an Agreement re-ranking mechanism that considers other types of relationships between objects, but uses a linear scoring method, that is, one of the components that calculates the rank-based score of an item is [1- (rank of the item-1) / N], such that the calculated value is proportional to the item's rank. This algorithm is named Modified Agreement 2.

This algorithm uses the same concept and procedure as Modified Agreement 1 in assigning a score to the item in question, but the score simulated from an item's rank is further distributed by the factor [1- (rank of item-1) / N], instead of simply using [1/rank of item]. The total score of the item in question is the sum of the product of the relevancy scores of the related items and the weighted score of the strengths of the corresponding relationship, plus the relevancy score of the item in question, which is [1- (rank of the item in question-1) / N]. Note that the score components that are calculated each time are normalized.

The formula for this algorithm is shown as follows.

$$\text{score}(l_i^s)\text{''} = \text{score}(l_i^s) + \sum_{\textit{relationships}} [1- (\text{rank}\ (l_j^e, e)-1) / N] * \textit{weighting}\ (l_i^s, l_j^e)$$

where

- $\text{score}(l_i^s)$ is the original score of the item in question l_i^s that is located at the i th position in search engine s, is initialized to be [1- (rank (l_i^s, s)-1) / N], and is normalized, where rank(l_i^s) is the rank of the item in question in search engine s;
- rank (l_j^e, e) is the rank of the j th item in search engine e, which is related to the item in question, where [1- (rank (l_j^e, e)-1) / N] is normalized;
- weighting (l_i^s, l_j^e) is the weighting of the relationships between the j th item in search engine e that is related to the item in question;
- $\text{score}(l_i^s)$" is the new score of l_i^s that is calculated by summing the original score and the total relationship scores until all of the related items in the retrieved results in the involved search engines are found;
- N is the total number of items retrieved by search engine s

From $\text{score}(l_i^s)\text{"} = \text{score}(l_i^s) + \sum_{\mathit{relationships}} [\ 1/\text{rank}\ (l_j^e, e)\ ] * \mathit{weighting}\ (l_i^s, l_j^e)$, we can see that the rank of the item in question l_i^s will be boosted if it has either many items that are related to it (higher value for $\sum_{\mathit{relationships}} \mathit{weighting}\ (l_i^s, l_j^e)$), a stronger relationship with other items (higher value for *weighting* (l_i^s, l_j^e)), related items that are important to the query (higher value for $1/\mathit{rank}\ (l_j^e, e)$), or a high relevance itself (higher value for $1/\mathit{rank}\ (l_i^s, s)$).

With all of the four algorithms, the system ranks the items based on their final scores as computed using the corresponding formula, and finally present them to users in descending order of their total score value. For re-ranking methods both that do and do not consider relationships other than duplicate relationships, the non-linear scoring scheme gives a non-linear increase or decrease in score, whereas the linear scoring scheme gives a linear increase or decrease. At this stage, we cannot tell which scoring scheme yields a better ranking performance as reflected by the retrieval effectiveness, and there is likely to be different retrieval effectiveness for scoring schemes that do and do not consider other kinds of relationships. Thus, an experiment needs to be conducted to find out which algorithm yields the best ranking performance generally. This is discussed thoroughly in the following chapter.

Chapter 4

Evaluation Methodology and Experimental Results

4.1 Objective

The experiment aims to find out whether the modified algorithms that take into account relationship factors improve the retrieval performance of a meta-search system. We aim to ascertain whether there is an increase in the percentage of relevant items retrieved out of the total number of items retrieved (precision), and an increase in the percentage of relevant items retrieved out of the total number of relevant items in the whole population set (recall) after applying the proposed algorithm compared to all of the search engines involved in the meta-searching and a system that only applies the unmodified Agreement merging mechanism. The proposed algorithm has two scoring schemes that calculate the relevancy score of the item based on its rank using either a non-linear or linear scoring scheme, and we also compare these two methods.

4.2 Experimental Design and Setup

4.2.1 Preparation of data

We now identify the data that are to be collected.

1. Source

All of the results for our meta-search were collected from four search engines that contain four types of media: web, image, audio, and video. The four underlying search engines are AltaVista {http://www.altavista.com}, AlltheWeb {http://www.alltheweb.com}, Lycos {http://www.lycos.com}, and Google {http://www.google.com}. But Google is used instead of Lycos for the meta-search of image items, and the other three are used for retrieving web, audio, and video objects, which means that the search for objects of each media format involves three search engines in the meta-search. These search engines were chosen as they give complete coverage of multimedia files and are commonly used engines. They also provide fresh, high-quality and relevant results by aggregating information into highly segmented indexes, which helps users to refine their searches and quickly access the most pertinent and useful information. AlltheWeb even combines one of the largest and freshest indices with the most powerful search features that allow users to find items faster than any other search engine. Most importantly, these search engines retrieve many documents that are inter-related and relevant to queries, especially those that are duplicated or in the same class, also retrieve distinct and unrelated irrelevant documents. Thus, there is scope for ameliorating the meta-search retrieval performance by applying our proposed merging algorithms to the merge results from these search engines.

2. Number of results captured

For each query, the first 100 results are captured for each medium from each underlying search engine. After re-ranking, there are 300 results for each medium, and these results are treated as the large pool of search results to be used to evaluate the recall. The first 30 results are to be retrieved and presented to users for evaluation of the precision of the search, as it is assumed that users only have an interest in or obtain the information that they require from the top 30 results. The 300 results in the three underlying search engines are used for the human evaluation of their relevancies. If the participating search engines return less than 100 results, then we simply retrieve all of the results, and similarly if less than 30 results can be presented to the user, then we present all of the results.

We recruited a candidate (user) to evaluate the system following the guidelines that are given in section 4.3.1. After collecting the targeted pool of data, the algorithms are applied to merge the data set to generate the results for evaluation. At this stage the parameters are determined as the weightings for all kinds of relationships, as defined in section 3.3.1. We designed evaluation programs to perform the experiment to apply the data merging mechanisms to the data collected to yield the results and present them to users.

Each item in the 300 results retrieved by the three underlying search engines is assigned a score by the system using a certain re-ranking method. The items are ordered by these scores in descending order, and only the first 30 results are presented to users. The order of the results differs according to the merge method that is used.

We compare the performance of AltaVista, AlltheWeb, Lycos (or Google for items in image format), system that uses the Agreement 1 re-ranking algorithm, system that uses the Agreement 2 re-ranking algorithm, system that uses the modified Agreement 1 re-ranking algorithm, and system that uses the modified Agreement 2 re-ranking algorithm, and employ user judgment of the relevance of the results to verify whether our proposed methodology improves the meta-search performance.

The system design is the same as that shown in Figure 1.1, but the re-ranking method is different. Note that we do not develop systems to implement the first three mechanisms, as AltaVista, AlltheWeb, Lycos, and Google are the participating search engines and already have their results ordered.

4.3 Evaluation Methodology

4.3.1 Evaluation of the relevance of a document to the corresponding query

The relevance of the results is determined by human judgment. Relevance is a measure of the contact between a source and a destination, that is, between a document and its user. As introduced in Section 2.4, we use explicit rating as the relevance judgment to evaluate the relevance, because explicit rating overcomes the problems of implicit rating, such as first-time users (user which is too late to discover the page or even not yet visited the page before), or history records due to multiple users (a page may not be referenced by the same user every time it is reached), or the article being so new that not many users have visited it.

In the explicit rating system, which asks users to evaluate the relevance of articles directly, each user is required to give their feedback on each result on scale of 0-100, where 50 is the neutral point. Users are also asked to rank the articles based on the order of relevance, where 1 is the most relevant. Results that obtain higher marks are more relevant from the user's point of view, and results that obtain zero marks are absolutely irrelevant to the query.

4.3.2 Performance Measures of the Evaluation

There are several performance measures [71] that reflect the effectiveness of a search engine, and we employ recall and precision to measure the effectiveness of a search system. We also use the F-measure, which is a widely used single measure that helps to strike a balance between recall and precision in our evaluation.

Recall, R

This represents the ability of the system to present all of the relevant items. It is a measure of whether or not a particular item is retrieved or the extent to which the retrieval of desirable items occurs.

R is the percentage of relevant items that are retrieved

$$\mathit{Recall} = \frac{\mathit{Number\ of\ relevant\ items\ retrieved}}{\mathit{Total\ number\ of\ relevant\ items\ in\ the\ collection}}\ x\ 100\%$$

Precision, P

This is the ability of the system is to present only items that are relevant while avoiding the retrieval of irrelevant items. This measure locates the extent to which the system is able to withhold unwanted items in a given situation.

P is the percentage of retrieved items that are relevant

$$Precision = \frac{Number\ of\ relevant\ items\ retrieved}{Total\ number\ of\ items\ retrieved}\ x\ 100\%$$

Recall thus relates to the ability of the system to retrieve relevant documents, and precision relates to the ability to avoid retrieving irrelevant documents.

F-measure, F

This is a combination of precision and recall. The higher the F value, the better the performance of the system.

$$F\text{-}measure = \frac{recall * precision}{(recall + precision)/2}$$

4.4 Experimental Results and Interpretation

We used the keywords "Keanu Reeves" as the query to retrieve audio items, and after user judgment on the relevance of the items, the evaluation statistics are as follows.

4.4.1 Precision

Algorithm used in system retrieving audio items on "Keanu Reeves"	Precision
AltaVista	0.8
AlltheWeb	0.9
Lycos	0.533333
Agreement 1	0.766667
Agreement 2	0.9
Modified Agreement 1	0.933333
Modified Agreement 2	1

Table 4.1 Precision of the systems in retrieving audio items on "Keanu Reeves"

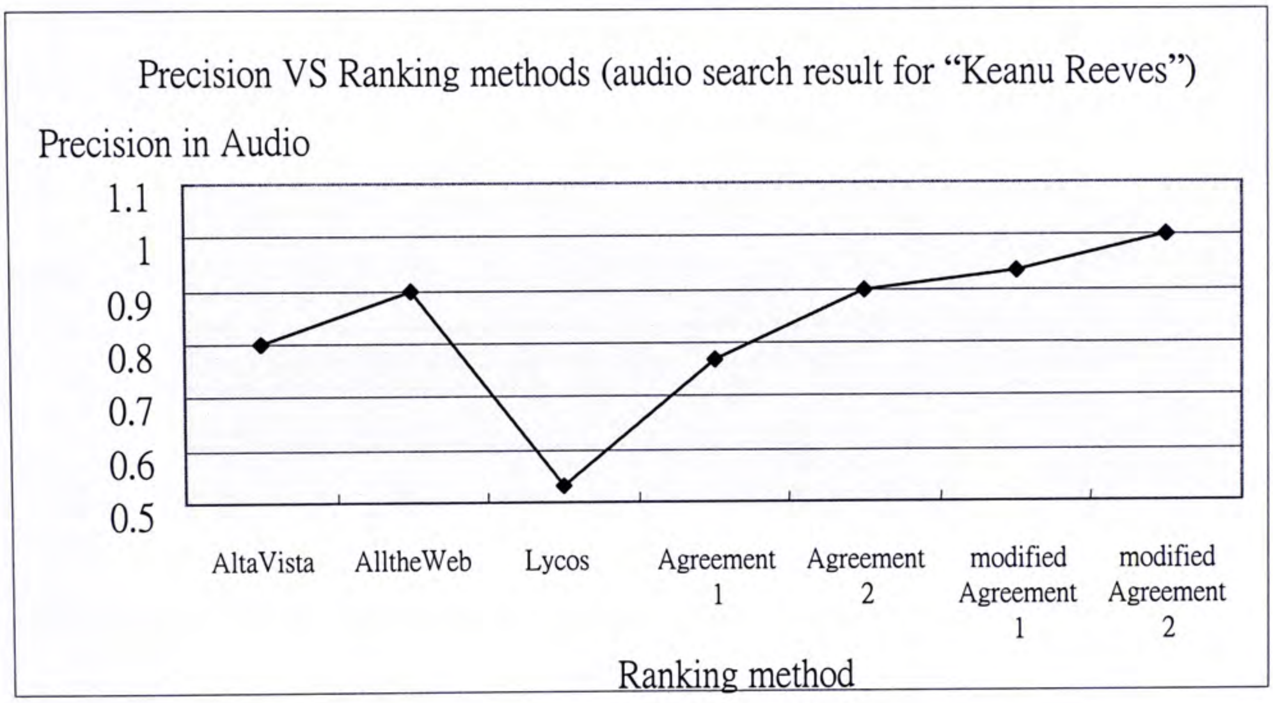


Figure 4.1 Precision vs ranking methods of the systems in retrieving audio items on "Keanu Reeves"

4.4.2 Recall

Algorithm used in system retrieving audio items on "Keanu Reeves"	Recall
AltaVista	0.183206
AlltheWeb	0.206107
Lycos	0.122137
Agreement 1	0.175573
Agreement 2	0.206107
Modified Agreement 1	0.21374
Modified Agreement 2	0.229008

Table 4.2 Recall of the systems in retrieving audio items on "Keanu Reeves"

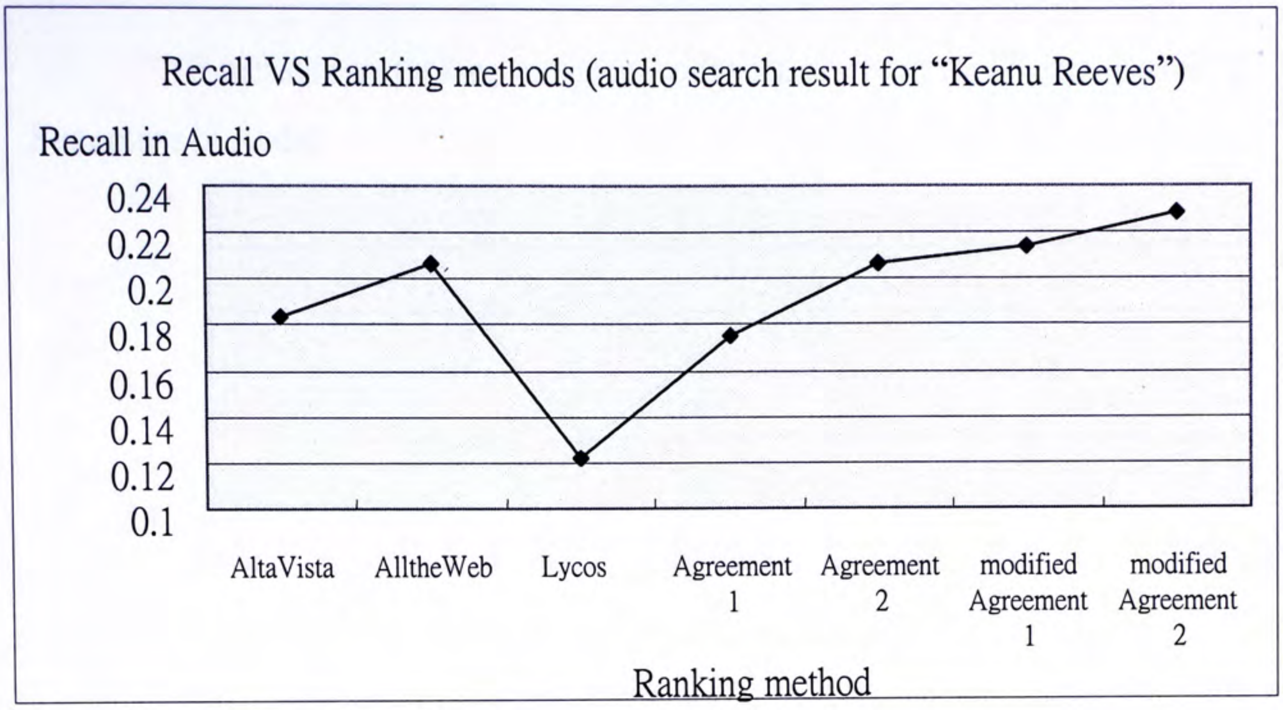


Figure 4.2 Recall vs ranking methods of the systems in retrieving audio items on "Keanu Reeves"

4.4.3. F-measure

Algorithm used in system in retrieving audio items on "Keanu Reeves"	F-measure
AltaVista	0.298137
AlltheWeb	0.335404
Lycos	0.198758
Agreement 1	0.285714
Agreement 2	0.335404
Modified Agreement 1	0.347826
Modified Agreement 2	0.372671

Table 4.3 F-measure of the systems in retrieving audio items on "Keanu Reeves"

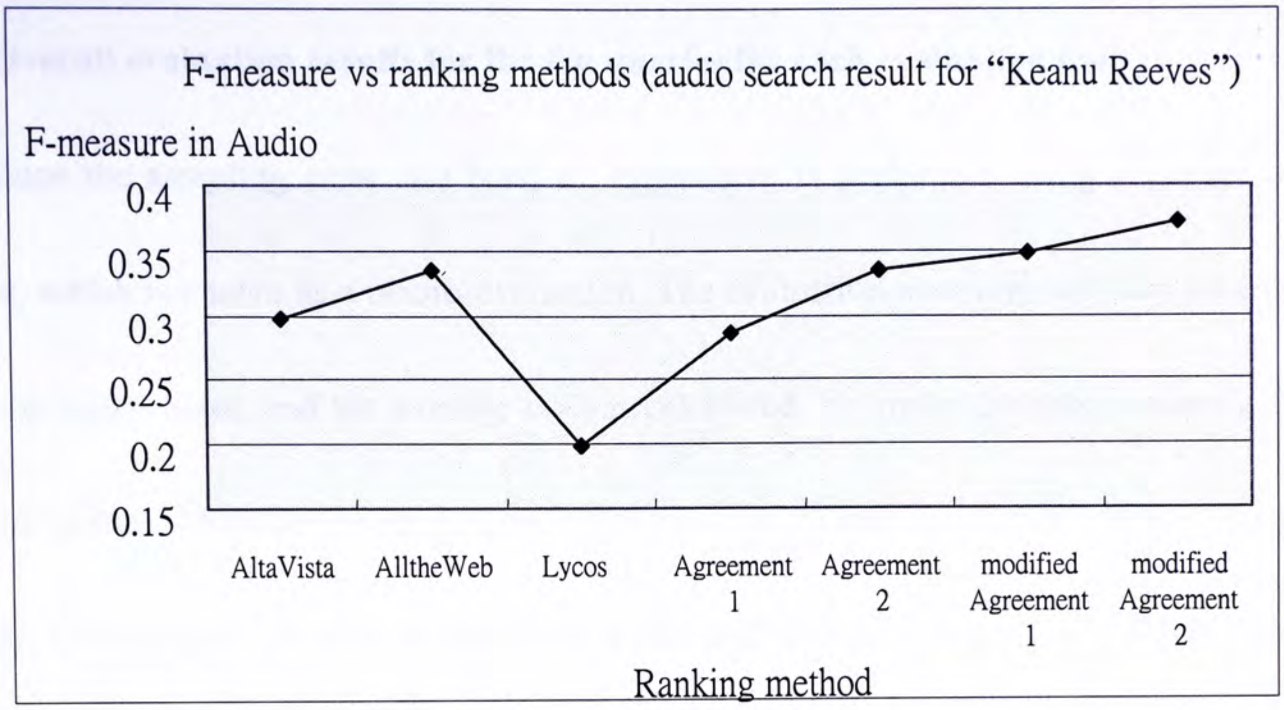


Figure 4.3 F-measure vs ranking methods of systems in retrieving audio items on “Keanu Reeves”

We can see that the proposed algorithm of Modified Agreement 2 yields the best performance as measured by precision, recall, and F-measure. Modified Agreement 1 also outperforms the other merging mechanisms, which shows that our proposed algorithms (modified Agreement 1 and modified Agreement 2) both improve a system's retrieval performance. Agreement 1 [8] has a lower precision, recall, and F-measure than the AltaVista and AlltheWeb engines. This is because some of the highly ranked items have duplicates in other search engines, but are not relevant to the query, and these irrelevant items obtain high scores and ranks after merging using Agreement 1. This problem is mentioned in chapter 3, and is eliminated in our modified Agreement 1 and modified Agreement 2 algorithms, as they do not consider only “duplicate” relationships but also other kinds of relationships.

4.4.4 Overall evaluation results for the ten queries for each evaluation tool

To reduce the sampling error and bias, an experiment is performed using a greater number of queries, which is known as a macro-evaluation. The evaluation measures are then ascertained on a query-by-query basis, and the average is then calculated. To make the results more accurate, we used 10 queries: "Moulin Rouge," "Lord of the Rings," "Orlando Bloom," "Faye Wong," "Janet Jackson," "Liv Tyler," "Norah Jones," "Ricky Martin," "Keanu Reeves," and "David Beckham."

The following tables show the average precision, recall, and F-measure for web, image, audio, and video items for the ten queries of systems using different merging methods.

Average Precision

Web

Precision (Web)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.8	0.833333	0.866667	0.933333	0.933333	0.633333	0.966667	0.9	0.9	0.866667	0.863333
AlltheWeb	0.833333	0.866667	0.833333	0.833333	0.766667	0.666667	0.966667	0.9	0.866667	0.833333	0.836667
Lycos	0.733333	0.766667	0.9	0.833333	0.866667	0.6	0.966667	0.933333	0.9	0.9	0.84
Agreement 1	0.733333	0.9	0.866667	0.933333	0.833333	0.766667	0.966667	0.933333	0.9	0.9	0.873333
Agreement 2	0.866667	0.9	0.8	0.933333	0.933333	0.733333	0.966667	0.933333	0.933333	0.9	0.89
Modified Agreement 1	0.9	0.866667	0.9	0.933333	0.9	0.866667	0.933333	0.9	0.933333	0.933333	0.906667
Modified Agreement 2	0.866667	0.866667	0.933333	0.933333	0.9	0.9	0.966667	0.966667	1	1	0.933333

Table 4.4 Average precision for systems in retrieving Web page items

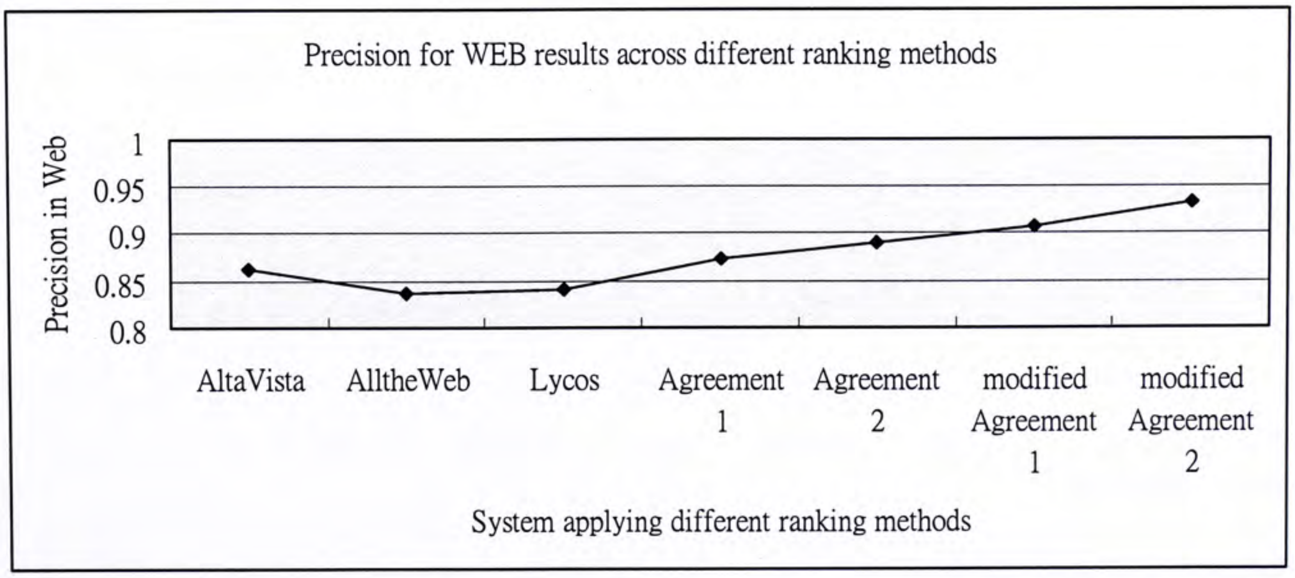


Figure 4.4 Average precision vs ranking methods for systems in retrieving Web page items

Image

Precision (Image)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.6	1	0.966667	0.7	0.533333	0.8	0.933333	0.733333	0.8	0.933333	0.8
AlltheWeb	0.6	1	0.966667	0.833333	0.5	0.733333	0.933333	0.833333	0.833333	0.933333	0.816667
Google	0.4	0.633333	0.833333	0.9	0.466667	0.833333	0.866667	0.466667	0.833333	0.8	0.703333
Agreement 1	0.6	0.966667	0.9	0.9	0.433333	0.9	0.866667	0.833333	0.866667	0.866667	0.813333
Agreement 2	0.7	0.966667	0.833333	0.9	0.7	0.9	0.933333	0.833333	0.9	0.9	0.856667
Modified Agreement 1	0.9	0.966667	0.966667	0.933333	0.966667	1	0.966667	1	0.966667	0.933333	0.96
Modified Agreement 2	0.933333	1	1	1	0.9	0.933333	1	1	1	1	0.976667

Table 4.5 Average precision for systems in retrieving image items

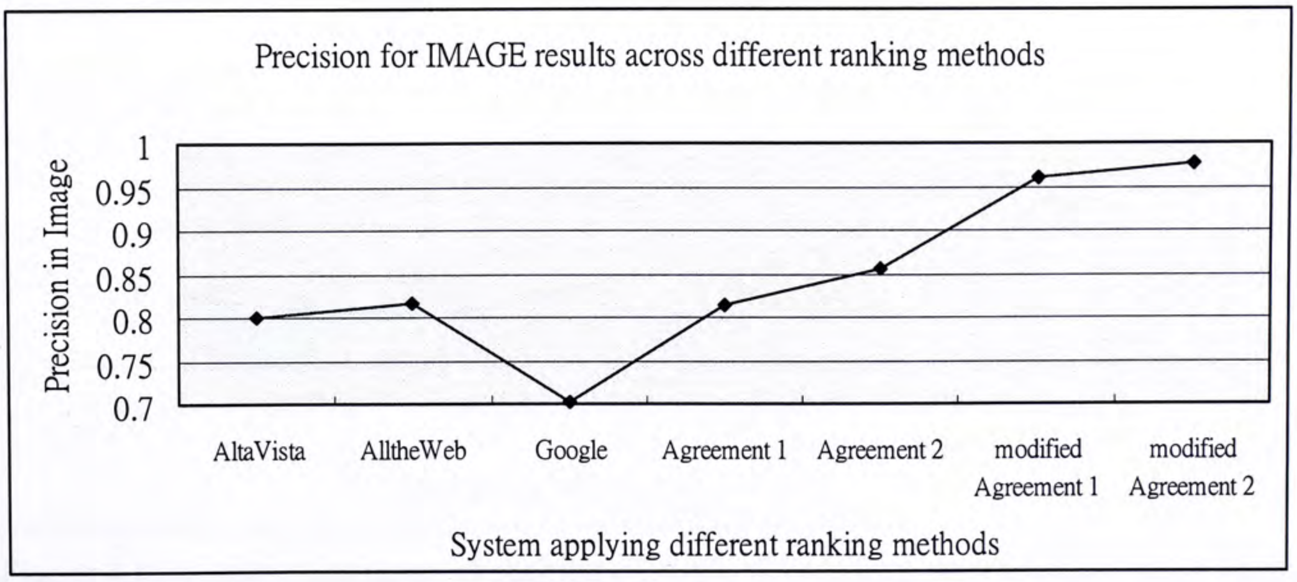


Figure 4.5 Average precision vs ranking methods for systems in retrieving image items

Audio

Precision (Audio)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.56666666	0.966667	0.866667	0.9	0.233333	0.866667	0.875	0.8	0.9	0.833333	0.780833
AlltheWeb	0.53333333	0.866667	0.833333	0.9	0.133333	0.866667	0.875	0.9	0.966667	0.714286	0.758929
Lycos	0.56666666	0.866667	0.933333	0.9	0.366667	0.8	0.866667	0.533333	0.9	0.866667	0.76
Agreement 1	0.6	0.933333	0.933333	0.9	0.333333	0.8	0.933333	0.766667	0.933333	0.833333	0.796667
Agreement 2	0.53333333	0.933333	0.933333	0.933333	0.3	0.866667	0.9	0.9	0.9	0.866667	0.806667
Modified Agreement 1	0.83333333	1	1	0.966667	0.466667	0.966667	0.966667	0.933333	0.9	0.866667	0.89
Modified Agreement 2	0.76666666	1	1	0.966667	0.566667	0.966667	0.966667	1	0.933333	0.866667	0.903333

Table 4.6 Average precision for systems in retrieving audio items

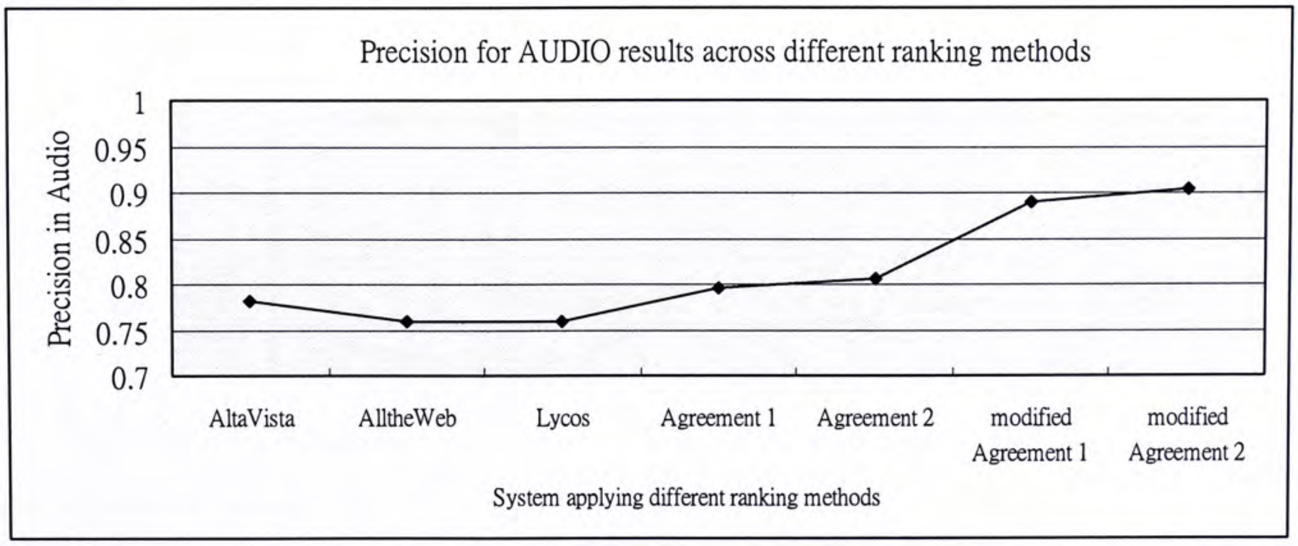


Figure 4.6 Average precision vs ranking methods for systems in retrieving audio items

Video

Precision (Video)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.933333	0.966667	0.733333	1	0.133333	0.777778	0.827586	0.433333	0.733333	0.833333	0.737203
AlltheWeb	0.933333	0.933333	0.666667	1	0.133333	0.666667	0.892857	0.333333	0.733333	0.833333	0.712619
Lycos	0.933333	0.933333	0.666667	1	0.266667	0.666667	0.9	0.366667	0.866667	0.9	0.75
Agreement 1	0.933333	0.933333	0.8	1	0.066667	0.666667	0.866667	0.433333	0.833333	0.966667	0.75
Agreement 2	0.933333	0.966667	0.8	1	0.133333	0.666667	0.866667	0.466667	0.866667	0.966667	0.766667
Modified Agreement 1	0.9	0.966667	0.866667	1	0.266667	0.888889	0.966667	0.5	0.9	0.966667	0.822222
Modified Agreement 2	0.866667	1	0.866667	1	0.333333	1	0.966667	0.433333	0.933333	0.966667	0.836667

Table 4.7 Average precision for systems in retrieving video items

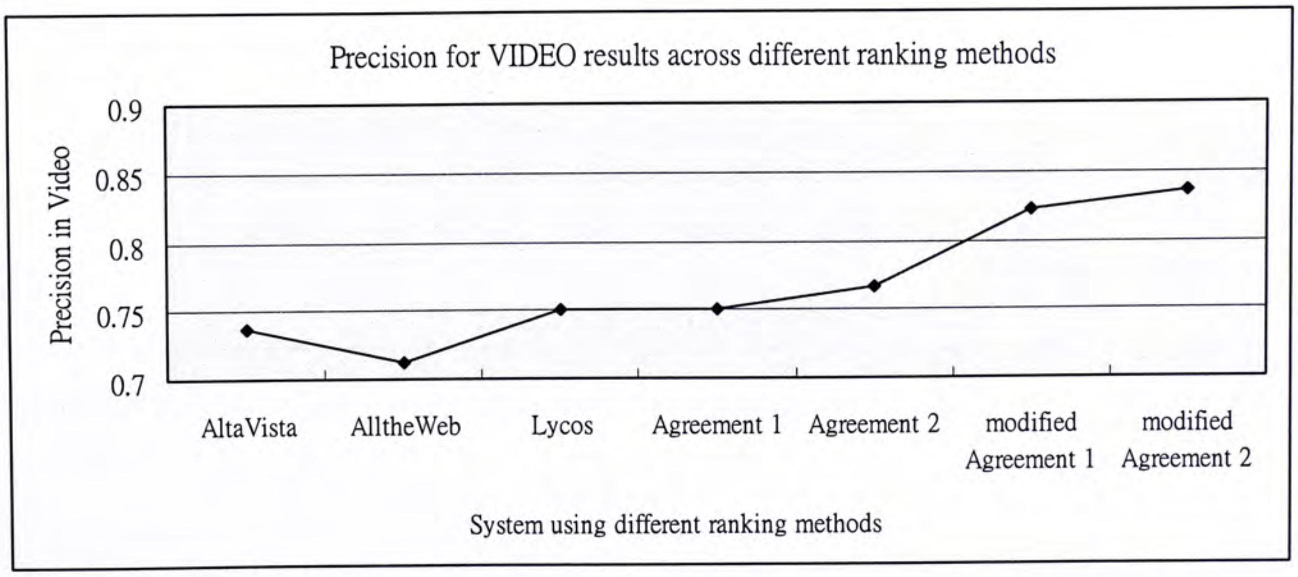


Figure 4.7 Average precision vs ranking methods for systems in retrieving video items

Average Recall

Web

Recall (Web)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.121212	0.113636	0.103586	0.125	0.127854	0.114458	0.109434	0.108871	0.102662	0.114537	0.114125
AlltheWeb	0.126263	0.118182	0.099602	0.111607	0.105023	0.120482	0.109434	0.108871	0.098859	0.110132	0.110845
Lycos	0.111111	0.104545	0.10757	0.111607	0.118721	0.108434	0.109434	0.112903	0.102662	0.118943	0.110593
Agreement 1	0.111111	0.122727	0.103586	0.125	0.114155	0.138554	0.109434	0.112903	0.102662	0.118943	0.115908
Agreement 2	0.131313	0.122727	0.095618	0.125	0.127854	0.13253	0.109434	0.112903	0.106464	0.118943	0.118279
Modified Agreement 1	0.136364	0.118182	0.10757	0.125	0.123288	0.156627	0.10566	0.108871	0.106464	0.123348	0.121137
Modified Agreement 2	0.131313	0.118182	0.111554	0.125	0.123288	0.162651	0.109434	0.116935	0.114068	0.132159	0.124458

Table 4.8 Average recall for systems in retrieving Web page items

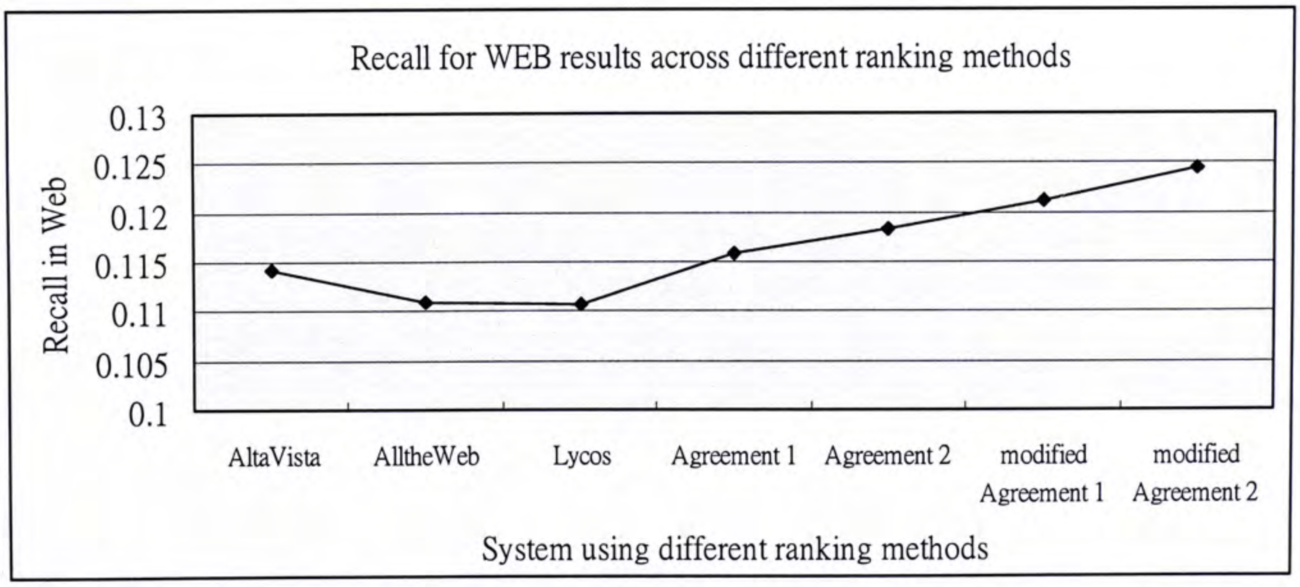


Figure 4.8 Average recall vs ranking methods for systems in retrieving Web page items

Image

Recall (Image)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.163636	0.126582	0.109434	0.099526	0.129032	0.152866	0.108108	0.127907	0.089888	0.122807	0.122979
AlltheWeb	0.163636	0.126582	0.109434	0.118483	0.120968	0.140127	0.108108	0.145349	0.093633	0.122807	0.124913
Google	0.109091	0.080169	0.09434	0.127962	0.112903	0.159236	0.100386	0.081395	0.093633	0.105263	0.106438
Agreement 1	0.163636	0.122363	0.101887	0.127962	0.104839	0.171975	0.100386	0.145349	0.097378	0.114035	0.124981
Agreement 2	0.190909	0.122363	0.09434	0.127962	0.169355	0.171975	0.108108	0.145349	0.101124	0.118421	0.13499
Modified Agreement 1	0.245455	0.122363	0.109434	0.132701	0.233871	0.191083	0.111969	0.174419	0.108614	0.122807	0.155272
Modified Agreement 2	0.254545	0.126582	0.113208	0.14218	0.217742	0.178344	0.11583	0.174419	0.11236	0.131579	0.156679

Table 4.9 Average recall for systems in retrieving image items

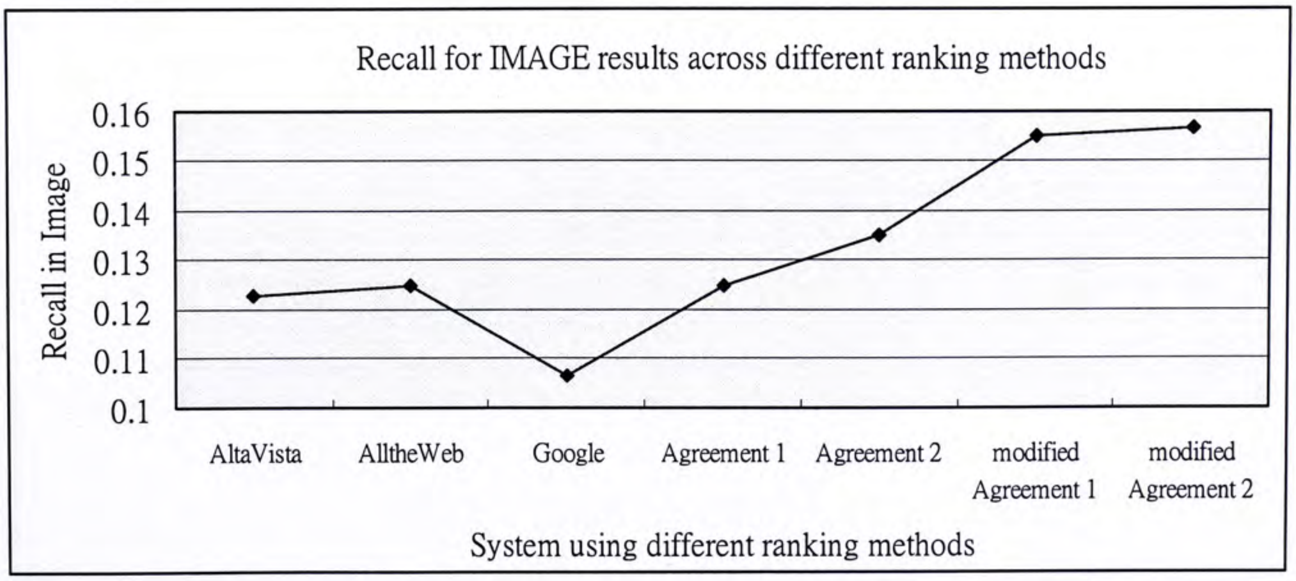


Figure 4.9 Average recall vs ranking methods for systems in retrieving image items

Audio

Recall (Audio)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.163636	0.126582	0.109434	0.099526	0.129032	0.152866	0.108108	0.127907	0.089888	0.122807	0.122979
AlltheWeb	0.163636	0.126582	0.109434	0.118483	0.120968	0.140127	0.108108	0.145349	0.093633	0.122807	0.124913
Lycos	0.109091	0.080169	0.09434	0.127962	0.112903	0.159236	0.100386	0.081395	0.093633	0.105263	0.106438
Agreement 1	0.163636	0.122363	0.101887	0.127962	0.104839	0.171975	0.100386	0.145349	0.097378	0.114035	0.124981
Agreement 2	0.190909	0.122363	0.09434	0.127962	0.169355	0.171975	0.108108	0.145349	0.101124	0.118421	0.13499
Modified Agreement 1	0.245455	0.122363	0.109434	0.132701	0.233871	0.191083	0.111969	0.174419	0.108614	0.122807	0.155272
Modified Agreement 2	0.254545	0.126582	0.113208	0.14218	0.217742	0.178344	0.11583	0.174419	0.11236	0.131579	0.156679

Table 4.10 Average recall for systems in retrieving audio items

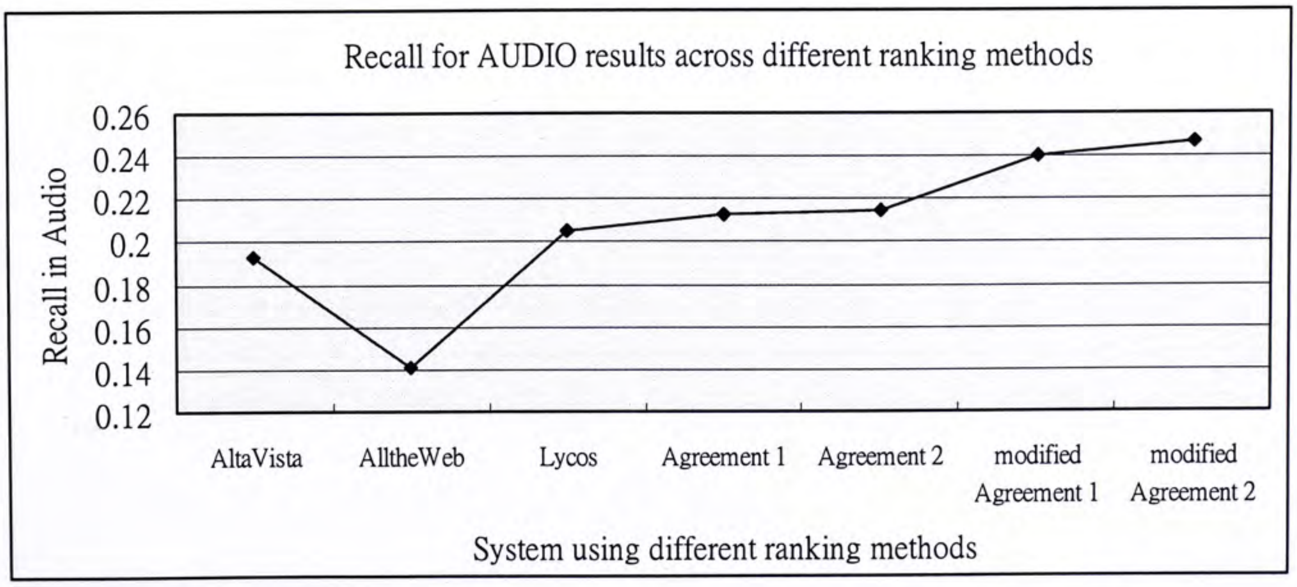


Figure 4.10 Average recall vs ranking methods for system in retrieving audio items

Video

Recall (Video)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.256881	0.118367	0.177419	0.25	0.086957	0.411765	0.292683	0.213115	0.192982	0.221239	0.222141
AlltheWeb	0.256881	0.114286	0.16129	0.25	0.086957	0.352941	0.304878	0.163934	0.192982	0.221239	0.210539
Lycos	0.229358	0.114286	0.16129	0.5	0.173913	0.352941	0.329268	0.2	0.22807	0.238938	0.252806
Agreement 1	0.256881	0.114286	0.193548	0.5	0.043478	0.352941	0.317073	0.213115	0.219298	0.256637	0.246726
Agreement 2	0.256881	0.118367	0.193548	0.5	0.086957	0.352941	0.317073	0.229508	0.22807	0.256637	0.253998
Modified Agreement 1	0.247706	0.118367	0.209677	0.5	0.173913	0.470588	0.353659	0.245902	0.236842	0.256637	0.281329
Modified Agreement 2	0.238532	0.122449	0.209677	0.5	0.217391	0.529412	0.353659	0.213115	0.245614	0.256637	0.288649

Table 4.11 Average recall for systems in retrieving video items

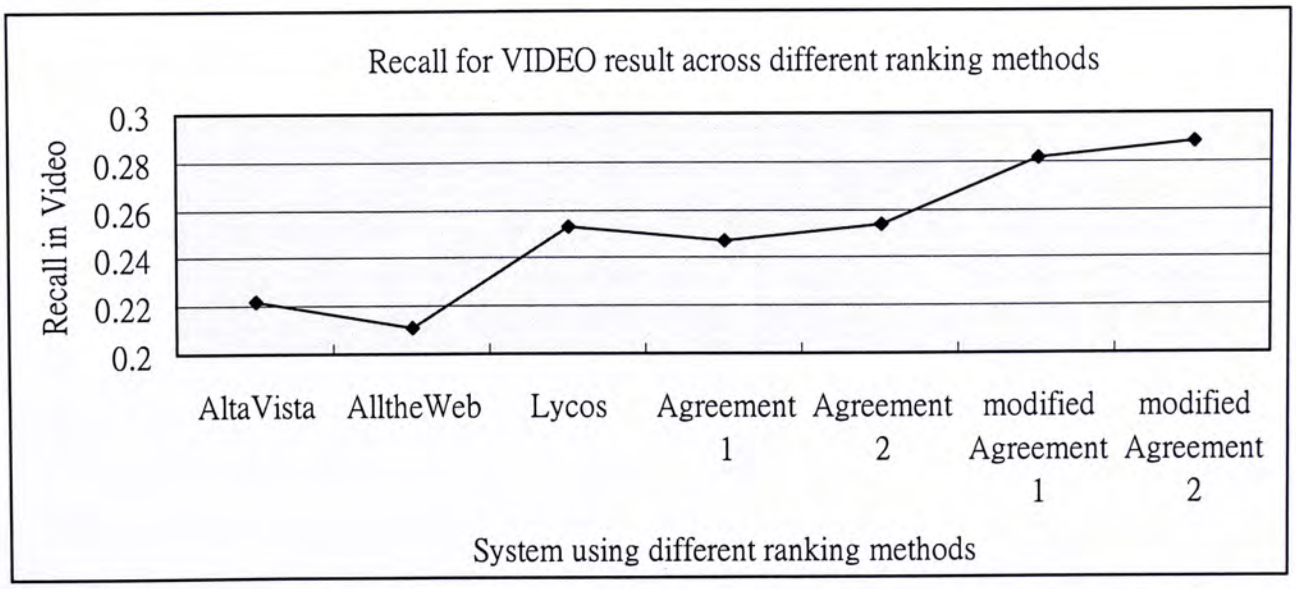


Figure 4.11 Average recall vs ranking methods for systems in retrieving video items

Average F-measure

Web

F-measure (Web)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.210526	0.2	0.185053	0.220472	0.2249	0.193878	0.19661	0.194245	0.1843	0.202335	0.201231
AlltheWeb	0.219298	0.208	0.177936	0.19685	0.184739	0.204082	0.19661	0.194245	0.177474	0.194553	0.195378
Lycos	0.192982	0.184	0.192171	0.19685	0.208835	0.183673	0.19661	0.201439	0.1843	0.210117	0.195097
Agreement 1	0.192982	0.216	0.185053	0.220472	0.200803	0.234694	0.19661	0.201439	0.1843	0.210117	0.204247
Agreement 2	0.228069	0.216	0.170819	0.220472	0.2249	0.22449	0.19661	0.201439	0.191126	0.210117	0.208404
Modified Agreement 1	0.236842	0.208	0.192171	0.220472	0.216867	0.265306	0.189831	0.194245	0.191126	0.217899	0.213276
Modified Agreement 2	0.228069	0.208	0.199288	0.220472	0.216867	0.27551	0.19661	0.208633	0.204778	0.233463	0.219169

Table 4.12 Average F-measure for systems in retrieving Web page items

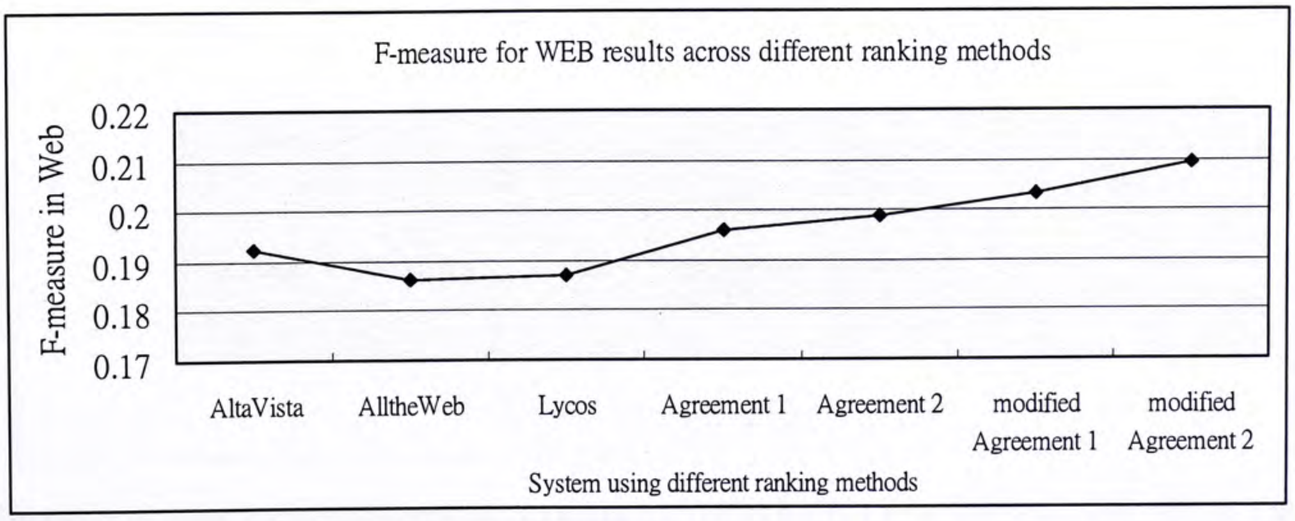


Figure 4.12 Average F-measure vs ranking methods for systems in retrieving Web page items

Image

F-measure (Image)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.257143	0.224719	0.19661	0.174274	0.207792	0.256684	0.193772	0.217822	0.161616	0.217054	0.210749
AlltheWeb	0.257143	0.224719	0.19661	0.207469	0.194805	0.235294	0.193772	0.247525	0.16835	0.217054	0.214274
Google	0.171429	0.142322	0.169492	0.224066	0.181818	0.26738	0.179931	0.138614	0.16835	0.186047	0.182945
Agreement 1	0.257143	0.217228	0.183051	0.224066	0.168831	0.28877	0.179931	0.247525	0.175084	0.209302	0.215093
Agreement 2	0.3	0.217228	0.169492	0.224066	0.272727	0.28877	0.193772	0.247525	0.181818	0.217054	0.231245
Modified Agreement 1	0.385714	0.217228	0.19661	0.232365	0.376623	0.320856	0.200692	0.29703	0.195286	0.224806	0.264721
Modified Agreement 2	0.4	0.224719	0.20339	0.248963	0.350649	0.299465	0.207612	0.29703	0.20202	0.232558	0.266641

Table 4.13 Average F-measure for systems in retrieving image items

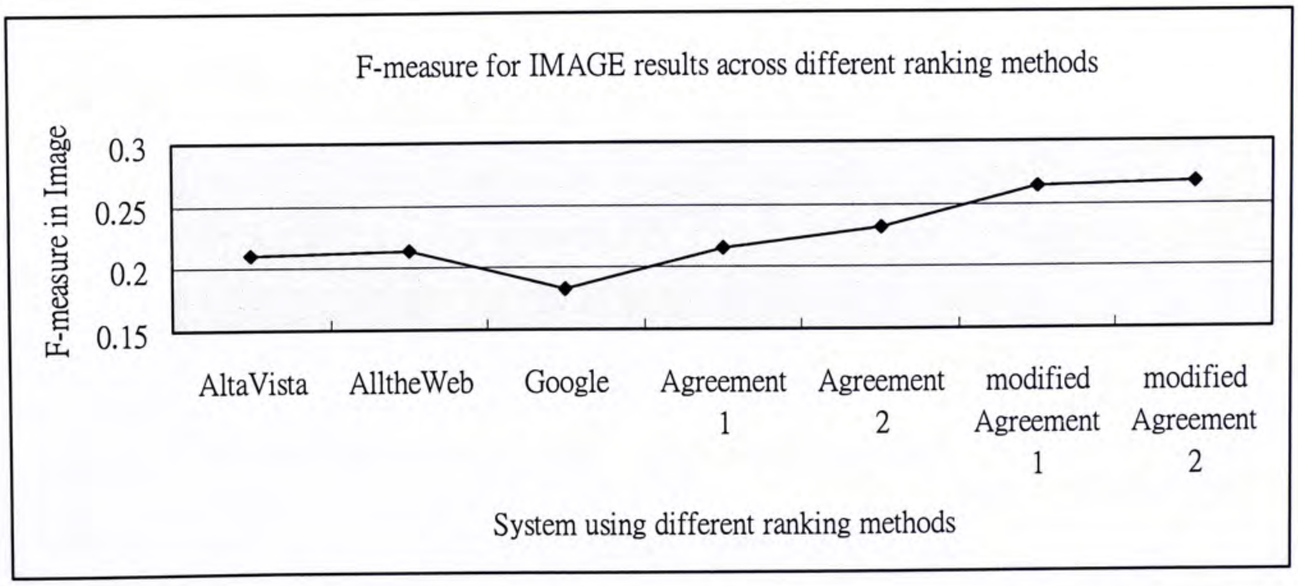


Figure 4.13 Average F-measure vs ranking methods for systems in retrieving image items

Audio

F-measure (Audio)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.237762236	0.219697	0.22807	0.3375	0.179487	0.248804	0.538462	0.298137	0.187125	0.49505	0.297009
AlltheWeb	0.237762236	0.19697	0.219298	0.3375	0.102564	0.248804	0.225806	0.335404	0.207885	0.235294	0.234729
Lycos	0.150442477	0.19697	0.245614	0.3375	0.282051	0.229665	0.619048	0.198758	0.193548	0.514851	0.296845
Agreement 1	0.251748251	0.212121	0.245614	0.3375	0.25641	0.229665	0.666667	0.285714	0.200717	0.49505	0.318121
Agreement 2	0.223776223	0.212121	0.245614	0.35	0.230769	0.248804	0.657963	0.335404	0.193548	0.514851	0.321285
Modified Agreement 1	0.349650349	0.227273	0.263158	0.3625	0.358974	0.277512	0.690476	0.347826	0.193548	0.514851	0.358577
Modified Agreement 2	0.321678321	0.227273	0.263158	0.3625	0.435897	0.277512	0.690476	0.372671	0.200717	0.514851	0.366673

Table 4.14 Average F-measure for systems in retrieving audio items

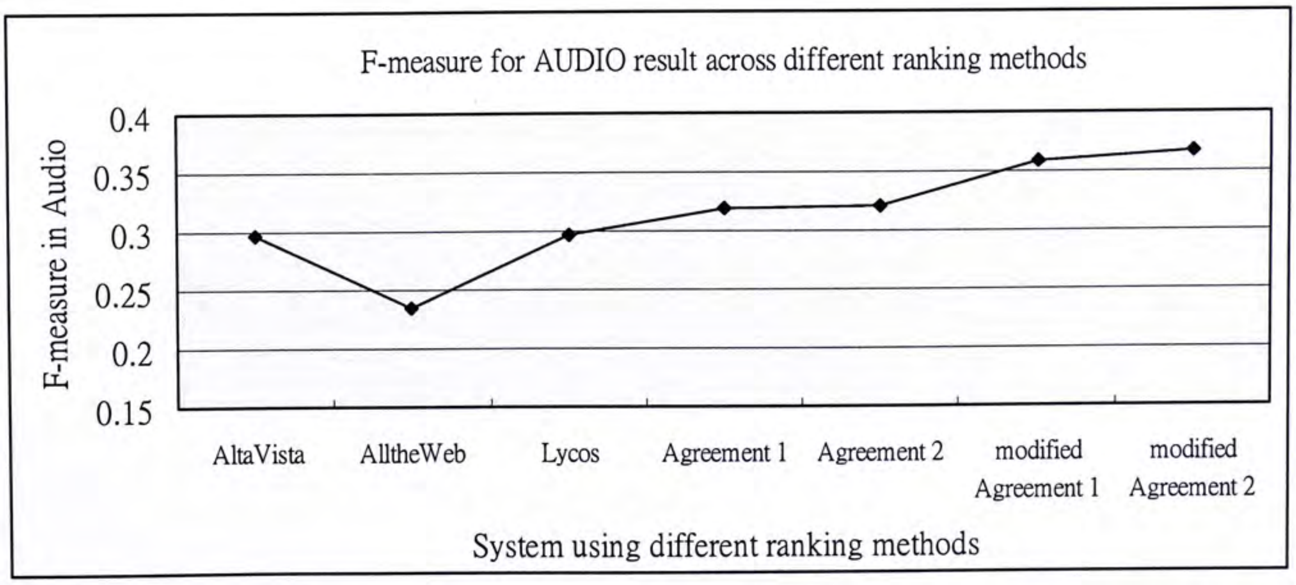


Figure 4.14 Average F-measure vs ranking methods for systems in retrieving audio items

Video

F-measure (Video)	Moulin Rouge	Lord of the Rings	Orlando Bloom	Faye Wong	Janet Jackson	Norah Jones	Liv Tyler	Keanu Reeves	Ricky Martin	David Beckham	AVERAGE
AltaVista	0.402878	0.210909	0.285714	0.4	0.105263	0.538462	0.432432	0.285714	0.305556	0.34965	0.331658
AlltheWeb	0.402878	0.203636	0.25974	0.4	0.105263	0.461538	0.454545	0.235294	0.305556	0.34965	0.31781
Lycos	0.359712	0.203636	0.25974	0.666667	0.210526	0.461538	0.482143	0.258824	0.361111	0.377622	0.364152
Agreement 1	0.402878	0.203636	0.311688	0.666667	0.052632	0.461538	0.464286	0.285714	0.347222	0.405594	0.360186
Agreement 2	0.402878	0.210909	0.311688	0.666667	0.105263	0.461538	0.464286	0.307692	0.361111	0.405594	0.369763
Modified Agreement 1	0.388489	0.210909	0.337662	0.666667	0.210526	0.615385	0.517857	0.32967	0.375	0.405594	0.405776
Modified Agreement 2	0.374101	0.218182	0.337662	0.666667	0.263158	0.692308	0.517857	0.285714	0.388889	0.405594	0.415013

Table 4.15 Average F-measure for systems in retrieving video items

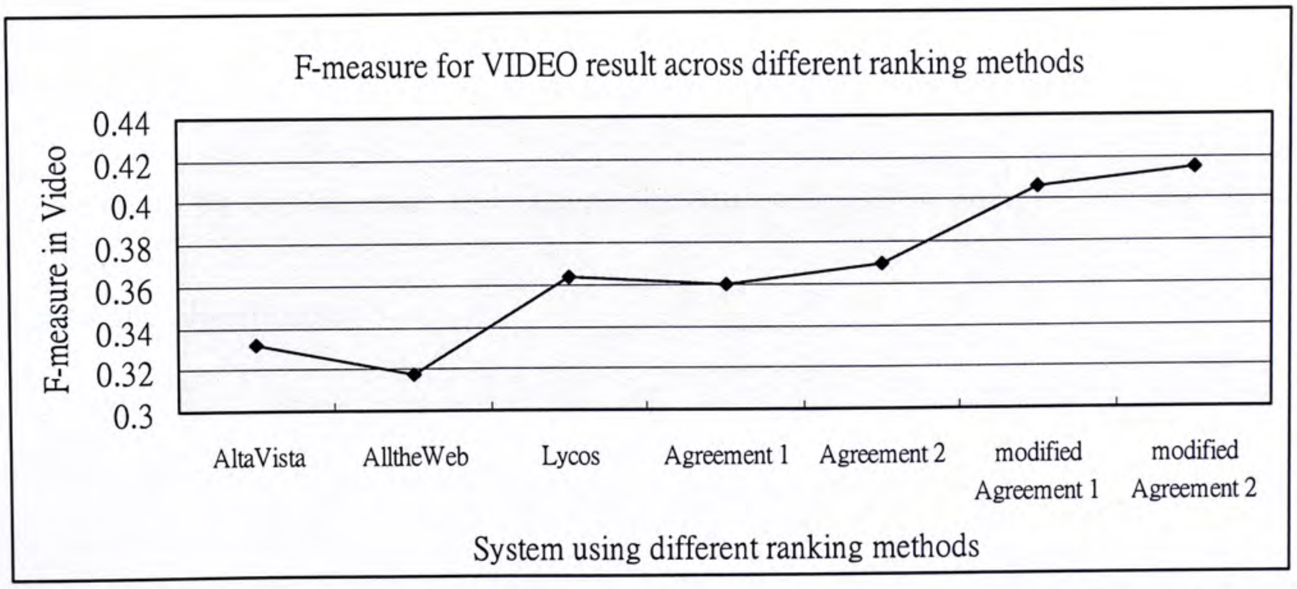


Figure 4.15 Average F-measure vs ranking methods for systems in retrieving video items

4.4.5 Discussion

From the foregoing tables, it can be seen that there is a gradual increase in precision, recall, and F-measure from the systems that do not use any merging methods (AltaVista, AlltheWeb, Lycos, and Google), to the systems that use the unmodified Agreement merging methods [8], and then to the systems that use our proposed merging algorithms.

We also find that the modified Agreement 2 algorithm (the re-ranking method that considers relationships using a linear scoring method) performs better than modified Agreement 1, which is a similar mechanism but uses a non-linear scoring method.

However, to calculate the degree of difference in the performance of the systems we perform statistical analyses on the evaluation measures.

The F-measure is the best performance measurement tool because it balances recall and precision, and thus we focus on this measure to evaluate the retrieval performance of the systems with the different merging algorithms.

4.5 Degree of difference between the performance of systems

4.5.1 Analysis using One-Way ANOVA

We conducted a one-way analysis of variance (ANOVA) to identify whether the merging algorithms have a significant effect on the retrieval performance of the meta-search engines as evaluated using F-measure, and the interactions between these effects.

Web

Source of Variation	*SS*	*df*	*MS*	*F*	***P-value***	***F crit***
Between Algorithms	0.004858	6	0.00081	2.618508	* **0.024905**	1.869819
Within	0.019479	63	0.000309			
Total	0.024337	69				

* *Significant at p < 0.05*

Table 4.16 One-way ANOVA examining the effects of the merging algorithms on the retrieval performance of the meta-search engines (retrieval of web items) as evaluated using the F-measure

Image

Source of Variation	*SS*	*df*	*MS*	*F*	***P-value***	*F crit*
Between Algorithms	0.055194	6	0.009199	3.797983	* **0.00272**	1.86982
Within	0.15259	63	0.002422			
Total	0.207784	69				

* *Significant at p < 0.01*

Table 4.17 One-way ANOVA examining the effects of the merging algorithms on the retrieval performance of the meta-search engines (retrieval of image items) as evaluated using the F-measure

Audio

Source of Variation	*SS*	*df*	*MS*	*F*	***P-value***	*F crit*
Between Groups	0.04654	6	0.007757	3.644331	* **0.003619**	1.869819
Within Groups	0.134091	63	0.002128			
Total	0.180631	69				

* *Significant at p < 0.01*

Table 4.18 One-way ANOVA examining the effects of the merging algorithms on the retrieval performance of the meta-search engines (retrieval of audio items) as evaluated using the F-measure

Video

Source of Variation	*SS*	*df*	*MS*	*F*	***P-value***	*F crit*
Between Groups	0.123204	6	0.020534	0.290398	**0.93938**	1.86982
Within Groups	4.454718	63	0.07071			
Total	4.577922	69				

Table 4.19 One-way ANOVA examining the effects of the merging algorithms on the retrieval performance of the meta-search engines (retrieval of video items) as evaluated using the F-measure

The p-values for the F-measures when all of the participating search engines and the other merging algorithms are considered together for the four media formats are 0.02491 for web items, 0.00272 for image items, 0.00361 for audio items, and 0.93938 for video items. There is a significant difference in the performance of the systems for different merging algorithms, except for the systems that retrieved the video items, which shows that the merging algorithms had a significant effect on the retrieval performance of the systems.

4.5.2 Analysis using paired samples T-test

To better understand the significance of the improvement that is brought about by using different merging algorithms, we employed a paired samples T-test to assess whether the averages of the F-measures of the two proposed merging algorithms are statistically different from each other. After performing the paired samples T-test on the 18 pairs of systems with different ranking methods, where each pair considers the F-measure performance of systems that use two different merging methods, we obtain the p-value for each pair of systems. We interpret the results in the following tables.

Web

Systems (retrieving web items) being compared/statistics	t Stat	P(T<=t) one-tail
AltaVista vs Agreement 1	-0.537	0.302
AltaVista vs Agreement 2	-1.853	** 0.048
AltaVista vs Modified Agreement 1	-1.639	* 0.068
AltaVista vs Modified Agreement 2	-2.258	** 0.025
AlltheWeb vs Agreement 1	-1.829	* 0.05
AlltheWeb vs Agreement 2	-3.126	** 0.006
AlltheWeb vs Modified Agreement 1	-2.902	** 0.009
AlltheWeb vs Modified Agreement 2	-3.554	** 0.003
Lycos vs Agreement 1	-1.475	* 0.087
Lycos vs Agreement 2	-2.143	** 0.03
Lycos vs Modified Agreement 1	-2.109	** 0.032
Lycos vs Modified Agreement 2	-2.915	** 0.009
Agreement 1 vs Agreement 2	-0.884	0.199
Agreement 1 vs Modified Agreement 1	-1.672	* 0.064
Agreement 1 vs Modified Agreement 2	-3.01	** 0.007
Agreement 2 vs Modified Agreement 1	-0.975	0.178
Agreement 2 vs Modified Agreement 2	-1.815	* 0.05
Modified Agreement 1 vs Modified Agreement 2	-2.349	** 0.021

** Significant at p < 0.1*

*** Significant at p < 0.05*

Table 4.20 P-values and other statistics from the paired samples T-test for the F-measures of systems retrieving web (textual) items

Image

Systems (retrieving image items) being compared/statistics	t Stat	P(T<=t) one-tail
AltaVista vs Agreement 1	-0.515	0.309
AltaVista vs Agreement 2	-2.254	** 0.025
AltaVista vs Modified Agreement 1	-2.91	** 0.009
AltaVista vs Modified Agreement 2	-3.329	** 0.004
AlltheWeb vs Agreement 1	-0.118	0.454
AlltheWeb vs Agreement 2	-1.68	* 0.063
AlltheWeb vs Modified Agreement 1	-2.543	** 0.016
AlltheWeb vs Modified Agreement 2	-3.014	** 0.007
Google vs Agreement 1	-2.42	** 0.019
Google vs Agreement 2	-3.182	** 0.0055
Google vs Modified Agreement 1	-3.338	** 0.004
Google vs Modified Agreement 2	-3.571	** 0.003
Agreement 1 vs Agreement 2	-1.5	* 0.084
Agreement 1 vs Modified Agreement 1	-2.357	** 0.021
Agreement 1 vs Modified Agreement 2	-2.709	** 0.012
Agreement 2 vs Modified Agreement 1	-2.96	** 0.008
Agreement 2 vs Modified Agreement 2	-3.585	** 0.003
Modified Agreement 1 vs Modified Agreement 2	-0.426	0.34

** Significant at p < 0.1*

*** Significant at p < 0.05*

Table 4.21 P-values and other statistics from the paired samples T-test for the F-measures of systems retrieving image items

Audio

Systems (retrieving audio items) being compared/statistics	t Stat	P(T<=t) one-tail
AltaVista vs Agreement 1	-1.447	* 0.091
AltaVista vs Agreement 2	-1.978	** 0.04
AltaVista vs Modified Agreement 1	-3.097	** 0.006
AltaVista vs Modified Agreement 2	-2.787	** 0.01
AlltheWeb vs Agreement 1	-1.687	* 0.063
AlltheWeb vs Agreement 2	-1.801	* 0.053
AlltheWeb vs Modified Agreement 1	-2.485	** 0.017
AlltheWeb vs Modified Agreement 2	-2.535	** 0.016
Lycos vs Agreement 1	-1.553	* 0.077
Lycos vs Agreement 2	-1.53	* 0.08
Lycos vs Modified Agreement 1	-2.967	** 0.008
Lycos vs Modified Agreement 2	-3.166	** 0.006
Agreement 1 vs Agreement 2	-0.431	0.338
Agreement 1 vs Modified Agreement 1	-3.495	** 0.003
Agreement 1 vs Modified Agreement 2	-2.884	** 0.009
Agreement 2 vs Modified Agreement 1	-2.435	** 0.019
Agreement 2 vs Modified Agreement 2	-2.299	** 0.024
Modified Agreement 1 vs Modified Agreement 2	-1.6	* 0.07

* *Significant at $p < 0.1$*

** *Significant at $p < 0.05$*

Table 4.22 P-values and other statistics from the paired samples T-test for the F-measures of systems retrieving audio items

Video

Systems (retrieving video items) being compared/statistics	t Stat	P(T<=t) one-tail
AltaVista vs Agreement 1	-0.967	0.179
AltaVista vs Agreement 2	-1.358	0.104
AltaVista vs Modified Agreement 1	-3.047	** 0.007
AltaVista vs Modified Agreement 2	-2.964	** 0.008
AlltheWeb vs Agreement 1	-1.563	* 0.076
AlltheWeb vs Agreement 2	-2.039	** 0.036
AlltheWeb vs Modified Agreement 1	-3.537	** 0.003
AlltheWeb vs Modified Agreement 2	-3.29	** 0.005
Lycos vs Agreement 1	*0.213*	0.418
Lycos vs Agreement 2	-0.387	0.354
Lycos vs Modified Agreement 1	-2.754	** 0.011
Lycos vs Modified Agreement 2	-2.407	** 0.02
Agreement 1 vs Agreement 2	-1.788	* 0.054
Agreement 1 vs Modified Agreement 1	-2.335	** 0.022
Agreement 1 vs Modified Agreement 2	-1.914	** 0.044
Agreement 2 vs Modified Agreement 1	-2.11	** 0.032
Agreement 2 vs Modified Agreement 2	-1.705	* 0.061
Modified Agreement 1 vs Modified Agreement 2	-0.867	0.204

* *Significant at* $p < 0.1$

** *Significant at* $p < 0.05$

Table 4.23 P-values and other statistics from the paired samples T-test for the F-measures of systems retrieving video items

The results highlight that our proposed algorithms outperform the existing algorithms significantly, and that Modified Agreement 2 (which uses the linear scoring method) outperforms Modified Agreement 1 (which uses the non-linear scoring method), with significant improvement for systems in retrieving web and audio items. This suggests that Modified Agreement 2 improved the retrieval performance of meta-search engines to the greatest extent and is thus the best of the algorithms in this thesis.

Appendix A Tables A.1 to A.24 give the complete statistical results for the paired samples T-test on the F-measures of systems retrieving items of the four types of media.

As a result from the T-tests and ANOVA, it is shown that the proposed algorithm with linear scoring method, Modified Agreement 2, always improves the performance of meta-search engines and even significantly in most cases.

Chapter 5

Conclusions and Future Work

5.1 Implications, Limitations, and Future Work

Implications

Meta-search engines are important in helping users to obtain information quickly and precisely through the Internet. However, as more and more information becomes available on the Internet a system that can retrieve information that is relevant to the queries of users and ranks this information according to relevance in descending order is therefore essential. In this thesis, we contribute to the development of this area by devising a meta-search merging algorithm that ascertains the relationships between the items that are retrieved by the participating search engines in a meta-search, and utilizes them to evaluate the importance of items with respect to the query. The algorithm then ranks the items in descending order of their relevance score, so that a higher F-measure of information retrieval can be achieved. Our experiment demonstrates that by considering both the quantity and strength of the relationships between items during the merging process, in addition to their original rankings, the retrieval performance is improved remarkably compared to situations in which merging methods are used that do not consider the relationship factor.

Limitations

In the experiment that we conducted, only ten queries were used, which may not be a large enough sample size. Another experiment with a larger sample size (for example, 30 queries) could be used to evaluate system performance. Besides, the same set of query terms are used, for both observation of the relationships between retrieved objects and the effects of these relationships on the relevance of objects in section 3.4.1, and for conduction of the experiment to evaluate the proposed merging algorithms in Chapter 4. This can cause bias, as the experimental result must match what we found from the observation to a certain extent, a different set of query terms should thus be used to conduct the experiment. Moreover, the experiment was performed by only one candidate (user), and the accuracy of the evaluation could thus be improved by recruiting more users as judges of relevancy. However, as the result illustrates that the performance is improved after the application of our proposed algorithm, it can be used as a benchmark for further studies. Furthermore, the relationship weightings that were found using the metrics as detailed in section 3.3.1 may not be optimal parameters, that is, a much better system performance may be reached if other sets of relationship weightings were used. These weightings should therefore be fine-tuned to obtain optimal ranking performance.

Future Work

In the future, we could improve the system by considering the aforementioned limitations in our current system. In addition, we could further extend the work by migrating the system to a mobile version so that users can benefit from using meta-search engines through their mobile phones or PDA.

5.2 Conclusions

This thesis presents a meta-search engine that helps users to search for relevant information online. The proposed merging algorithm (Modified Agreement 2) is shown to enable the better retrieval of relevant information over existing algorithms.

Bibliography

[1] D. Dreilinger and A. E. Howe. Experiences with Selecting Search Engines Using Metasearch. In *Proceedings of the ACM Transactions on Information Systems*, Volume 15, No. 3, pages 195-222, July 1997.

[2] Y. P. Shen, D. L. Lee. A Meta-search Method Reinforced by Cluster Descriptors. In *Second International Conference on Web Information Systems Engineering (WISE'01)*, Volume 1, pages 125-132, 2001.

[3] W. Y. Meng and Z. H. Wu. A Highly Scalable and Effective Method for Metasearch. In *Proceedings of the ACM Transactions on Information Systems (TOIS)*, Volume 19, No. 3, pages 310-335, July 2001.

[4] J. Callan, F. Crestani, H. Nottelmann, P. Pala, X. M. Shou. Resource Selection and Data Fusion in Multimedia Distributed Digital Libraries. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Canada*, pages 363-364, July 28-August 1, 2003.

[5] Javed A. Aslam, M. Montague. Models for Metasearch. *Annual ACM Conference on Research and Development in Information Retrieval, Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276-284, September 2001.

[6] M. G. S. Gauch, G. Wang. Profusion - Intelligent fusion from multiple, distributed search engines. *Journal of Universal Computer Science*, Volume. 2, No. 9, pages 637-649, Sept 1996.

[7] J. A. Shaw and A. E. A. Fox. Combination of Multiple Searches. In *Proceedings of the 2nd Text Retrieval Conference (TREC-2)*, pages 243-252, 1993.

[8] V. Kumar, B. U. Oztekin, G. Karypis. Expert Agreement and Content Based Reranking in a Meta Search Environment using Mearf. In *Proceedings of the eleventh international conference on World Wide Web Conference*, Honolulu, Hawaii, USA, pages 333-344, May 7-11, 2002.

[9] X. H. Yang, H. Yang, M. J. Zhang. Fusion Methods based on common order invariability for meta search engine systems. In *Proceedings of International Conference on Intelligent Agents, Web Technologies, and Internet Commerce*, Las Vegas, pages 310-317, 2001.

[10] E. Selberg and O. Etzioni. The MetaCrawler Architecture for Resource Aggregation on the Web. *IEEE Expert*, Volume. 12, No. 1, pages 8-14, January/February 1997.

[11] E. Selberg and O. Etzioni. Multi-Service Search and Comparison Using the MetaCrawler. In *Proceedings of the 4th International World-Wide Web Conference*, Volume. 39, No.11, pages 65-68, 1995.

[12] B. Schmitt and S. Oberlander. Evaluating and Enhancing Meta-Search Performance in Digital Libraries. In *Proceedings of the 3rd International Conference on Web*

Information Systems Engineering (WISE), Singapore, pages. 93-104, December 12 - 14, 2002.

[13] Christopher C. Vogt. Adaptive Combination of Evidence for Information Retrieval. *PhD thesis*, University of California, San Diego, 1999.

[14] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. In *Proceedings of the ACM Transactions on Information Systems*, Volume. 9, No. 3, pages 187-222, 1991.

[15] D. Dreilinger and A. E. Howe. SAVVYSEARCH: A metasearch engine that learns which search engines to query. *Journal of AI Magazine*, Volume. 18, No. 2, pages 19-25, 1997.

[16] P. Callan, Z. Lu, and W. B. Croft. Searching Distributed Collections with Inference Networks. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, Washington, ACM Press, pages 21-28, 1995.

[17] L. Gravano & H. Garcia-Molina. Merging Ranks from Heterogeneous Internet Sources. In *Proceedings of the 23rd VLDB Conference*, Athens, Greece, 1997.

[18] Dr. Wolfgang Sander-Beuermann. The Next Generation of Internet Search engines, *http://meta.rrzn.uni-hannover.de/eusidic/index.htm, Presentation at the EUSIDIC* Spring Meeting, Strassbourg, 1999.

[19] Y. Tzizikas. Democratic Data Fusion for Information Retrieval Mediators. In *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications*, page 531, 2001.

[20] Christopher C. Vogt and G. W. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, Volume. 1, Issue 3, pages 151-173, 1999.

[21] T. H. Haveliwala, A. Gionis, D. Klein, P. Indyk. Evaluating Strategies for Similarity Search on the Web. In *Proceedings of the eleventh international conference on World Wide Web*, Honolulu, Hawaii, USA, pages 432-442, May 7-11, 2002.

[22] C. Yu, W. Y. Meng, W. S. Wu, K. L. Liu. Efficient and Effective Metasearch for Text Databases Incorporating Linkages among Documents. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, Santa Barbara, California, United States, pages 187-198, 2001.

[23] S. Lawrence and C. L. Giles. Searching the world wide web. *Science*, 280:98-100, 1998.

[24] S. Lawrence and C. L. Giles. Accessibility of information on the web. *Nature*, 400:107-109, 1999.

[25] A. Broder. Web searching technology overview. *In Advanced school and Workshop on Models and Algorithms for theWorld Wide Web*, Udine, Italy, 2002.

[26] B. J. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real life Information Retrieval: A study of user queries on the Web. *ACM SIGIR Forum*, 32:5–17, 1998.

[27] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a very large AltaVista query log. *Technical Report 1998-014, Digital SRC*, 1998.

[28] B. Shu and S. Kak. A neural network-based intelligent metasearch engine. *Information Sciences*, 120:1-11, 1999.

[29] S. Lawrence and C. L. Giles. Context and page analysis for improved web search. *IEEE Internet Computing*, pages 38-46, July 1998.

[30] N. J. Belkin, P. Kantor, E. A. Fox & J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Information Processing & Management*, Volume 31, No.3, pages 431-448, 1995.

[31] R. R. Yager & A. Rybalov. On the Fusion of Documents from Multiple Collection Information Retrieval Systems. *Journal of the American Society for Information Science*, Volume 49, No. 13, pages 1177-1184, 1998.

[32] E. M. Voorhees, N. K. Gupta. and B. Johnson-Laird. The collection fusion problem. In *Proceedings of the Third Text Retrieval (TREC-3) Conference*, pages 95-104, 1994.

[33] Mario Gomez Susan Gauch, Guijun Wang. Information Fusion with ProFusion. In *Proceedings of the WebNet96: The First Conference on the Web Society*, San Francisco, CA, USA, October 1996.

[34] J. Savoy, A. Le Calve and D. Vrajitoru. Report on the TREC-5 Experiment: Data Fusion and Collection Fusion. In *Proceedings TRECS*, NIST Publication 500-238, Gaithersburg (MD), pages 489-502, 1996.

[35] C. Yu, Z.G. Li, W.J. Meng, Z.H. Wu. A highly scalable and effective method for metasearch. In *Proceedings of ACM Transactions on Information Systems (TOIS)*, Volume 19, Issue 3, pages 310 – 335, 2001.

[36] J. X. Xu and J. Callan. Effective retrieval with distributed collections. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, August 1998.

[37] W. B. Croft, A. Moffat, C.J. vab Rijsbergen, R. Wilkinson, and J. Zobel. In *SIGIR'98, Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, ACM Press, New York, August 1998.

[38] D. Hawking, N. Craswell and P. Thistlewaite. Merging results from isolated search engines. In *Proceedings of the Tenth Australian Database Conference*, Aukland, New Zealand, January 1999.

[39] W. Garbe. BINGOOO – transforming the world wide web into a virtual database. *Wirtschaftsinformatik*, Volume 43, No. 5, pages 511, October 2001.

[40] J. Callan, L. S. Larkey, M. E. Connell. Collection Selection and Results Merging with Topically Organized U.S. Patents and TREC Data. In *Proceedings of the ninth international conference on Information and knowledge management*, November 2000.

[41] C. Yu, W. Y. Meng, K. L. Liu, W. S. Wu, N. Rishe. Efficient and Effective Metasearch for a Large Number of Text Databases. In *Proceedings of the eighth international conference on Information and knowledge management*, November 1999.

[42] A. L. Powell, J. C. French, J. Callan, M. Connell, C. L. Viles. The impact of database selection on distributed searching. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, Athens, Greece, pages 232 – 239, 2000.

[43] Christopher C. Vogt and G. W. Cottrell. Predicting the performance of linearly combined IR systems. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, August 1998.

[44] Y. Chen, Q. Zhu, N. Wang. Query processing with quality control in the World Wide Web. *World Wide Web*, Volume 1 Issue 4, April 1998.

[45] B. Chidlovskii, U. M. Borghoff. Semantic caching of Web queries. *The International Journal on Very Large Databases*, Volume 9 Issue 1, March 2000.

[46] T. Strzalkowski, J. Wang, B. Wise. Summarization-based query expansion in information retrieval. In *Proceedings of the 17th international conference on Computational linguistics*, Volume 2, pages 1258 – 1264, Montreal, Quebec, Canada, 1998.

[47] G. M. Héctor, C. Chen, K. Chang, A. Paepcke. Predicate rewriting for translating Boolean queries in a heterogeneous information system. In *ACM Transactions on Information Systems (TOIS)*, Volume 17, Issue 1, pages 1 - 39, January 1999.

[48] O. Zamir. Visualization of search results in document retrieval systems. *General Examination Report*, University of Washington, 1998.

[49] O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 46-54, 1998.

[50] T.M. Mann. Visualization of WWW-Search Results. In *Proceedings of the International Workshop on Web-Based Information Visualization (WebVis'99)*, Florence, Italy, pages 264-268, September 1-3 1999.

[51] J. Cugini, S. Laskowski, M. Sebrechts. Design of 3D Visualization of Search Results: Evolution and Evaluation. In *Proceedings of IST/SPIE's 12th Annual International Symposium: Electronic Imaging 2000: Visual Data Exploration and Analysis (SPIE 2000)*, pages 23-28, San Jose, CA, January 2000.

[52] J. V. Cugini, M. M. Sebrechts, S. J. Laskowski, J. Vasilakis, M. S. Miller. Visualization of search results: a comparative evaluation of text, 2D, and 3D interfaces. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3 - 10, Berkeley, California, United States, 1999.

[53] M. Carey, D. C. Heesch and S. M. Rüger. Info Navigator: A Visualization Tool for

Document Searching and Browsing. In *Proceedings of the International Conference on Distributed Multimedia Systems, Florida*, September 2003.

[54] S. Roy, K. Kummamuru, R. Lotlikar, K. Singal, R. Krishnapuram. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In *Proceedings of the 13th conference on World Wide Web*, pages 658 – 665, New York, USA, 2004.

[55] M .H. Chignell, J. Gwizdka, R. C. Bodner. Discriminating meta-search: a framework for evaluation. In *Information Processing & Management*, Volume 35, No. 3, pages 337-362, 1999.

[56] E. J. Glover, S. Lawrence, W. P. Birmingham, C. L. Giles. Architecture of a meta-search engine that supports user information needs. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 210 – 216, Kansas City, Missouri, United States, 1999.

[57] H. Chen, S. Dumais. Bringing order to the Web: automatically categorizing search results. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145 – 152, The Hague, The Netherlands, 2000.

[58] Y. Lu, C.i Hu, X. Q. Zhu, H. J. Zhang, Q. Yang. A unified framework for semantics and feature based relevance feedback in image retrieval systems. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 31 – 37, Marina del Rey, California, United States, 2000.

[59] Y. H. Gong, X. Liu. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19 – 25, New Orleans, Louisiana, United States, 2001.

[60] J. Rucker and M. J. Polanco. Siteseer: Personalized navigation for the web. In *Communications of the ACM*, 40(3):73–75, March 1997.

[61] C. R. Anderson, E. Horvitz. Web Montage: A dynamic personalized start page. In *Proceedings of the eleventh international conference on World Wide Web*, pages 704 – 712, Honolulu, Hawaii, USA, 2002.

[62] S. Dumais, T. Joachims, K. Bharat, A. Weigend. SIGIR 2003 Workshop Report : Implicit Measures of User Interests and Preferences. *ACM SIGIR Forum*, Volume 37, Issue 2, pages 50 – 54, 2003.

[63] D. Kelly, J. Teevan. Implicit feedback for inferring user preference: a bibliography. *ACM SIGIR Forum*, Volume 37, Issue 2, pages 18 – 28, 2003.

[64] D.K. Harman. *The Second Text Retrieval Conference (TREC-2)*, Gaithersburg, MD, USA, March 1994.

[65] K. B. Ng, D. Loewenstern, C. Basu, H. Hirsh, and P. B. Kantor. Data fusion of machine-learning methods for the TREC5 routing task (and other work). In *The Fifth Text REtrieval Conference (TREC-5),* pages 477-488.

[66] E. M. Voorhees and D. K. Harman. *The Fifth Text Retrieval Conference (TREC-5)*,

Gaithersburg, MD, USA, 1997.

[67] K. L. Fox, O. Frieder, M. Knepper, and E. Snowberg. SENTINEL - A multiple engine information retrieval and visualization system. *Journal of the ASIS*, 50(7), May 1999.

[68] J. H. Lee. Analysis of Multiple Evidence Combination". In *Proceedings of the 20th annual ACM SGIR Conference*, pages 267-275, Philadelphia, Pennsylvania, July 1997.

[69] E. W. Selberg. Towards Comprehensive Web Search. *PhD thesis, University of Washington*, 1999.

[70] K. B. Ng and P. Kantor. An investigation of the preconditions for effective data fusion in IR: a pilot study. In *Proceedings of the 61st Annual Meeting of the American Society for Information Science*, 1998.

[71] C. J. V. Rijsbergen. Information Retrieval. *Butterworths*, Boston, London, 2nd edition, 1979.

[72] J. Wiley & Sons, Inc. MetaSpider:meta-searching and categorization on the Web. *Journal of the American Society for Information Science and Technology*, Volume 52, Issue 13, pages 1134–1147, November 2001.

[73] S. Lawrence, C. L. Giles. Inquirus - Text and Image Metasearch on the Web. In *International Conference on Parallel and Distributed Processing Techniques and Applications*, pages 829–835, 1999.

[74] J. Calmet, P. Kullmann. KOMET - Meta Web Search with KOMET. In *Intelligent Information Integration*, Volume 3, July 1999.

[75] D. A. Hawking, P. R. Bailey, and D. P. Campbell. PADRE - Parallel Document Retrieval Server For The World Wide Web. In *Proceedings of Australian Document Computing Symposium*, pages 73-78, Melbourne Australia, March 1996.

Appendix A

Paired samples T-test for F-measures of systems retrieving all media's items

Tests below are all done at $\alpha = 0.1$ significance level

Paired samples T-test for F-measure of systems retrieving web (textual) items

F-measures								
	AltaVista	Agreement 1		AltaVista	Agreement 2		AltaVista	modified Agreement 1
Mean	0.201232	0.204247		0.201232	0.208404		0.201232	0.213276
Variance	0.000188	0.000259		0.000188	0.000336		0.000188	0.00058
Observations	10	10		10	10		10	10
Pearson Correlation	0.300492			0.744344			0.346166	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-0.53754			-1.85328			-1.63934	
P(T<=t) one-tail	0.301966			0.048422			0.067783	
t Critical one-tail	1.383029			1.383029			1.383029	
P(T<=t) two-tail	0.603931			0.096845			0.135565	
t Critical two-tail	1.833113			1.833113			1.833113	

Table A.1 Paired samples T-test for F-measures between systems (retrieval of Web items)

F-measures								
	AltaVista	modified Agreement 2		Agreement 1	Agreement 2		Agreement 1	modified Agreement 1
Mean	0.201232	0.219169		0.204247	0.208404		0.204247	0.213276
Variance	0.000188	0.000535		0.000259	0.000336		0.000259	0.00058
Observations	10	10		10	10		10	10
Pearson Correlation	0.144275			0.633319			0.706642	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-2.25777			-0.88356			-1.67239	
P(T<=t) one-tail	0.02518			0.19996			0.064389	
t Critical one-tail	1.383029			1.383029			1.383029	
P(T<=t) two-tail	0.05036			0.399919			0.128778	
t Critical two-tail	1.833113			1.833113			1.833113	

Table A.2 Paired samples T-test for F-measures between systems (retrieval of Web items) - continued

F-measures								
	Agreement 1	modified Agreement 2		Agreement 2	modified Agreement 1		Agreement 2	modified Agreement 2
Mean	0.204247	0.219169		0.208404	0.213276		0.208404	0.219169
Variance	0.000259	0.000535		0.000336	0.00058		0.000336	0.000535
Observations	10	10		10	10		10	10
Pearson Correlation	0.736239			0.754772			0.612012	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-3.01047			-0.97472			-1.8149	
P(T<=t) one-tail	0.007352			0.177583			0.051466	
t Critical one-tail	1.383029			1.383029			1.383029	
P(T<=t) two-tail	0.014705			0.355166			0.102933	
t Critical two-tail	1.833113			1.833113			1.833113	

Table A.3 Paired samples T-test for F-measures between systems (retrieval of Web items) - continued

F-measures								
	modified Agreement 1	modified Agreement 2		AlltheWeb	Agreement 1		AlltheWeb	Agreement 2
Mean	0.213276	0.219169		0.195379	0.204247		0.195379	0.208404
Variance	0.00058	0.000535		0.000172	0.000259		0.000172	0.000336
Observations	10	10		10	10		10	10
Pearson Correlation	0.9444			0.465268			0.695296	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-2.34995			-1.82962			-3.12591	
P(T<=t) one-tail	0.021654			0.050278			0.006101	
t Critical one-tail	1.383029			1.383029			1.383029	
P(T<=t) two-tail	0.043309			0.100555			0.012202	
t Critical two-tail	1.833113			1.833113			1.833113	

Table A.4 Paired samples T-test for F-measures between systems (retrieval of Web items) - continued

F-measures							
	AlltheWeb	modified Agreement 1		AlltheWeb	modified Agreement 2		Lycos	Agreement 1
Mean	0.195379	0.213276		0.195379	0.219169		0.195098	0.204247
Variance	0.000172	0.00058		0.000172	0.000535		9.37E-05	0.000259
Observations	10	10		10	10		10	10
Pearson Correlation	0.588767			0.426633			-0.10192	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-2.90186			-3.55445			-1.47495	
P(T<=t) one-tail	0.008771			0.003086			0.087164	
t Critical one-tail	1.383029			1.383029			1.383029	
P(T<=t) two-tail	0.017542			0.006173			0.174329	
t Critical two-tail	1.833113			1.833113			1.833113	

Table A.5 Paired samples T-test for F-measures between systems (retrieval of Web items) - continued

F-measures								
	Lycos	Agreement 2		Lycos	modified Agreement 1		Lycos	modified Agreement 2
Mean	0.195098	0.208404		0.195098	0.213276		0.195098	0.219169
Variance	9.37E-05	0.000336		9.37E-05	0.00058		9.37E-05	0.000535
Observations	10	10		10	10		10	10
Pearson Correlation	0.124246			-0.14749			-0.11941	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-2.14327			-2.10897			-2.91526	
P(T<=t) one-tail	0.030349			0.032088			0.008582	
t Critical one-tail	1.383029			1.383029			1.383029	
P(T<=t) two-tail	0.060697			0.064177			0.017164	
t Critical two-tail	1.833113			1.833113			1.833113	

Table A.6 Paired samples T-test for F-measures between systems (retrieval of Web items) - continued

Paired samples T-test for F-measures of systems retrieving image items

F-measures								
	AltaVista	Agreement 1		AltaVista	Agreement 2		AltaVista	modified Agreement 1
Mean	0.210748651	0.215093182		0.210748651	0.231245252		0.210748651	0.264721119
Variance	0.000976528	0.001596764		0.000976528	0.002014702		0.000976528	0.005534676
Observations	10	10		10	10		10	10
Pearson Correlation	0.745939214			0.771486169			0.660323409	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-0.515468996			***-2.253888024***			-2.909601349	
P(T<=t) one-tail	0.309323646			0.025340139			0.008661243	
t Critical one-tail	1.383028739			***1.383028739***			1.383028739	
P(T<=t) two-tail	0.618647292			0.050680278			0.017322486	
t Critical two-tail	1.833112923			***1.833112923***			1.833112923	

Table A.7 Paired samples T-test for F-measures between systems (retrieval of Image items)

F-measures								
	AltaVista	modified Agreement 2		Agreement 1	Agreement 2		Agreement 1	modified Agreement 1
Mean	0.210748651	0.266640667		0.215093182	0.231245252		0.215093182	0.264721119
Variance	0.000976528	0.004637274		0.001596764	0.002014702		0.001596764	0.005534676
Observations	10	10		10	10		10	10
Pearson Correlation	0.65669515			0.681437718			0.45395948	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-3.328974074			-1.495170069			-2.357263017	
P(T<=t) one-tail	0.004406733			0.08454144			0.021396422	
t Critical one-tail	1.383028739			1.383028739			1.383028739	
P(T<=t) two-tail	0.008813466			0.169082881			0.042792844	
t Critical two-tail	1.833112923			1.833112923			1.833112923	

Table A.8 Paired samples T-test for F-measures between systems (retrieval of Image items) - continued

F-measures								
	Agreement 1	modified Agreement 2		Agreement 2	modified Agreement 1		Agreement 2	modified Agreement 2
Mean	0.215093182	0.266640667		0.231245252	0.264721119		0.231245252	0.266640667
Variance	0.001596764	0.004637274		0.002014702	0.005534676		0.002014702	0.004637274
Observations	10	10		10	10		10	10
Pearson Correlation	0.480279138			0.93891971			0.928669576	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-2.709193561			-2.960304584			-3.584895987	
P(T<=t) one-tail	0.012014603			0.007975812			0.002942768	
t Critical one-tail	1.383028739			1.383028739			1.383028739	
P(T<=t) two-tail	0.024029206			0.015951624			0.005885536	
t Critical two-tail	1.833112923			1.833112923			1.833112923	

Table A.9 Paired samples T-test for F-measures between systems (retrieval of Image items) - continued

F-measures								
	modified Agreement 1	modified Agreement 2		AlltheWeb	Agreement 1		AlltheWeb	Agreement 2
Mean	0.264721119	0.266640667		0.214274112	0.215093182		0.214274112	0.231245252
Variance	0.005534676	0.004637274		0.000752081	0.001596764		0.000752081	0.002014702
Observations	10	10		10	10		10	10
Pearson Correlation	0.983843209			0.85167156			0.711495905	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-0.425654579			-0.117949			-1.684392	
P(T<=t) one-tail	0.340178659			0.454349273			0.063195622	
t Critical one-tail	1.383028739			1.3830287			1.3830287	
P(T<=t) two-tail	0.680357319			0.908698546			0.126391244	
t Critical two-tail	1.833112923			1.8331129			1.8331129	

Table A.10 Paired samples T-test for F-measures between systems (retrieval of Image items) - continued

F-measures								
	AlltheWeb	modified Agreement 1		AlltheWeb	modified Agreement 2		Google	Agreement 1
Mean	0.214274112	0.264721119		0.214274112	0.266640667		0.182944777	0.215093182
Variance	0.000752081	0.005534676		0.000752081	0.004637274		0.001442474	0.001596764
Observations	10	10		10	10		10	10
Pearson Correlation	0.57619896			0.634510252			0.4199671	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-2.542902			-3.01363472			-2.42018284	
P(T<=t) one-tail	0.015780064			0.007314711			0.019299528	
t Critical one-tail	1.3830287			1.383028739			1.383028739	
P(T<=t) two-tail	0.031560129			0.014629421			0.038599055	
t Critical two-tail	1.8331129			1.833112923			1.833112923	

Table A.11 Paired samples T-test for F-measures between systems (retrieval of Image items) - continued

F-measures								
	Google	Agreement 2		Google	modified Agreement 1		Google	modified Agreement 2
Mean	0.182944777	0.231245252		0.182944777	0.231245252		0.182944777	0.266640667
Variance	0.001442474	0.002014702		0.001442474	0.002014702		0.001442474	0.004637274
Observations	10	10		10	10		10	10
Pearson Correlation	0.338412809			0.338412809			0.113252417	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-3.182514698			-3.182514698			-3.570775836	
P(T<=t) one-tail	0.005569843			0.005569843			0.003008467	
t Critical one-tail	1.383028739			1.383028739			1.383028739	
P(T<=t) two-tail	0.011139687			0.011139687			0.006016934	
t Critical two-tail	1.833112923			1.833112923			1.833112923	

Table A.12 Paired samples T-test for F-measures between systems (retrieval of Image items) - continued

Paired samples T-test for F-measures of systems retrieving audio items

	AltaVista	Agreement 1		AltaVista	Agreement 2		AltaVista	modified Agreement 1
Mean	0.297009321	0.318120612		0.297009321	0.321285157		0.297009321	0.358576943
Variance	0.015734403	0.022280942		0.015734403	0.023195367		0.015734403	0.021660134
Observations	10	10		10	10		10	10
Pearson Correlation	0.958307004			0.97946376			0.905732359	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-1.446786787			-1.977968368			-3.096608111	
P(T<=t) one-tail	0.090935447			0.039659289			0.006396144	
t Critical one-tail	1.383028739			1.383028739			1.383028739	
P(T<=t) two-tail	0.181870895			0.079318578			0.012792287	
t Critical two-tail	1.833112923			1.833112923			1.833112923	

Table A.13 Paired samples T-test for F-measures between systems (retrieval of Audio items)

F-measures								
	AltaVista	modified Agreement 2		Agreement 1	Agreement 2		Agreement 1	modified Agreement 1
Mean	0.297009321	0.366673366		0.318120612	0.321285157		0.318120612	0.358576943
Variance	0.015734403	0.022146028		0.022280942	0.023195367		0.022280942	0.021660134
Observations	10	10		10	10		10	10
Pearson Correlation	0.84727403			0.98834851			0.969608916	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-2.786916464			-0.431057641			-3.495326056	
P(T<=t) one-tail	0.010580102			0.338283332			0.003386569	
t Critical one-tail	1.383028739			1.383028739			1.383028739	
P(T<=t) two-tail	0.021160204			0.676566663			0.006773138	
t Critical two-tail	1.833112923			1.833112923			1.833112923	

Table A.14 Paired samples T-test for F-measures between systems (retrieval of Audio items) - continued

F-measures								
	Agreement 1	modified Agreement 2		Agreement 2	modified Agreement 1		Agreement 2	modified Agreement 2
Mean	0.318120612	0.366673366		0.321285157	0.358576943		0.321285157	0.366673366
Variance	0.022280942	0.022146028		0.023195367	0.021660134		0.023195367	0.022146028
Observations	10	10		10	10		10	10
Pearson Correlation	0.936218507			0.948280982			0.914288186	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-2.884220683			-2.43534197			-2.299088869	
P(T<=t) one-tail	0.009026719			0.018825606			0.023534755	
t Critical one-tail	1.383028739			1.383028739			1.383028739	
P(T<=t) two-tail	0.018053439			0.037651211			0.04706951	
t Critical two-tail	1.833112923			1.833112923			1.833112923	

Table A.15 Paired samples T-test for F-measures between systems (retrieval of Audio items) - continued

F-measures								
	modified Agreement 1	modified Agreement 2		AlltheWeb	Agreement 1		AlltheWeb	Agreement 2
Mean	0.256215	0.260688		0.23472877	0.318120612		0.23472877	0.321285157
Variance	0.004187	0.003657		0.004537392	0.022280942		0.004537392	0.023195367
Observations	10	10		10	10		10	10
Pearson Correlation	0.992283			0.118260416			0.225747558	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-1.59826			-1.6868301			-1.8008802	
P(T<=t) one-tail	0.072225			0.06295571			0.052621974	
t Critical one-tail	1.383029			1.38302874			1.38302874	
P(T<=t) two-tail	0.14445			0.125911421			0.105243949	
t Critical two-tail	1.833113			1.83311292			1.83311292	

Table A.16 Paired samples T-test for F-measures between systems (retrieval of Audio items) - continued

F-measures								
	AlltheWeb	modified Agreement 1		AlltheWeb	modified Agreement 2		Lycos	Agreement 1
Mean	0.23472877	0.358576943		0.23472877	0.366673366		0.296844781	0.318120612
Variance	0.004537392	0.021660134		0.004537392	0.022146028		0.023561207	0.022280942
Observations	10	10		10	10		10	10
Pearson Correlation	0.068228096			-0.02056177			0.959422597	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-2.4846906			-2.534789			-1.552811	
P(T<=t) one-tail	0.01736159			0.015991536			0.077442922	
t Critical one-tail	1.38302874			1.38302874			1.3830287	
P(T<=t) two-tail	0.03472318			0.031983072			0.154885844	
t Critical two-tail	1.83311292			1.83311292			1.8331129	

Table A.17 Paired samples T-test for F-measures between systems (retrieval of Audio items) - continued

F-measures								
	Lycos	Agreement 2		Lycos	modified Agreement 1		Lycos	modified Agreement 2
Mean	0.296844781	0.321285157		0.296844781	0.358576943		0.296844781	0.366673366
Variance	0.023561207	0.023195367		0.023561207	0.021660134		0.023561207	0.022146028
Observations	10	10		10	10		10	10
Pearson Correlation	0.945436407			0.905082961			0.89402777	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-1.529748			-2.967187			***-3.166418***	
P(T<=t) one-tail	0.08021714			0.007887162			0.005715813	
t Critical one-tail	1.3830287			1.3830287			***1.3830287***	
P(T<=t) two-tail	0.16043428			0.015774325			0.011431626	
t Critical two-tail	1.8331129			1.8331129			***1.8331129***	

Table A.18 Paired samples T-test for F-measures between systems (retrieval of Audio items) - continued

Paired samples T-test for F-measures of systems retrieving video items

F-measures								
	AltaVista	Agreement 1		AltaVista	Agreement 2		AltaVista	modified Agreement 1
Mean	0.331657838	0.36018557		0.331657838	0.369762691		0.331657838	0.40577601
Variance	0.014858633	0.027343068		0.014858633	0.023376063		0.014858633	0.02359278
Observations	10	10		10	10		10	10
Pearson Correlation	0.831131632			0.814436875			0.868804685	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-0.967377989			-1.35764988			-3.046781953	
P(T<=t) one-tail	0.179315068			0.103818173			0.006932451	
t Critical one-tail	1.383028739			1.383028739			1.383028739	
P(T<=t) two-tail	0.358630135			0.207636346			0.013864902	
t Critical two-tail	1.833112923			1.833112923			1.833112923	

Table A.19 Paired samples T-test for F-measures between systems (retrieval of Video items)

F-measures								
	AltaVista	modified Agreement 2		Agreement 1	Agreement 2		Agreement 1	modified Agreement 1
Mean	0.331657838	0.415013184		0.36018557	0.369762691		0.36018557	0.40577601
Variance	0.014858633	0.0264164		0.027343068	0.023376063		0.027343068	0.02359278
Observations	10	10		10	10		10	10
Pearson Correlation	0.84207158			0.99739977			0.927646488	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-2.963972026			-1.788148371			-2.334555726	
P(T<=t) one-tail	0.007928443			0.053692439			0.022207268	
t Critical one-tail	1.383028739			1.383028739			1.383028739	
P(T<=t) two-tail	0.015856887			0.107384877			0.044414537	
t Critical two-tail	1.833112923			1.833112923			1.833112923	

Table A.20 Paired samples T-test for F-measures between systems (retrieval of Video items) - continued

F-measures								
	Agreement 1	modified Agreement 2		Agreement 2	modified Agreement 1		Agreement 2	modified Agreement 2
Mean	0.36018557	0.415013184		0.369762691	0.40577601		0.369762691	0.415013184
Variance	0.027343068	0.0264164		0.023376063	0.02359278		0.023376063	0.0264164
Observations	10	10		10	10		10	10
Pearson Correlation	0.847521399			0.938014061			0.860152707	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-1.914203807			-2.110457697			-1.70504782	
P(T<=t) one-tail	0.043934166			0.032011089			0.061188865	
t Critical one-tail	1.383028739			1.383028739			1.383028739	
P(T<=t) two-tail	0.087868333			0.064022177			0.12237773	
t Critical two-tail	1.833112923			1.833112923			1.833112923	

Table A.21 Paired samples T-test for F-measures between systems (retrieval of Video items) - continued

F-measures								
	modified Agreement 1	modified Agreement 2		AlltheWeb	Agreement 1		AlltheWeb	Agreement 2
Mean	0.40577601	0.415013184		0.317810141	0.36018557		0.317810141	0.369762691
Variance	0.02359278	0.0264164		0.013655877	0.027343068		0.013655877	0.023376063
Observations	10	10		10	10		10	10
Pearson Correlation	0.978856924			0.870748109			0.854669686	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-0.866876718			-1.56332558			-2.03906367	
P(T<=t) one-tail	0.204263105			0.076206612			0.035936911	
t Critical one-tail	1.383028739			1.383028739			1.383028739	
P(T<=t) two-tail	0.408526209			0.152413223			0.071873822	
t Critical two-tail	1.833112923			1.833112923			1.833112923	

Table A.22 Paired samples T-test for F-measures between systems (retrieval of Video items) - continued

F-measures								
	AlltheWeb	modified Agreement 1		AlltheWeb	modified Agreement 2		Lycos	Agreement 1
Mean	0.317810141	0.40577601		0.317810141	0.415013184		0.364152016	0.36018557
Variance	0.013655877	0.02359278		0.013655877	0.0264164		0.020721502	0.027343068
Observations	10	10		10	10		10	10
Pearson Correlation	0.865268372			0.825158546			0.936451001	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	***-3.53661761***			-3.2902798			0.212513267	
P(T<=t) one-tail	0.003173846			0.00468708			0.418220916	
t Critical one-tail	***1.383028739***			1.383028739			1.383028739	
P(T<=t) two-tail	0.006347691			0.00937416			0.836441833	
t Critical two-tail	***1.833112923***			1.833112923			1.833112923	

Table A.23 Paired samples T-test for F-measures between systems (retrieval of Video items) - continued

F-measures								
	Lycos	Agreement 2		Lycos	modified Agreement 1		Lycos	modified Agreement 2
Mean	0.364152016	0.369762691		0.364152016	0.40577601		0.364152016	0.415013184
Variance	0.020721502	0.023376063		0.020721502	0.02359278		0.020721502	0.0264164
Observations	10	10		10	10		10	10
Pearson Correlation	0.954184955			0.950431022			0.911924456	
Hypothesized Mean Difference	0			0			0	
Df	9			9			9	
t Stat	-0.38748352			***-2.75352891***			-2.40657664	
P(T<=t) one-tail	0.353701314			0.011173719			0.019734982	
t Critical one-tail	1.383028739			***1.383028739***			1.383028739	
P(T<=t) two-tail	0.707402628			0.022347437			0.039469964	
t Critical two-tail	1.833112923			***1.833112923***			1.833112923	

Table A.24 Paired samples T-test for F-measures between systems (retrieval of Video items) - continued