# Optimizing Web Search Using Web Click-through Data*

Gui-Rong Xue[1] Hua-Jun Zeng[2] Zheng Chen[2] Yong Yu[1]
Wei-Ying Ma[2] WenSi Xi[3] WeiGuo Fan[3]

[1]Computer Science and Engineering
Shanghai Jiao-Tong University
Shanghai 200030, P.R.China
grxue@sjtu.edu.cn,
yyu@cs.sjtu.edu.cn

[2]Microsoft Research Asia
5F, Sigma Center, 49 Zhichun Road
Beijing 100080, P.R.China
{hjzeng, zhengc,
wyma}@microsoft.com

[3]Computer Science
Virginia Polytechnic Institute and State University
Virginia, U.S.A
{xwensi, wfan }@vt.edu

## ABSTRACT
The performance of web search engines may often deteriorate due to the diversity and noisy information contained within web pages. User click-through data can be used to introduce more accurate description (metadata) for web pages, and to improve the search performance. However, noise and incompleteness, sparseness, and the volatility of web pages and queries are three major challenges for research work on user click-through log mining. In this paper, we propose a novel iterative reinforced algorithm to utilize the user click-through data to improve search performance. The algorithm fully explores the interrelations between queries and web pages, and effectively finds "virtual queries" for web pages and overcomes the challenges discussed above. Experiment results on a large set of MSN click-through log data show a significant improvement on search performance over the naive query log mining algorithm as well as the baseline search engine.

## Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *Search process*; H.2.8 [**Database Management**]: Database Applications - *Data mining*

## General Terms
Algorithms, Performance, Experimentation.

## Keywords
Click-through Data, Iterative Algorithm, Log Mining, Search Engine

## 1. INTRODUCTION

Existing web search engines often calculate the relevancy of web pages for a given query by counting the search keywords contained in the web pages. This method works well when users' queries are clear and specific. However, in real world, web search queries are often short (less than 3 words [1]) and ambiguous, and web pages contain a lot diverse and noisy information. These will very likely lead to the deteriorating of the performance of web search engines, due to the gap between query space and document space [4][6]. This problem can be partially solved by using external evidence to enrich the content of existing web pages – the so-called *surrogate document* approach. One of such examples is to use anchor texts as additional description of target Web pages. Previous research [14][20][22] show that this method yields better search result than searching on Web page content alone. This is because anchor texts represent the view of a web page by other web editors rather its own author. Another solution is to introduce additional description by using click-through data, which has not been extensively studied.

User click-through data can be extracted from a large amount of search logs accumulated by web search engines. These logs typically contain user-submitted search queries, followed by the URL of Web pages which are clicked by users in the corresponding search result page. Although these clicks don't reflect the exact relevancy, they provide valuable indications to the users' intention by associating a set of query terms with a set of web pages. If a user clicks on a web page, it is likely that the web page is relevant to the query, or at least related to some extent. Many valuable applications have been proposed along this direction, such as term suggestion [3][15], query expansion [4], and query clustering [5][10].

In this paper we try to use user click-through data as the additional metadata to bridge the gap between users' information need and the content of the web pages. The query log based web page metadata generation method has three important properties. First, click-through data can be regarded as web searchers' view of web pages, and they are more valuable than anchor texts because the performance of web search engines are evaluated by web users not editors of web pages. Second, since such metadata can be combined with the content and other representations of Web pages, we reduce the risk of losing relevant web pages in a pure reranking algorithm. Third, the correlations between web pages and queries may evolve with the accumulation of click-through data. This process can reflect and update users' view of web pages as time goes by.

A naive method of applying user click-through data is to associate the queries with the clicked web page as the metadata of the Web pages. Furthermore, associated queries can be found by analyzing co-visited relationship of web pages [2], which we denoted as co-visited based method. The basic assumption of the co-visited method is that two web pages are similar if they are co-visited by users with similar queries, and the associated queries of the two web pages can be taken (merged) as the metadata for each other.

However, several issues are not solved in these methods. They are:


*CIKM'04*, November 8–13, 2004, Washington, DC, USA.


*This work was conducted while the author was doing internship at Microsoft Research Asia.

1. The clicks-through data may be very noisy and incomplete, and introduce inaccurate metadata to associated web pages. The co-visited method only considers the similarity of queries by content; it does not take into account that two queries are similar if they lead to the visit of similar web pages. Such kind of similarity can be propagated between the web pages and the queries, and the effect of noisy information in click-through data can be constrained.

2. Click-through data is very sparse, because web users are more likely to click popular (typically top 10) Web pages returned by a search engine. Existing methods cannot introduce metadata for most web pages with few clicks.

3. In real world, new web pages and queries often emerge constantly. We can not take advantage of the click-through data over the new items on web.

To address the above three issues, we propose in this paper an iterative reinforcement algorithm for computing the similarity between web pages and queries, which fully explore the relations between web pages and queries. The assumption of our algorithm is that, the similarities of queries can be affected by the similarities of web pages, and vice versa. Such a procedure is executed iteratively until the algorithm reaches a fixed point. After the similarity between web pages are computed, two web pages can share queries if their similarity value is bigger than a given threshold. We then assign the queries of one page to its similar pages as extra metadata.

Experimental results on a real large query click-through log, i.e. MSN query log data, indicate that our proposed algorithm relatively outperforms the baseline search system by 157%, naïve query log mining by 17% and co-visited algorithm by 17% on top 20 precision respectively.

The rest of this paper is organized as follows. In Section 2, we review some related work on web search improvement. In Section 3, we present the basic algorithm, and describe the mining technology to construct related queries to a web page. Our experimental results are presented in Section 4. Finally, conclusions and future works are discussed in Section 5.

## 2. RELATED WORK

Several methods have been proposed to find new content as additional metadata to enhance the performance of web search. For example, anchor text [14][20][21][22], title or surrounding text of the anchor were used to enhance the web search. These methods improve the performance of web search to some extent. For example, Google's search engine [20] takes the anchor text as it's metadata to improve the performance of search.

Query log analysis is extensively investigated in recent years. [23] proposed to reuse past optimal queries to improve search by re-formulating new queries. Recently, Joachims [11] propose a method of utilizing click-through data in learning of a retrieval function (e.g., a meta search function). Specifically, he introduces a new method for training a retrieval function on the basis of click-through data, which he calls Ranking SVM. His method is unique in that it takes the relative positions of the clicks in a rank as training data. New approaches [5][10] on query log analysis focus on query clustering and web pages clustering. The use of query log data to measure similarity between objects was found to be better than calculating similarities using objects' content vector. For example, Beeferman and Berger proposed an innovative query clustering method [5] based on click-through data. Each record of click-through data consists of a user's query to the search engine and the URLs that user actually visit among the list provided by the search engine. Treating click-through data sets as a bipartite graph and identifying the mapping between queries and the clicked URLs, queries with similarly clicked URLs can be clustered. It ignores the content features in both query and document, and the hyperlink interconnectivity information of web pages, either. Wen et. al [10] describes a query clustering method using user logs, in which two queries are similar if they contain the same terms or leading to the selection of the same retrieved documents. Unfortunately, these methods do not consider the web pages and queries as an integrated fashion, where each could reinforce the similarity of the other.

There are several work on bridging the gap between the query space and document space, such as term suggestion [3][15], query expansion[4] and spreading activation method [8]. Cui et.al [4] try to use click-through data to solve the problem of mismatch between the query term and document terms. Using click-through data, the probabilistic correlations between query terms and documents terms can be extracted and used in high-quality expansion of terms in new queries. Term suggestion is another method to associate user to browse. Huang et,al. [3] proposed to use the co-occur in similar query sessions to finding related terms. Salton et.al. [8] used spreading activation method to calculate the terms related to the already used terms and expand documents space.

## 3. GENERATING METADATA FOR WEB PAGES

In this section, we first define the problem of generating metadata from click-through data, followed by the naive method to solve such a problem. Then, a co-visited algorithm is proposed to find the missing click-through to solve the incompleteness issue of the query log mining. Finally, we propose our iterative algorithm which reinforces the similarity computing by a recursive procedure to solve the three issues present in Section 1.

### 3.1 Problem Description

We define click-through data as a set *Session*, each of which is defined as a pair of a query and a web page the user clicked on. Click-through data is generated from raw search logs, which may contain large amount of useless logs such as images and scripts, and random user behaviors. Through certain session split algorithm and noise filtering (which will be described in the experiment section), we could get more accurate click-through. We further assume that the set of clicked web pages $c$ is relevant to the query $q$. This assumption might be too strong in some cases because of some noisy clicks inside the data. But most users usually are likely to click on a relevant results, thus we benefit from a large quantity of query logs. Experiments show that 82% of the queries are in fact related to the topics of the clicked Web pages.

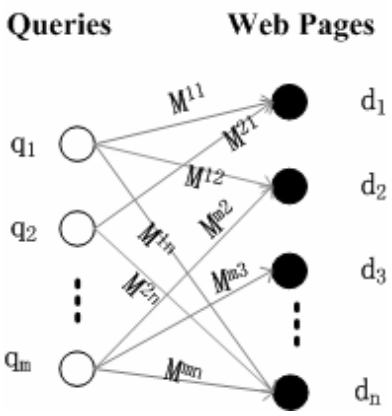


**Figure 1. Interrelations between queries and Web pages**

By merging same queries and web pages in the above sessions, click-through data could be modeled as a weighted directed bipartite graph $G=(V, E)$ as shown in Figure 1, where nodes in $V$ represent web pages and queries, edges $E$ represent the click-throughs from a query to a clicked web page, $M$ represents the weight on the edges. We can divide $V$ into two subsets $Q=\{q_1, q_2, \ldots, q_m\}$ and $D=\{d_1, d_2, \ldots, d_n\}$ where $Q$ represents the queries and $D$ represents the Web pages. The Based on the bipartite graph $G$, we select from queries $Q$ as additional metadata for web pages $D$. For example, in Figure 1, the web page $d_2$ is pointed by queries $q_1$ and $q_m$. From the view of the user, $q_1$ and $q_m$ can be taken as the related content or topics of the web pages $d_2$.

## 3.2 Naive Method (NM)

Given the bipartite graph $G$, an intuitive method, which is similar to DirectHIT, is to take the queries that associate with the web pages as the web pages' metadata directly.

As an example in Figure 1, web page $d_i$ ($i \in \{1, 2, \ldots, n\}$)is clicked by users on queries $q_1, q_2, \ldots, q_m$ with frequencies in different query sessions.

Each page $d_i$ and its query $q_j$ also have a weight $W_{ij}$, which is defined as following:

$$W_{ij} = W(d_i, q_j) = \frac{M_{ij}}{\sum_{k \in \{1,\ldots,m\}} M_{kj}} \quad (1)$$

Thus $w_{i1} \cdot q_1 + w_{i2} \cdot q_2 + \ldots + w_{im} \cdot q_m$ is taken as additional metadata for Web page $d_i$. Then, such metadata is treated as the same way of other metadata data, such as anchor text and title of the Web page.

## 3.3 Co-Visited Method (CVM)

It is easy to demonstrate that Naïve Method could achieve good performance if the query click-through data is complete, i.e. each query is associated with all the related documents. But unfortunately, we found that in the real world, each query will randomly be associated with only a few individual documents instead of whole list. This data incompleteness problem makes the performance of the naïve method drop significantly. Deriving from the co-citation in the scientific literature [9][12][19], we develop an analogous approach to find similar web pages. As shown in Figure 2, if the two web pages are clicked by mostly the same queries, it is possible that two web pages are similar. We define a term ***co-visited*** to represent such a relationship, which means that if two web pages are clicked by users with the same query, the two web pages are co-visited. Then, the queries, which are associated with one of the web pages, could be used as the related queries for the other web page as well.

Next we describe how to measure the similarity of two co-visited web pages using the click-through information. All possible pairs and their frequency are calculated from all the sessions. Precisely, the number of visit times of a web page $d_i$, denoted as $visited(d_i)$, refers to the number of the sessions that the web page $d$ is visited by all the related queries. The number of co-visited times of a two web pages pair $(d_i, d_j)$, denoted $visited(d_i, d_j)$, is defined in a similar way.

With the above definitions, the similarity $S$ between two web pages $d_i$ and $d_j$ based on the co-visited relationship can be computed as:

$$S(d_i, d_j) = \frac{visited(d_i, d_j)}{visited(d_i) + visited(d_j) - visited(d_i, d_j)} \quad (2)$$

The measure is scaled to [0, 1].

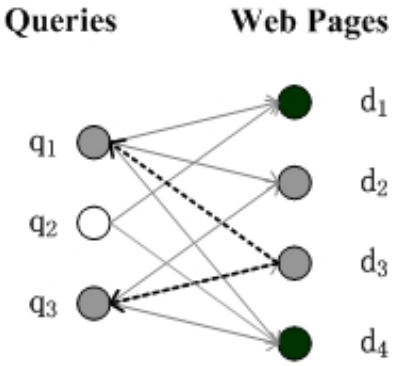


**Figure 2. Co-visited method**

For example, if we assume that each web page is visited by the queries only once in the Figure 2, we can apply the above formula to compute the similarity of any two web pages. The result is shown as follows:

$$S(d_2, d_3) = \frac{1}{2+1-1} = 0.5$$

$$S(d_3, d_4) = \frac{1}{1+3-1} = 0.33$$

If the similarity value between two web pages is greater than a minimum threshold $\sigma$, the two web pages are treated as similar. So if $\sigma$ is equal to 0.4, the web pages $d_2$ and $d_3$ are similar to each other, and the web pages $d_3$ and $d_4$ is dissimilar by definition. Furthermore, if $\sigma$ is set to 1, which means that two web pages are clicked by exact the same queries, this algorithm is the same as the naïve method; if $\sigma$ is set to 0, which means that any two pages

that have one common query are similar to each other. Later experiments will show that the precision of queries associated with a given page is highest when $\sigma$ is equal to 0.3. All the similar pages of a given page $d$ is denoted as $Sim(d)$.

Now, we describe the expansion method to generate the metadata of a given web page: the associated queries are added to the given web page, we define a function to measure the weight of the expansion query to the given web page as the following:

$$W_{ij} = W(d_i, q_j) = \sum_{k \in Sim(d_i)} S(d_i, d_k) \times W(d_k, q_j) \qquad (3)$$

Thus $W_{i1} \cdot q_1 + W_{i2} \cdot q_2 + ... + W_{im} \cdot q_m$ is taken as the additional metadata for web page $d_i$.

## 3.4 Iterative Algorithm (IA)

From the analysis of the co-visited based method, we can see that the co-visited method only considers similarity computing from the side of the web pages, while discarding the similarity of the queries. As a result, the similarity of any two web pages is not really precise. Another problem is the sparseness of the relationship between a query and web pages --- the average number of queries to a web page is 1.5. This makes it very hard for co-visited method to work well since the co-visited method only works on the dense data. Based on above discussion, we propose an iterative algorithm in which the similarity score could be flowed from the similar queries to the associated web pages, and vice versa. As the basic case, we consider an object maximally similar to itself, to which we can assign a similarity score of 1.

Before we go into details of our similarity algorithm, let us briefly walk through an example that illustrates the similarity flowing procedure. We now consider an example, where a bipartite graph is constructed as shown in Figure 3.

Clearly, queries $q_1$ and $q_2$ in $Q$ are similar because they link to the same web page $d_2$ in $D$. Web pages $d_1$ and $d_2$ are similar since they are linked by the same query $q_1$ in $Q$, while web pages $d_2$ and $d_3$ are similar for the same reason. Moreover the similarity between web page $d_1$ and $d_3$ are propagated because queries $q_1$ and $q_2$ in $Q$ are similar. The procedure is computed iteratively until the similarities among the objects reach a fixed point.

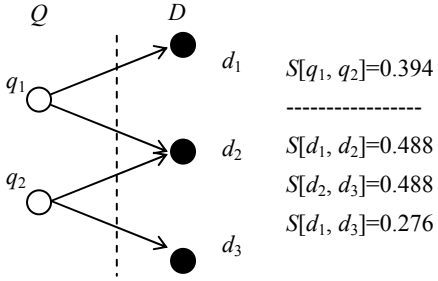


**Figure 3. Example of our iterative algorithm**

Here, similarity of queries and similarity of web pages are mutually reinforcing notions:

- Web pages are similar if they are visited by similar queries.
- Queries are similar if they visit similar web pages.

Let us denote the similarity of the two queries $q_s$ and $q_t$ in $Q$ by $S_Q[q_s, q_t] \in [0,1]$ and the similarity of the two Web pages $d_s$ and $d_t$ in $D$ by $S_D[d_s, d_t] \in [0, 1]$. The mutually recursive equations that formalize these notions are given as the following:

We define an equation for $S_Q[q_s, q_t]$ as follows: If $q_s = q_t$ then $S_Q[q_s, q_t]$ is defined to be 1. Otherwise, the equation of similarity of two queries is written as:

$$S_Q[q_s, q_t] = \frac{C}{|O(q_s)||O(q_t)|} \sum_{i=1}^{|O(q_s)||O(q_t)|} \sum_{j=1} S_D[O^i(q_s), O^j(q_t)] \qquad (4)$$

where $C$ is decay factor, $O(q)$ is the set of the web pages where query $q$ clicked and the $i^{th}$ individual in the set is denoted as $O^i(q)$.

We also can infer the similarity of two Web pages analogously: If $d_s = d_t$ then $S_D[d_s, d_t]=1$. Otherwise, the equation of similarity of two Web pages is written as

$$S_D[d_s, d_t] = \frac{C}{|I(d_s)||I(d_t)|} \sum_{i=1}^{|I(d_s)||I(d_t)|} \sum_{j=1} S_Q[I^i(d_s), I^j(d_t)] \qquad (5)$$

In this paper, we set the decay factor $C$ as 0.7, $I(d)$ is the set of the queries that clicked the web page $d$ and the $i^{th}$ individual in the set is denoted as $I^i(d)$. The similarity result of the example is shown in Figure 3.

As we have said above, this equation is recursive, and the similarity of the objects can be propagated and spread at next recursion. We start with $S^0$:

$$S^0(d_s, d_t) = \begin{cases} 0 & (d_s \neq d_t) \\ 1 & (d_s = d_t) \end{cases} \qquad (6)$$

The computation $S^{i+1}$ is from $S^i$. The values $S^k$ are non-decreasing as $k$ increases and will converge eventually.

After computing the similarity between any two objects (e.g. queries or web pages), we use the results to find queries that are associated a given web page.

Given a minimum similarity threshold $\delta$, any pairs of web pages with similarity below $\delta$, is filtered. All the similar pages of a given page $d_i$ is denoted as $Sim(d_i)$. To a given web page $d_i$, we compute the $i^{th}$ raw of $W_{ij}$, then $W_{i1} \cdot q_1 + W_{i2} \cdot q_2 + ... + W_{im} \cdot q_m$ is taken as additional metadata for the web page $d_i$.

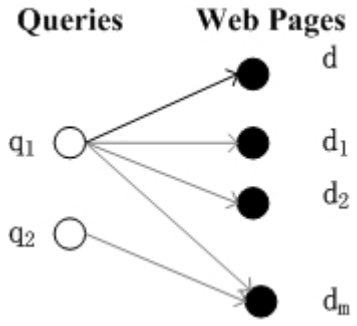


**Figure 4. New web page scenario**

It is true that many new web pages will emerge on the web everyday. And there are also many new queries submitted to search engines. It is very necessary for a search engine to quickly adapt to such changes and provides user more relevant information. Unlike the co-visited method, the iterative algorithm could effectively analyze such new incoming queries and web pages.

For example, in Figure 4, a new page $d$ is only clicked by a query $q_1$, which is associated with many old pages $d_1, d_2, \ldots, d_k$.

Under the co-visited method, any of $d_1, d_2, \ldots, d_k$ appears equally similar to $d$. If any page of $d_1, d_2, \ldots, d_k$ is visited by other queries, the co-visited similarity score between $d$ and each of $d_1, d_2, \ldots, d_k$ will be even low. In the iterative algorithm, the queries associated with the pages $d_1, d_2, \ldots, d_k$ are also taken into account, and they affect the similarity scores between $d$ and each of $d_1, d_2, \ldots, d_k$. Those pages which are clicked by other queries similar to $q_1$ will have higher similarity to $d$. In Figure 4, $d_k$ is shown to be a better match for $d$ than $d_1$, since $d_k$ is clicked by $q_2$ which is similar to $q_1$.

## 3.5 Improving Web Search based Query Metadata

To improve the web search leveraging on the log-based query metadata, we propose two kinds of fusion methods: data fusionand result fusion.

The data fusion is to combine the query metadata with the original document content as a virtual document. The OKAPI system [18] is then used to index all the virtual documents in the collection.

The result fusion is to index the query metadata and the content separately, and then linear combination is used to re-rank the search result. The re-ranking method uses a linear combination of content-based similarity score and the metadata similarity score:

$$Score = \alpha \times SimContent + (1- \alpha) \times SimMetadata \quad (\alpha \in [0, 1]) \quad (7)$$

where *SimContent* is the content-based similarity between web-pages and query words, and *SimMetadata* is the content-based similarity between metadata and query words.

We will test both of these methods in our experiments.

# 4. EXPERIMENTS

In this section, we introduce the experimental data set, our evaluation metrics, and the experimental result based on those metrics.

## 4.1 Data Set

In order to study the effectiveness of the proposed iterative algorithm for optimizing search performance, our experiments are conducted on a real click-through data which is extracted from the log of the MSN search engine [13] in August, 2003. It contains about 1.2 million query requests recorded over three hours. The log we obtained is already processed into a predefined format, i.e. each query request is associated with one click web page. We called it "query session", which can be defined as follows:

*Query Session: = query text [clicked Web page *]*

The average query length is about 2.8 words. A small sample of the raw data is shown in Table 1. Before doing experiment, some preprocessing steps are applied to queries and web pages in the raw log. All queries are converted into lower-case, stemmed by the Porter algorithm; stop words are removed. The query sessions sharing a same query are merged into a large query session, with the frequencies being summed up. After preprocessing, the log contains 13,894,155 sessions, 507,041 pages and 862,464 queries. We use a crawler to download the content of all web pages contained in this log. After downloading the pages, Okapi system [18] is used to index the full text using BM25 formula.

**Table 1. A sample of the raw MSN query click-through data**

| Query | Clicked Web Page |
|---|---|
| University of Arizona | www.hsu.edu/faculty/dailyc/sife.html |
| maps | www.mapquest.com |
| www.teen+titan.com | www.cartoonnetwork.com/titans/ |
| www.ikea.com | www.ikea-usa.com |
| cokemusic.com | www.cokemusic.com |
| motel6 | www.motel6.com |
| pampered chef | www.pamperedchef.com |
| WEATHER | www.nws.noaa.gov |

Click-through data is very sparse, because web users are more likely to click top n (typically 10) web pages returned by a search engine. According to the statistics from the MSN click-through data, the average query frequency for a web page is 1.5. Furthermore, the distribution satisfied the Power Law. As shown in Figure 5, most pages are only associated with few queries, while only a few pages are associated with a large number of queries. So it is necessary to exploit the query log data and to mine out the latent association relationship between the web pages and the queries.

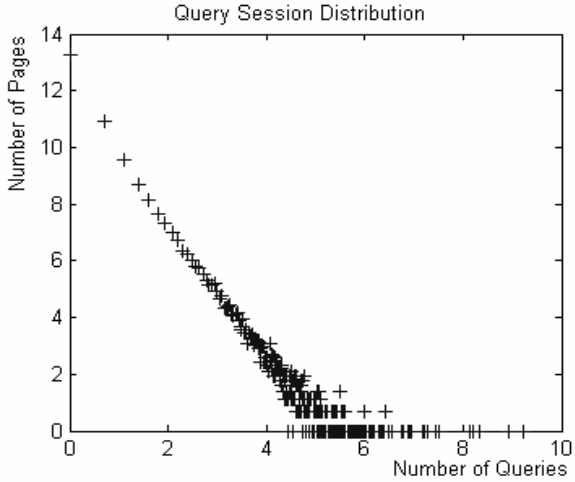


**Figure 5. Query session distribution (Logarithm on X and Y)**

The relevancy of queries to the contents of web pages is evaluated from a user perspective. We randomly select three subsets which contain 900 query sessions in total. Ten volunteer graduated students are chosen as our evaluation subjects. They are asked to evaluate whether the queries are relevant to the web pages according to the content of the pages. As shown in Table 2 about 82.7% of queries in average are relevant to the contents of the Web pages.

**Table 2. Relevance between the Queries and Web Page**

| Subset | Session | Relevant | Ratio |
|---|---|---|---|
| 1 | 300 | 247 | 0.82 |
| 2 | 300 | 262 | 0.87 |
| 3 | 300 | 228 | 0.76 |
| Average | | | **0.817** |

## 4.2 Evaluation Criteria

The *Precision* in IR is applied to measure the performance of our proposed algorithm. Given a query $Q$, let $R$ be the set of the relevant pages to the query and $|R|$ be the size of the set; let $A$ be the set of top 20 results returned by our system. *Precision* is defined as:

$$Precision = \frac{|R \cap A|}{|A|} \quad (8)$$

In order to evaluate our method effectively, we also propose a new evaluation metric *Authority*. Given a query, we ask the ten volunteers to identify top 10 authoritative pages according to their own judgments. The set of 10 authoritative web-pages is denoted by $M$ and the set of top 10 results returned by search engines is denoted by $N$.

$$Authority = \frac{|M \cap N|}{|M|} \quad (9)$$

*Precision* measures the degree to which the algorithm produces an accurate result; while *Authority* measures the ability of the algorithm to produce pages that are most likely to be visited by users. *Authority* measurement is more relevant to users' degree of satisfactory on the performance of a web search engine.

## 4.3 Performance

We fixed several parameters for the rest experiments. i.e. minimum similar threshold as 0.3, using result fusion to measure the performance, the weight of the content as 0.4,and iterative times as 10. These parameters are determined based on an extensive experiment which will be discussed in section 4.5.

First, the volunteers were asked to evaluate the *Precision* and *Authority* of search results for 10 queries (which are *MapQuest*, *Baby Care*, *Cribs*, *windows*, *Google*, *Search Engine*, *Airline United*, *Auto Trader*, *Cartoon Network,* and *Universal studios*). The final relevance judgment for each document is decided by majority votes. Figure 6 shows the comparison of our approach -- iterative algorithm (IA) with content based search (CB), naive method (NM), and co-visited method (CVM).

From Figure 6 and Figure 7, we found that the performance of the full text search technique is poor, demonstrating the gap between the document space and the query space. When click-through data is introduced, the search performance is improved. The more click-through data is introduced, the higher is the performance of search.

In Figure 6 and Figure 7, Co-visited method has a medium performance in all the algorithms. Co-visited method outperforms the naive method because it takes the queries of the similar pages as its virtual queries. However some noise data is also introduced into the metadata of web pages.

Our IA method is able to constrain the noise data and also improve the precision on the similarity of the web pages. As a result, our method achieves the best performance.

Furthermore, our IA method can significantly accelerate to bridge the gap between the query space and document space by adding extra relevant metadata to the web pages.

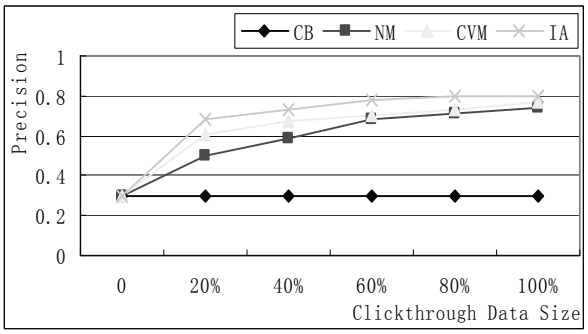


**Figure 6.The precision on different data size**

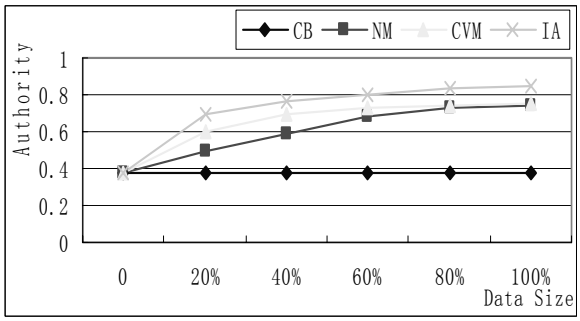


**Figure 7. The authority on different data sizes**

The next experiment is to evaluate *Precision* and *Authority* of search results for the above 10 queries. The final judgment for each document is also decided by majority votes. Figure 8 and Figure 9 shows the comparison results. The right-most label "Avg" stands for the average authority value for the 10 queries. As shown in Figure 8, our algorithm outperforms the other four algorithms on top 20 precision. Relatively, the average improvement over the full text is 157%, Naive method 17%, and Co-Visited method 17%. As shown in Figure 9 , our algorithm outperforms the other four algorithms on top 10 authorities. Relatively, the average improvement over the full text is 123%, Naive method 16%, and Co-Visited method 15%.

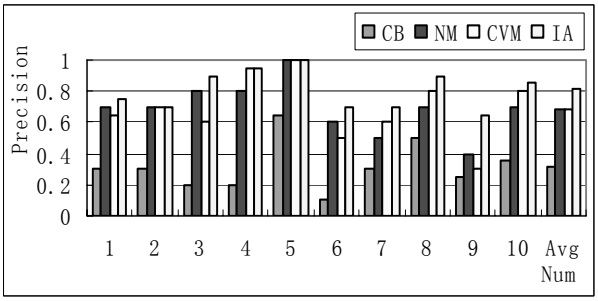


**Figure 8. The precision on different methods**

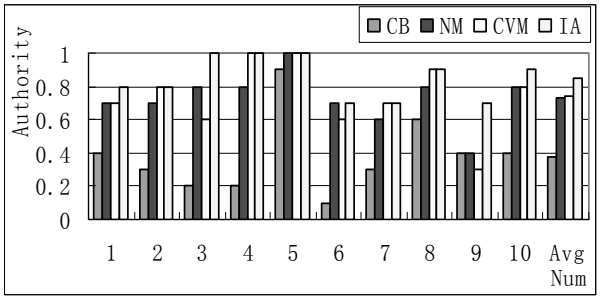


**Figure 9. The authority on different methods**

With metadata, the authority of the Web pages also is improved. This implies that most of the users would like to select the authority web pages associated with the queries.

## 4.4 Case Study

Table 3 shows the URLs contained in the results of co-visited method but not contained in the results of the naive method. From the table, we could find that such results are not so relevant with the query "Cribs". While these pages are co-visited with the pages that have a query "Cribs". From the Table 3, many noise metadata are created by the co-visited method.

**Table 3. Results of co-visited based method**

| URL | Topic |
|---|---|
| http://www.citikids.com/ | The Complete Children's Store |
| http://www.richardsonbrothers.com/ | furniture |
| http://www.dftoys.com/ | Toys |
| http://www.apastyle.org/stylehelper/ | American Psychological Association |
| http://www.millenniumbaby.com/ | Millennium baby |

Table 4 shows the URLs contained in the results of IA method but not contained in the results of the naive method and Co-visited method. From the table, we can find that these URLs are relevant to the query "Cribs", because they are about baby bedding. With the reinforcement of the both sides, the precision of the similarity between the web pages will be improved.

**Table 4. Results of IA based method**

| URL | Topic |
|---|---|
| http://www.babysupermall.com/main/browse/crib-bedding-sets.html | Crib bedding |
| http://www.sears.com/sr/entry.jsp?keyword=Baby+Bedding&sid=I0004607410000400085 | Baby bedding |
| http://www.best-deals-baby-shopping.com/baby-bedding.html | Baby bedding |
| http://www.cheap-baby-stuff.com/toddler-bed-bedding-sets.html | Baby bedding |
| http://www.kids--bedding.com/ | Baby bedding |

## 4.5 Parameters Selection

As we mentioned, several parameters are used in the experiments, such as minimum co-visited threshold, the weight of linear combination, using result fusion and the iterative times of the IA. Here we provide experiments for setting those parameters.

The density of relationships between two types of objects has significant impact on the precision of similarity calculation. In Figure 10, we empirically analyze the precision of finding the similar queries, given different interrelationship density between two types of objects. In this experiment, we randomly select 10%, 30%, 50%, 70% and 90% of the click-through data to represent different degree of how tightly objects are interrelated.

The results show that the degree of how tightly the objects are interrelated with each other has significant impact on the precision of similarity measurement. When objects become more strongly interrelated, the precision of the similarity measure would be improved.

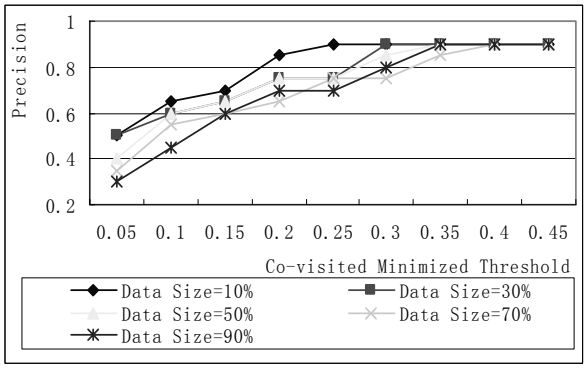


**Figure 10. Precision on different threshold**

As stated previously in Section 3.3, the similar web pages of a given web page are generated by the restriction of the minimum co-visited similarity. The higher the co-visited similarity of two Web pages, the higher the probability of the two pages is similar. As shown in Figure 11, the precision monotonously increases as the minimum co-visited similarity increases where the δ increase from 0 to 0.3. When the threshold δ=0.3, the precision is nearly the highest. A larger threshold will not lead to further increment of the precision. So we choose the minimum co-visited threshold as 0.3.

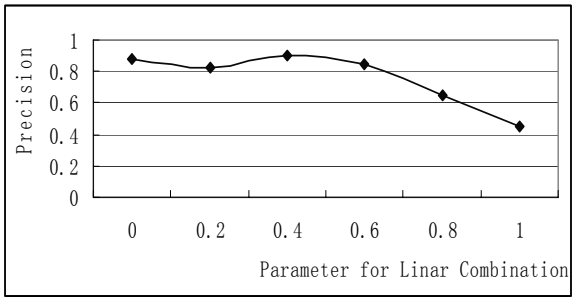


**Figure 11. The precision on different parameters**

In order to measure the weight between the content and the click-through data, we tune the parameter of $\alpha$ (the weight for the content) and $\beta$ (the weight for the metadata from click-through data) from 0 to 1. Since $\alpha + \beta = 1$, we only change the $\alpha$ in our experiment. The experimental results on 10 selected queries are shown in Figure 11. We found that the precision is improved while introducing some content. The system achieves the best

precision when $\alpha$=0.4 and $\beta$=0.6. If we continue to introduce more content into consideration, the precision drop down since there are too many noises embedded in the content.

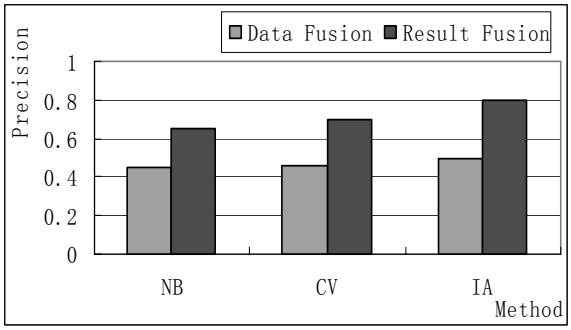


**Figure 12. The precision of different fusion method**

Figure 12 compares the results of two different fusion methods: data fusion and result fusion. Clearly, the results in Figure 14 show that the result fusion achieves higher precision than the data fusion. By assigning the metadata into the content of the web page, the effect of the metadata is so little in the VSM model. Result fusion could achieve high performance by linear combination. The weight of the metadata could be improved by selecting the parameters.

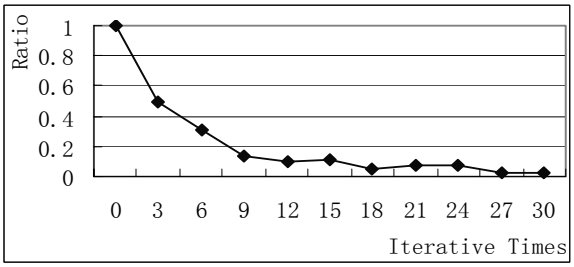


**Figure 13. Convergence of the IA algorithm**

The convergence curve of several algorithms is shown in Figure 13. The difference of the similarity between the consecutive iterations is computed using the norm function. The vertical axis denotes the ratio between the difference of the current iteration and the difference at the previous iteration. The figure shows the convergence of this algorithm after 30 iterations, and shows a strong tendency toward zero.

# 5. CONCLUSION

The gap between the query space and the document space may be filled in when more accurate content or metadata for web pages is introduced. Click-through data is supposed to add more accurate content (metadata) for web pages, thus improve the relevance measurement. However, click-through data is often noisy, incomplete, and sparse, and new documents and new queries often emerge. In this paper, we propose a novel iterative reinforced algorithm to utilize click-through data. The algorithm could fully explore the interrelations between heterogeneous data objects, and effectively find the virtual associated queries for web pages, thus deal with the above issues. Experiment results on a large set of MSN click-through data show a significant improvement of search performance.

Our work can be extended in several directions. For our problem, the content of the queries and the web pages is not considered to calculate the similarity of the web pages, so the future work should take the content into account to measure the similarity of the Web pages. Another problem is that we now only consider the web pages which have been clicked by at least one query, there exist lots of web pages that don't have the click-through data. In the next step, we want to integrate the click-through data, hyperlink structure, anchor text and the content of the web pages together and to measure the similarity of the Web pages.

# 6. REFERENCES


[1] Bernard J. Jansen , Amanda Spink , Judy Bateman , and Tefko Saracevic. Real life information retrieval: a study of user queries on the Web, *ACM SIGIR Forum*, v.32 n.1, p.5-17, Spring 1998.

[2] Brian D.D., David, G.D., and David B.L. Finding Relevant Website Queries, in *Proceedings of the Twelfth International World Wide Web Conference*, 2003.

[3] Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang. Relevant term suggestion in interactive web search based on contextual information in query session logs. *JASIST* 54(7): 638-649,2003.

[4] Cui H., Wen J.R., Nie J.Y., and Ma W.Y., Query Expansion by Mining User Logs, *IEEE Transaction on Knowledge and Data Engineering*, Vol. 15, No. 4, July/August 2003.

[5] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 407-415, 2000.

[6] Funas, G.W., Landauer,T.K., Gomez,L.M. and Dumais, S.T. 1987. The vocabulary problem in human-system communication. *Communications of the ACM* 20,11, Pages 946-971, Nov.1987.

[7] G. Jeh and J. Widom. SimRank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 2002.

[8] G. Salton and C. Buckley. On the use of spreading activation methods in automatic information, in *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, p.147-160, Grenoble, France, May 1988.

[9] H. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24:265–269, 1973.

[10] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang. Clustering user queries of a search engine. In *Proceedings of the Tenth International World Wide Web Conference*, Hong Kong, May 2001.

[11] Joachims T. Optimizing Search Engine using Clickthrough Data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, 2002.

[12] M. M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14:10–25, 1963.

[13] MSN Search Engine, http://www.msn.com.

[14] Nick C., David H., and Stephen R. Effective Site Finding using Link Anchor Information, *ACM SIGIR'01*, New Orleans, 2001.

[15] Nicolas J. Belkin, Helping people find what they don't know, *Communications of the ACM*, v.43 n.8, p.58-61, Aug. 2000.

[16] Porter, M. An algorithm for suffix stripping. Program, Vol. 14(3), pp. 130137, 1980.

[17] R. Baeza-Yates and B.Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[18] Robertson, S.E. et al. Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference(TREC-3)*, 109-126, 1995.

[19] R. R. Larson. Bibliometrics of the World-Wide Web: An exploratory analysis of the intellectual structure of cyberspace. In *Proceedings of the Annual Meeting of the American Society for Information Science*, Baltimore, Maryland, October 1996.

[20] S. Brin and L. Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, in *Proceedings of the 7th international World Wide Web Conference*. Vol.7, 1998.

[21] S. Chakrabarti et al., Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text, in: *Proceedings of the 7th International World Wide Web Conference*, 1998.

[22] Thijs W., Wessel K., and Djoerd H., Retrieving Web Pages using Content, Links, URLs and Anchors, *TREC10*, 2002.

[23] V. V. Raghavan and H. Sever. On the reuse of past optimal queries. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 344-350, Seattle, WA, July 1995.