Link Fusion: A Unified Link Analysis Framework for Multi-Type Interrelated Data Objects[I]

Wensi Xi[1], Benyu Zhang[2], Zheng Chen[2], Yizhou Lu[3], Shuicheng Yan[3], Wei-Ying Ma[2], Edward A. Fox[1]

[1]Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24061, U.S.A.

{xwensi, fox}@vt.edu

[2]Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, P.R. China

{byzhang, zhengc, wyma}@microsoft.com

[3]School of Mathematical Sciences, Peking University, Beijing 100871, P.R. China

luyizhou@pku.edu.cn scyan@math.pku.edu.cn

ABSTRACT

Web link analysis has proven to be a significant enhancement for quality based web search. Most existing links can be classified into two categories: intra-type links (e.g., web hyperlinks), which represent the relationship of data objects within a homogeneous data type (web pages), and inter-type links (e.g., user browsing log) which represent the relationship of data objects across different data types (users and web pages). Unfortunately, most link analysis research only considers one type of link. In this paper, we propose a unified link analysis framework, called "link fusion", which considers both the inter- and intra- type link structure among multiple-type inter-related data objects and brings order to objects in each data type at the same time. The PageRank and HITS algorithms are shown to be special cases of our unified link analysis framework. Experiments on an instantiation of the framework that makes use of the user data and web pages extracted from a proxy log show that our proposed algorithm could improve the search effectiveness over the HITS and DirectHit algorithms by 24.6% and 38.2% respectively.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information search and retrieval; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms

Algorithms, Experimentation

Keywords

Link fusion, Link analysis algorithms, Information retrieval, Data fusion.

1. INTRODUCTION

The World Wide Web is estimated to contain 3-5 billion web pages nowadays and is still growing at a rate of 10 million per day. The content of web pages ranges from dish washer advertisement to the proceedings of the W3C conference. With such huge volume and great variation in contents, finding useful information effectively from the web becomes a very challenging job. Traditional "keyword based" text search engines cannot provide satisfying results to web queries since: (1) Users tend to submit very short, sometime ambiguous queries and they are reluctant to provide feedback information [3]. (2) The quality of web pages varies greatly [6], and users usually prefer high quality pages over low quality pages in the result set returned by the search engine. (3) A non-trivial number of web queries target at finding a "navigational starting point" [9] or "URL of a known-item" [8] on the web. Thus, web pages containing textually "similar" content to the query may not be relevant at all.

Based on the observations above, researchers tried different approaches to improve the effectiveness of web search engines. One of the representative solutions is re-ranking the top retrieved web pages by their importance [1, 11, 17], which is calculated by analyzing the hyperlinks among web pages. Hyperlink analysis (such as [1, 3-7, 17-19]) has been shown to achieve much better performance than full text search, in production systems.

According to their types, links can be classified into two categories: intra-type links, which represent the relationship of data objects within a homogeneous data space, and inter-type links, which represent the relationship of data objects between heterogeneous data spaces. Most current web link analysis research only analyzes the hyperlinks within web pages, which can be considered as a homogeneous data space. But in the real world, the web pages will often interact with other types of objects, such as users and the queries. In this paper we try to deal with these inter-relationships by expanding the link analysis to combine both inter-type link analysis and intra-type link analysis, and thereby improve web search performance. In Figure 1, we show an example of inter and intra type links by analyzing the relationship of three related data types in the web environment: user, web page, and query.

Users and the queries they submit, plus the web pages they browse, form three homogeneous data spaces. They are correlated when a user submits queries, a user browses web pages, and a query references web pages. The three operations: submit, browse, and reference, involve inter-type links across these data spaces. The hyper-links within web pages, content-based

WWW 2004, May 17-22, 2004, New York, NY USA.

ACM 1-58113-844-X/04/0005

[I] This research work is done at Microsoft Research Asia.

similarity of queries, and social structure of users are intra-type relationships within each space. It is obvious that when analyzing the attributes of web pages, not only the hyper-links between them, but also the users who browse them and the queries that reference them can play important roles.

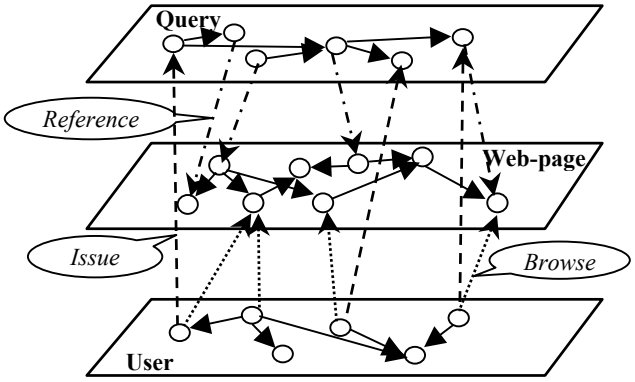

Figure 1: An example of multi-type interrelated data spaces

Most existing web related research fits into the web multi-space model we described in Figure 1. For example, web search [4, 17] uses the web page space and hyperlinks within the space; collaborate filtering [14] uses the document (web page) space, the user space, and the browsing relationship in-between; web query clustering [22] uses the web page space, query space, and reference relationship in-between. Unfortunately, most of these works only consider one type of link/relationship when analyzing the links/relationships of objects, and they can be classified into intra-type link analysis and inter-type link analysis regarding the type of links they use. In intra-type link analysis, the attribute of a data object is directly reinforced by the same attribute of other data objects in the same data space. For example, in Google's PageRank algorithm [4], the "popularity" attributes of web pages are reinforcing each other via the hyper-link structure within them. In inter-type link analysis, the attribute of one type of data objects is reinforced by attributes of data objects from other data spaces. (Examples of inter-type link analysis will be given in Section 3.) Hyperlink analysis reflects the attributes of web pages from the editor's view. The assumption of the hyperlink analysis is that users agree with the editor/author of the web pages in terms of the link structure. It may not work well when a user's perception of a web page differs from that of the authors/editors. Another example of inter-type link analysis, the DirectHit algorithm [11], well captures the web user's view of the web pages from their interactions with the Web. DirectHit utilizes the inter-type links provide by end-users for web search assuming that the more frequently users visit a web page the more important the web page is.

It is natural to ask: Is it possible to combine the process of intra-type link analysis for the same data type and inter-type links across different data types together to improve the process of understanding the organizational relationship of data objects and finding the correct order of data objects regarding different attributes in multiple data types? Intuitively, a simple way is to calculate the data object attributes using inter-type and intra-type link analysis individually, and then combine the results together. However, this solution does not fully utilize the fact that inter- and intra- type links may reinforce the attribute of a data object at the same time. Hence, a unified framework for link analysis is proposed in this paper. The assumption is that the attribute of a data type is influenced not only by the intra-links of its own type but also influenced by the inter-links from other attributes of other different data types. Furthermore, different attributes of different data types can reinforce each other. The problem of leveraging link structures within and across different data types to gain more understanding of the organizational structure and attribute order of objects within each data type can be referred to as the "**Link Fusion**" problem. This name is borrowed from the concept of "**Data Fusion**" in information retrieval where multiple sources of evidences are combined in order to improve the prediction of the relevanc of documents to a query. Experiments on an instantiation of the framework that makes use of users and web pages from a proxy log show that by using our approach, the search precision is improved by 24.6% and 38.2% compared to the traditional HITS [17] and DirectHit [11] algorithms, respectively.

The rest of this paper is organized as follows. In Section 2, we present related work on current state-of-the-art link structure analysis algorithms. In Section 3, we present the proposed unified link analysis framework for multi-type inter-related data objects, which can support HITS and PageRank, as well as the DirectHit algorithm. Then, we show the experimental results in Section 4. Finally, we conclude in Section 5

2. RELATED WORKS

Research on analyzing link structures to better understand the informational organization within data spaces can be traced back to research on "Social Networks" [13]. A good example comes from the telephone bill graph. By searching connected and isolated components, scientists can estimate the diameter of the whole graph and hunt for each complete sub-graph or "clique", to indicate contacts among people. Another interesting example is the famous sociology phrase "six degree of separation", which means that any pair of people on the earth can get acquaint through no more than six intermediaries. Although proving this is still far from complete, some sub-graphs of human society can be explored easily and thoroughly. For instance, members of an enterprise can form an operation graph. By recognizing the functional relationship of each employee, one can learn structural and relative "importance" of each employee within the organization. The problem of link structure of social networks can be reduced to a graph G = (V, E), where set V refers to people, and set E refers to the relationship among people. Katz [16] tried to measure the "importance" of a node in a graph by calculating the in-degree (both direct and indirect) of that node. Hubbell [15] tried to do the same thing by propagating the "importance" weights on the graph so that the weight of each node achieves "equilibrium".

Researchers from the bibliometrics area claimed that scientific citations could be regarded as a special social network, where journals and papers are the nodes and the citation relationships are edges in the graph. Garfield's famous "impact factor" [12] calculates the importance of a journal by counting the citations the journal received (the in-link) within a fixed amount of time. Pinski and Narin [20] claimed that the importance of a journal is recursively defined as the sum of the importance of all journals that cited it. Based on this hypothesis, they designed the following measure of importance. Consider matrix A is the link matrix in the journal space. A_{ij} denotes the fraction of the number of citations from journal i to journal j. Suppose w_j is the importance value of journal j, their calculation can be represented

as $w_j = \sum_i A_{ij} w_i$ By iteratively calculating the formula above, it leads to $A^T w = w$, where w is the vector of important weights of journals. It is easy to find out that w is the principle eigenvector of A^T. Following the same rationale, Brin and Page [4] design the PageRank algorithm to calculate the importance of web pages in the Web. In addition to Pinski and Narin's algorithm, PageRank simulates a web surfer's behavior on the web. That is, with probability 1-ε, the surfer randomly picks one of the hyperlinks on the current page and jumps to the page it links to; with probability ε, the user "resets" by jumping to a web page picked uniformly and at random from the collection. This defines a Markov chain on the web pages, with the transition matrix $\varepsilon U + (1-\varepsilon)M$, where U is the transition matrix of uniform transition probabilities ($u_{ij} = 1/n$ for all i, j). The vector of PageRank scores w is then defined to be the stationary distribution satisfying $(\varepsilon U + (1-\varepsilon)M)^T w = w$. Adding the random surfer model can prevent the "sink node problem" in the PageRank calculation.

Kleinberg [17] claimed that web pages and scientific documents are governed by different principles. Journals have approximately the same purpose, and highly authoritative journals always refer to other authoritative journals. The World Wide Web, however, is heterogeneous, with different pages serving different roles. Authoritative web pages do not necessary link to other authoritative pages, thus Pinski and Narin's hypothesis for scientific literature does not hold in the web. Based on his observations, Kleinberg divides the notion of "importance" of web pages into two related attributes: "Hub" (measured by the "authority" score of other pages that a page links to), and "Authority" (measured by the "hub" score of the pages that link to the page). Different from the PageRank algorithm which calculates the importance of web pages independently from the search query, Kleinberg presented his Hyperlinked-Induced Topic Search (HITS) algorithm as following: (1) Use an ordinary search engine to search the query and form the root set as the starting point; (2) Get the base set by adding pages pointing to or pointed at root pages; (3) Count the authority and hub weights of each page in the base set with an iterative algorithm: for each page, let $a(p)$ and $h(p)$ denote its authority attribute weight and hub attribute weight. The two attributes can be calculated as:

$$a(p) = \sum_{q \to p} h(q) \quad and \quad h(p) = \sum_{p \to q} a(q)$$

Let A denote the adjacency matrix of the base set: a_{ij}=1 if page i has a link to page j, and 0 otherwise. Vectors a and h correspond to the authority and hub scores of all pages in the base set, hence, a=A$^T h$ and h=Aa. It is easy to show that a and h are eigenvectors of matrices A^{T}A and AA^T. The search system [1] developed using the HITS algorithm achieves comparable performance with "Yahoo!", which maintains a manual compilation of net resources. Many researchers have extended the HITS algorithms to improve its efficiency. Chakrabarti et al. [5, 6] used texts that surround hyperlinks in source web pages to help express the content of destination web pages. They also reduce weight factors of hyperlinks from the same domain to avoid a single website dominating the results of HITS. Lempel and Morgan [18] extend HITS by replacing Kleinberg's Mutual Reinforcement approach with a new stochastic approach (SALSA), which can be considered as a weighted link structure analysis of the web sub-graph. In their work, they identify the *Tightly Knit Community (TKC)* Effect in the web communities that hampers the HITS algorithm to identify meaningful authorities, and they show that SALSA is less vulnerable to the *TKC* effect than the HITS algorithm. Ng et al. [19] presented randomized HITS and subspace HITS algorithms to enhance the stability of the basic HITS. The former imitates a random walk on web pages and defines the authority/hub weight as a chance of visiting that page in time step t (t is large enough). The latter uses the first k eigenvectors instead of the entire matrix A^{T}A to count the authority values. Cohn et al. [7] introduced a probabilistic factor into HITS and applied the EM model. All these show that the authority idea has great potential in web applications.

Inter-type links (links that connect different types of data objects) represent relationships of different domains. Researchers also analyzed this kind of link to find out whether it can help improve the link analysis of the data objects within the same data type. For example, DirectHit [11] harnesses the web pages visited by millions of daily Internet searchers to provide more relevant and better-organized search results. Based on the assumption that the most relevant pages of a topic are those most visited, DirectHit's ranking algorithm is used by Lycos, Hotbot, MSN, Infospace, About.com, and roughly 20 other search engines. Miller [18] proposed a modified HITS algorithm, which also utilizes the users' behavior on the web to improve the calculation of hub and authority scores. In his algorithm, the adjacency matrix A is modified and the value of a_{ij} in A is increased whenever a user travels from page i to page j (information obtained by analyzing web-site access logs). Although Miller uses links from two different spaces (user and web space), he only converted inter-type links (links between users and web-pages) to intra-type links (links within web-pages) to enhance the link analysis for web pages. The users' importance is ignored in this algorithm.

Most recently, Davison [10] analyzed multiple term document relationships by expanding the traditional document-term matrix into a matrix with term-term and doc-doc sub-matrices in the diagonal direction and term-doc and doc-term sub-matrices in the anti-diagonal direction. The term-term sub-matrix represents term relationships (e.g., term similarity), and the doc-doc sub-matrix represents document relationships (e.g., link matrix for web pages). He proposed that the links of the search objects (web-page or terms) in the expanded matrix could be emphasized. With enough emphasis, the principal eigenvector of the extended matrix will have the search object on top with the remaining objects ordered according to their relevance to the search object. Considering that terms and documents each form a different data space, with the doc-term and term-doc matrices representing inter-type links, and the term-term and doc-doc matrices as intra-type links, Davison's proposed research fits our framework very well.

3. THE LINK FUSION ALGORITHM

There are similarities among link analysis in social networks, scientific citations, and hyperlink analysis in the web. The data objects in these examples form one or multiple data spaces of different types. Each data space contains one specific attribute of data. Researchers take advantage of the links/relationships either within each data space (intra-type links) or across different data spaces (inter-type links) to calculate the specific attribute of the objects in each of the data spaces. In this Section, we generalize previous link analysis studies and propose a unified link analysis

framework to calculate the attributes of data objects within multiple data spaces. We call this unified link analysis framework "Link Fusion algorithm".

Suppose we have n different types of objects X_1, X_2… X_n. Each type of data object X_i contains a specific attribute F_i. Data objects within the same type are interrelated with intra-type relationships $R_i \subseteq X_i \times X_i$. Data objects from two different types are related with inter-type relationships $R_{ij} \subseteq X_i \times X_j$ ($i \neq j$). Suppose attributes of different types of data objects are comparable (e.g., similar in nature). We borrow and extend Pinski and Narin's recursive definition of importance [20] and define that the specific attribute of a data object in one data type equals the sum of the attributes of other data objects in the same data space that link to it, plus the sum of other related attributes of data objects in other data spaces and links to it, mathematically as:

$$F_i = F_i R_i + \sum_{j \neq i} F_j R_{ji} \tag{1}$$

For simplicity, we first explain the case that only contains two types of related objects as example to illustrate Eq. (1). We consider two types of objects $X = \{x_1, x_2, \cdots x_m\}$, and $Y = \{y_1, y_2, \cdots y_n\}$ and relationships of R_X, R_Y, R_{XY} and R_{YX}. The adjacency matrices are used to represent the link information. L_X and L_Y stand for the adjacency matrices of link structures within set X and Y, respectively. L_{XY} and L_{YX} stand for the adjacency matrix of links from objects in X to objects in Y and adjacency matrix of links from objects in Y to objects in X respectively. $L_{XY}(i, j) = 1$, if there is a link from node x_i to node y_j, and $L_{XY}(i, j) = 0$ otherwise. Suppose w_x is the attribute vector of objects in X, w_y is the attribute vector of objects in Y, Eq. (1) can be mathematically represented as:

$$\begin{cases} w_y = L_y^T w_y + L_{xy}^T w_x \\ w_x = L_x^T w_x + L_{yx}^T w_y \end{cases} \tag{2}$$

and it can be easily extended into N interrelated data spaces, as shown in Eq. (3)

$$w_M = L_M^T w_M + \sum_{\forall N \neq M} L_{NM}^T w_N \tag{3}$$

There are two issues that need to be considered in Eq. (3):

First, as noted by Bharat and Henzinger [3], mutually reinforcing relationships between objects may give undue weight to objects. Ideally, we would like all the objects to have the same influence on the other objects they connect to. This can be solved by normalizing the binary adjacency matrix in such a way that if an object is connected to n other objects in one adjacency matrix, each object it connects to receives $1/n$ of its attribute value. The random surfer model used in PageRank also can be introduced here to simulate random connection, and avoid sink nodes during the computation.

Second, it is too naïve to assume that attributes from different data spaces are equally important, when used to calculate the attribute of data objects. This can be solved by changing Eq. (2) into a weighted sum of attributes. With the consideration of the two issues above, Eq. (3) can be further improved into Eq. (4):

$$\begin{cases} w_M = \alpha_M \acute{L}_M^T w_M + \beta_{NM} \sum_{\forall N \neq M} \acute{L}_{NM}^T w_N \\ where \\ \alpha_M + \sum_{\forall N \neq M} \beta_{NM} = 1; \;\; \alpha_M > 0 \;\; \beta_{NM} > 0 \;\; ; \\ \acute{L}_M = \varepsilon U + (1-\varepsilon) L_M; \;\; 0 < \varepsilon < 1 \;\; ; \\ \acute{L}_{NM} = \delta_N U + (1-\delta_N) L_{NM}; 0 < \delta_N < 1. \end{cases} \tag{4}$$

In Eq. (4), U is the transition matrix of uniform transition probabilities ($u_{ij}=1/n$ for all i, j; where n is the total number of objects in data space N). δ and ε are smoothing factors used to used to simulate random relationships in matrices L_M and L_{NM}. L_M and L_{NM} are normalized adjacency matrices.

As with the PageRank and HITS algorithms, the attribute value of objects in our framework can be obtained by iteratively calculating Eq. (4) until the result converges. With the definition of Eq. (4), we actually created a unified square matrix A, as shown in Eq. (5), where n is the total number of all involved objects in different data spaces. The unified matrix A has $\acute{L}_M$ on the diagonal direction, and $\acute{L}_{NM}$ in other parts of the unified matrix as illustrated below.

$$A = \begin{vmatrix} \alpha_1 \acute{L}_1 & \beta_{12} \acute{L}_{12} & \cdots & \beta_{1n} \acute{L}_{1n} \\ \beta_{21} \acute{L}_{21} & \alpha_2 \acute{L}_2 & \cdots & \beta_{2n} \acute{L}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{n1} \acute{L}_{n1} & \beta_{n2} \acute{L}_{n2} & \cdots & \alpha_n \acute{L}_n \end{vmatrix} \tag{5}$$

Suppose w is the attribute vector of all the data objects in different data spaces. The proposed iterative approach is actually transforming the vector w using matrix A (e.g., $w=A^T w$). It is relatively easy to find out that when the calculation converges, w is the principle eigenvector of matrix A. The formal mathematical proof of the convergence of the calculation can be found in the appendix. Two problems need to be addressed in the construction of the unified matrix A.

Suppose M and N are two heterogeneous data spaces, when a data object in M has no linking relationship to any data objects in N, we set all the elements in the corresponding row of the sub-matrix $\acute{L}_{NM}^T$ to $1/n$, where n is the total number of objects in data space N. The reason we use random relationship to represent no relationship is to guarantee all the sub-matrix $\acute{L}_{NM}^T$ to be non-zero and to prevent "sink nodes" that may eat up all the weights during the calculation (as suggested by the PageRank algorithm). However, in practice, we can always ignore undesired intra/inter type relationships by setting the corresponding α or β to 0.

In the unified matrix, if $\beta_{MN}>0$, then $\beta_{NM}>0$. This is a necessary condition for the recursive calculation to converge, (as explained in the appendix). However, if the relationship of $\acute{L}_{NM}^T$ is really undesirable for the link analysis, we can always assign a very small positive β_{NM} to reduce the effect of $\acute{L}_{NM}^T$.

By constructing a unified matrix using all the adjacency matrices, we actually construct a unified data space, which contains different types/attributes of data objects. Previous inter-type links are now intra-type links in the unified space, and the "link fusion algorithm" is reduced to link analysis in a single data space.

The proposed framework can be easily used to explain previous works on link analysis.

The PageRank algorithm can be considered as a special case of our unified link analysis framework. In PageRank, there is only one attribute (popularity) of one kind of data object (web pages) being considered. Having α=1 and β=0, (4) reduces to $w = L^T w$ which is the original definition of PageRank algorithm.

The HITS algorithm also can be considered as a special case of the unified link analysis. In the HITS algorithm, two attributes (hub and authority) of the same type of data objects (web pages) are being considered. Hub attributes and authority attributes of the same set of web pages each form a data space; the hyperlinks in-between web pages are now inter-type links that connect the Hub space and Authority space as illustrated in Figure 2.

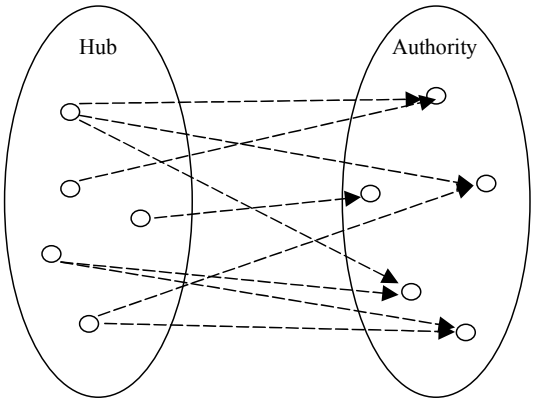

Figure 2: Hub and Authority Spaces in HITS algorithm

Since there are not intra-links in each data space, we set α=0 and β=1 and derive the recursive updating equation from Eq. (4): $w_a = L_{ha}{}^T w_h$ and $w_h = L_{ah}{}^T w_a$, where w_a is the authority value vector, w_h is the hub value vector and L_{ha} L_{ah} are adjacency matrices. Considering the normalization of the adjacency matrices and the introducing of smoothing factor ε, this is by definition the Randomized HITS algorithm [19], which is more robust and stable than the traditional HITS algorithm.

4. EXPERIMENTS

4.1 Experimental Data Set

We use 10 days log from a proxy server at Microsoft to evaluate the effectiveness of our proposed Link Fusion algorithm. The raw proxy logs records user visit information, in which one record corresponds to one HTTP request for a web object from an IP address. In other words, different users from the same IP address are considered as the same user in our experiments. Some heuristic rules (e.g., the words within the hyperlinks, the extension of the filenames, etc.) are applied to filter out the unrelated information, (e.g., ads, images, etc.). Only text pages are reserved in the final dataset, which contains 2,998,821 visit records to 1,773,718 pages by 38,887 users.

4.2 Experimental Approach

Our goal is to improve the end-user's search effectiveness through re-ranking the search results by our proposed Link Fusion algorithm. In order to fit into our framework, we extended the underlying assumption of the HITS algorithm to incorporate the notion of user's "popularity" attribute, and it is defined as below:

- A popular user always look at good hub and good authority pages;
- A good Hub page always points to good Authority pages and is always visited by popular users;
- A good Authority page is always pointed at by good Hub pages and is always visited by popular users too.

The Hub, or Authority attribute of web pages, and the Popularity attribute of users form three different data spaces. These three data spaces are correlated via the hyper links between web pages and user access information from the web proxy log. Their relationships are more clearly illustrated in Figure 3 below.

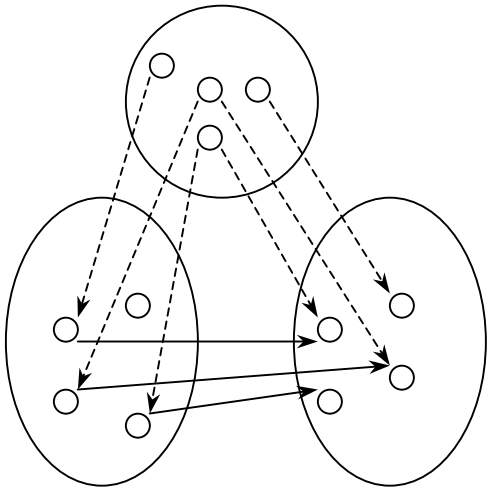

Figure 3: Hub, Authority and User Spaces

We find that the three data spaces and the links in-between them fit our Link Fusion algorithm perfectly. We apply the Link Fusion algorithm from Eq. (5) into this case, and derive the unified adjacent matrix as Eq. (6):

$$A = \begin{vmatrix} \alpha_u L'_u & \beta_{uh} L'_{uh} & \beta_{ua} L'_{ua} \\ \beta_{hu} L'_{hu} & \alpha_h L'_h & \beta_{ha} L'_{ha} \\ \beta_{au} L'_{au} & \beta_{ah} L'_{ah} & \alpha_a L'_a \end{vmatrix} \quad (6)$$

where the sub-scripts a, h and u denote the Authority, Hub, and User space respectively. Since in our case, each data space has no intra-links, we set α_i = 0 ($i = a, h, u$), and we set all the β equal to 0.5. The initial attribute value of each object is set to $1/n$, where n is the total number of objects in the corresponding data space N. Suppose w is the attribute value vector of all the data objects in the three spaces, their final attribute values in w can be obtained by recursively calculating w^{i+1}=A$^T w^i$ (where i is the iteration number) until converge (e.g., $d = \left\| w^{i+1} - w^i \right\|_1$ is smaller than a thresh-hold value)

After generating the link matrix, we calculate the different attributes of web pages and users and use the "Authority" attribute of web pages to re-rank the search results. The detailed approach is described as follows.

We choose 10 sample queries (shown in Figure 1.) to evaluate the Link Fusion algorithm. Detailed experiment steps for each of the sample queries are:

Step 1: Creating the Hub space and Authority space. The Hub space and Authority space are constructed in a way similar to the

HITS algorithm. That is, the query is first sent to a text-based search engine, and the top 200 matching web pages are retained as the root set. Then, the root set is expanded to the base set by its neighborhoods, which are the web pages that either point to or are pointed at by pages in the root set. In this experiment, we set the maximum in-degree of nodes as 50, which is commonly adopted by the previous works [3, 17]. The expanded set of web pages forms the data objects in Hub space and Authority space. Hyperlinks between web pages not on the same web site form the directed links connecting the Hub and Authority space.

Step 2: Creating the User space. After we created the Hub/Authority spaces, we compare the web pages in these spaces with the MSN proxy log data, and extract out the overlapping web pages. The users who browsed these overlapping web pages form the User space, and their browsing activity forms the links from the User space to the Hub/Authority space.

In this experiment we tried to select a set of popular web search queries to test the effectiveness of our Link Fusion algorithm. The queries we selected are shown in Table 1.

Table 1. Queries used in Experiments

ID	Query	PN	LN	UN
1	search engine	3756	406	9317
2	telephone service	3969	320	20406
3	audi car	2438	220	15369
4	baby care	6050	419	7637
5	windows XP	2288	788	16892
6	computer vision	6116	440	10289
7	notebook computer	3071	299	7810
8	online dictionary	5529	324	8255
9	network security	4762	514	14054
10	daily news	3762	367	8387

In Table 1, PN denotes the total number of pages in the formed Hub/Authority space. LN is the number of pages in the Hub/Authority space that were linked by User space (or the number of links from User to Hub or Authority Space). UN denotes the total number of different users in the User space.

Step 3: Calculation. After creating all three data spaces, we assign an initial weight to each data object, as introduced in Section 4.1. and start the recursive calculation on the different attribute in the data spaces according to Eq. (6) until convergence.

Step 4: Evaluation. Finally, we re-rank the top returned documents according to the Authority value we derived from recursive calculation of $w^{i+1}=A^T w^i$. Then we use precision at top 10 documents to compare our results with other algorithms.

4.3 Results Evaluation

In this section, we compare the performance of Link Fusion algorithm with that of the text-based retrieval algorithm, HITS algorithm and DirectHit algorithm. DirectHit algorithm is achieved by re-ranking the top 200 text-based search result according to their number of visits from the user space. For each of the queries listed in Table 1, the union set of top 10 documents returned from the 4 algorithms are pooled together and rated for relevance by 5 volunteers. The final relevance judgment for each <query, document> pair is decided by majority votes (e.g., the pair is relevant only if more than 3 volunteers voted it as relevance). We then computed precision at top 10 documents (p@10) for each of the four algorithms. This measurement is defined as: $p@10 = r/10$, where r is the number of relevant documents in the top 10 pages returned. The comparison of precision for 4 algorithms is shown in Figure 4. The label "avg" is the average *p@10* across the 10 queries.

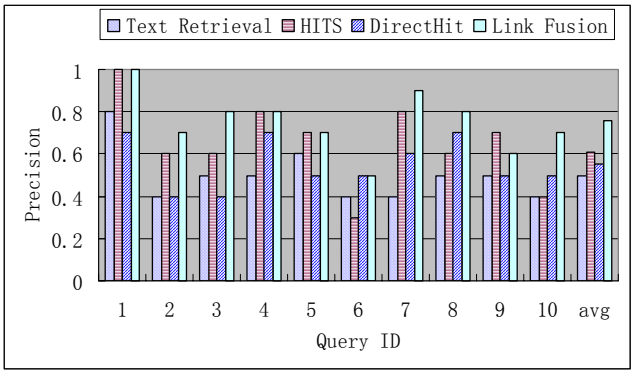

Figure 4. The Precision Comparison of 4 Algorithms

We can see from Figure 4 that our proposed Link Fusion algorithm outperforms the basic HITS algorithm and DirectHit algorithm by 24.6% and 38.2% respectively.

4.4 Case Studies

We give a more detailed analysis of the results by looking at the top URLs returned by three algorithms for several queries. First we show the results of query "audi car" in Table 2. Shaded cells in the table indicate relevant pages. We found that the Link Fusion algorithm had returned 7 out of 9 relevant pages returned by HITS algorithm and DirectHit algorithm combined together, while only keep 2 of the 8 non relevant pages returned by HITS and DirectHit algorithm. Furthermore, Link Fusion algorithm had returned one more relevant page: http://www.s-cars.org/ that has not been found in the top 10 results from either HITS or DirectHit algorithm.

The above observations shows that the Link Fusion algorithm has the capability of keeping the correct results from different link analysis algorithms it combined, while filter out incorrect results returned from these algorithms. Researchers had reported similar findings from data fusion experiments in information retrieval [21]. They claimed that the combined search engine could keep the relevant results returned by different single search algorithms, while filter out those non-relevant results returned by single search algorithms. However, whether the prerequisite conditions for data fusion in information retrieval to be effective are also valid for Link Fusion problem is still left to be explored.

Table 2. Top 10 results for query "audi car"

HITS	DirectHit	Link Fusion
http://www.audiworld.com	http://www.audiusa.com/	http://www.audiusa.com/
http://www.audiusa.com/	http://www.autotrader.com/	http://www.audiworld.com
http://www.audicanada.ca/	http://www.nytimes.com/pages/automobiles/index.html	http://www.uvas.com/
http://www.vindis-cambridge.audi.co.uk/	http://pages.ebay.com/ebaymotors/browse/cars.html	http://www.s-cars.org/
http://pages.ebay.co	http://www.thecarconn	http://communities

m/ebaymotors/browse/cars.html	ection.com/	.msn.co.uk/AudiSCarsUK/pictures
http://www.quattroclubusa.org	http://www.gearheadcafe.com/mags.html	http://www.autotrader.com/
http://www.karquattro.com/	http://www.uvas.com/	http://www.quattroclubusa.org
http://www.porsche.com/	http://communities.msn.co.uk/AudiSCarsUK/pictures	http://www.a4.org/
http://www.vwvortex.com	http://www.autotrader.com/	http://www.vwvortex.com
http://www.nytimes.com/pages/automobiles/index.html	http://www.a4.org/	http://www.thecarconnection.com/

We also found that the binary relevance judgment of a web page we applied in this experiment cannot always fully reflect the "value" of a web page. Although the number of relevant pages returned within top 10 pages by the Link Fusion algorithm (8) is slightly better than that of the HITS algorithm (6), the relevant pages returned by the Link Fusion algorithm (e.g., http://www.a4.org, http://www.s-ars.org) are more authoritative than the relevant pages returned by HITS (e.g. http://www.vindis-cambridge.audi.co.uk). This problem is well represented by another case below.

Table 3. Top 10 results for query "search engine"

HITS	DirectHit	Link Fusion
http://www.google.com/	http://www.google.com/	http://www.google.com/
http://www.ubnmovies.com/	http://dailynews.yahoo.com/fc/Tech/Internet_Portals_and_Search_Engines	http://www.excite.com/
http://www.arelanrecords.com/	http://www.search.com/	http://www.lycos.com/
http://www.novanw.com/	http://www.decideinteractive.com/	http://search.msn.com/
http://www.megaspider.com/	http://www.usaweeend.com/01_issues/010722/010722web.html	http://www.megaspider.com/
http://www.excite.com/	http://www.galaxy.com/	http://www.arelanrecords.com/
http://www.asiaco.com/	http://searchenginwatch.com/awards/	http://www.ubnmovies.com/
http://www.lycos.com/	http://www.bcentral.com/products/si/default.asp	http://www.novanw.com/
http://search.ietf.org/search/brokers/internet-drafts/query.html	http://ixquick.com/	http://www.ixquick.com/
http://www.searchenginewatch.com/	http://www.infospace.com/	http://www.dogpile.com/

Although almost all the pages retrieved by the three algorithms are correct web pages for query "search engine", it is easy to see that the Link Fusion algorithm apparently gives higher ranks to more popular search engines (e.g., http://www.excite.com, http://www.lycos.com) than the other two algorithms. While in HITS and DirectHit algorithms, correct but not very popular search engine web pages (e.g., http://www.ubnmovies.com/, http://www.search.com/) are returned on top. This is because that if a correct web page is returned on top by the Link Fusion algorithm it must be favored by both the web editors (represented by hyperlinks) and the web users (represented by user links) rather than just one of them (e.g., HITS or DirectHit). Thus the Link Fusion algorithm returns more popular results on top than HITS and DirectHit algorithm and also more robust than the other two algorithms.

Below are the results of query "daily news". We can find from this example that the Link Fusion algorithm had both keep the correct results from HITS and DirectHit algorithm and rank the popular correct pages (e.g. http://www.nytimes.com) much higher than the other two algorithms.

Table 4. Top 10 results of query "daily news"

HITS	DirectHit	Link Fusion
http://www.surfinfo.com/html/visreport.html	http://www.msnbc.com/m/hor/horoscope_front.asp	http://www.nytimes.com/
http://dailythong.dhs.org/index.php3	http://daily.webshots.com/	http://sportsillustrated.cnn.com/
http://www.sportspages.com/regions/mw.htm	http://www.thedaily.com/bikini.html	http://encarta.msn.com/
http://www.gossipcentral.com/	http://www.poems.com/today.htm	http://www.thedaily.com/overlook.html
http://www.thedaily.com/overlook.html	http://www.alrai.com/	http://abcnews.go.com/
http://www.webcomics.com/daily.html	http://www.poems.com/	http://www.poems.com/
http://www.guampdn.com/classifieds/index.html	http://www.thedaily.com/overlook.html	http://www.thedaily.washington.edu/
http://www.nytimes.com/	http://cityguide.guampdn.com/fe/index.asp	http://www.gossipcentral.com/
http://www.thedaily.washington.edu/	http://www.thedaily.washington.edu/	http://www.thedaily.com/bikini.html
http://www.smartertimes.com/	http://dailythong.dhs.org/index.php3	http://abcnews.go.com/sections/entertainment/

5. CONCLUSIONS AND FUTURE WORK

In this paper, we first defined two kinds of links among data objects within different data types: intra-type links, which represent the relationship of data objects within a homogeneous data type, and inter-type links, which represent the relationship of data objects between different heterogonous data types. Then, we proposed a unified link analysis framework, called "link fusion", to analyze inter- and intra-type links and to bring order to data objects in different data spaces at the same time.

Next, we evaluated the effectiveness of our proposed link fusion algorithm by applying it into a real world scenario of three data spaces: Hub-page space, Authority-page space, and User space. Experimental results on 10 real world sample queries show that the Link Fusion algorithm achieved 24.6% improvement over the HITS algorithm and 38.2% improvement over the DirectHit algorithm based on the measurement of precision at top 10

documents returned. After a few case studies, we found that the Link Fusion algorithm has the capability of keeping the correct answers returned by each of the link analysis algorithm it combined and trend to return the most popular results on top of its return list. These results support our assumption that the Link Fusion algorithm when used properly can help find the correct order of attributes of data objects within different data spaces.

Although the Link Fusion algorithm seems to be promising according to our preliminary experiments, there are still many issues that need to be explored. For example, in our experiment, we assumed the links from different data spaces are equally important when calculating the attributes of objects across different data spaces. However, this assumption is overly naïve, and it is almost never the case that the links from different data spaces are equally important. It is natural to think: Is there any way to identify the relative importance of links from different spaces automatically? We will explore this problem in our future research works.

6. ACKNOWLEDGMENTS

We thank Dr. Weiguo (Patrick) Fan from Virginia Tech and Li Wang from University of Michigan for their kindly help and suggestions.

7. REFERENCES

[1] The Clever Searching, the Clever project of IBM Almaden Research Center, www.almaden.ibm.com/cs/k53/clever.html.

[2] Berman, A. and Plemmons, R.J. Nonnegative matrices in the mathematical sciences. in Classics in Applied Mathematics, 1994.

[3] Bharat, K. and Henzinger, M.R., Improved algorithms for topic distillation in a hyperlinked environment. in 21st ACM SIGIR International Conference on Research and Development in Information Retrieval, (Melbourne, Australia, 1998), 104-111.

[4] Brin, S. and Page, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems, 30. 107-117.

[5] Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P. and Rajagopalan, S., Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text. in 7th international conference on World Wide Web, (Brisbane, Australia, 1998), 65 – 74.

[6] Chakrabarti, S., Dom, B.E., Kumar, S.R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D. and Kleinberg, J.M. Mining the Web's Link Structure. IEEE Computer, 32 (8). 60-67.

[7] Cohn, D. and Chang, H., Learning to Probabilistically Identify Authoritative Documents. in 17th International Conference on Machine Learning, (Stanford, CA 2000), 167-174.

[8] Craswell, N. and Hawking, D., Overview of the TREC-2002 Web Track. in 11th Text Retrieval Conference, (Gaithersburg, MD,2002).

[9] Craswell, N., Hawking, D. and Robertson, S., Effective Site Finding using Link Anchor Information. in 24th annual international ACM SIGIR conference on Research and development in information retrieval, (New Orleans, LA, 01), 250-257.

[10] Davison, B.D., Toward a unification of text and link analysis. in 26th annual international ACM SIGIR conference on Research and development in information retrieval, (Toronto, Canada, 2003), 367-368.

[11] DirectHit. http://www.directhit.com.

[12] Garfield, E. Citation analysis as a tool in journal evaluation. Science, 178. 471-479.

[13] Hayes, B. Graph Theory in Practice, 2000.

[14] Herlocker, J.L., Konstan, J.A., Borchers, A. and Riedl, J., An algorithmic framework for performing collaborative filtering. in 22nd annual international ACM SIGIR conference on Research and development in information retrieval, (Berkeley, CA 1999), 230-237.

[15] Hubbell, C.H. An input-output approach to clique identification. Sociometry, 28. 377-399.

[16] Katz, L. A new status index derived from sociometric analysis. Psychometrika, 18 (1). 39-42.

[17] Kleinberg, J.M. Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM), 46 (5). 604-632.

[18] Lempel, R., Moran, S. SALSA: the Stochastic Approach for Link-Structure Analysis (TOIS), 19 (2). 131-160.

[19] Miller, J.C., Rae, G., Schaefer, F., Ward, L.A., LoFaro, T. and Farahat, A., Modifications of Kleinberg's HITS algorithm using matrix exponentiation and web log records. in 24th annual international ACM SIGIR conference on Research and development in information retrieval, (New Orleans, LA, 2001), 444-445.

[20] Ng, A.Y., Zheng, A.X. and Jordan, M.I., Stable algorithms for link analysis. in 24th ACM SIGIR International Conference on Research and Development in Information Retrieval, (New Orleans, LA 2001), 258-266.

[21] Pinski, G. and Narin, N. Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. Information Process and Management, 12. 297-312.

[22] Vogt, C.C. and Cottrell, G.W., Predicting the performance of linearly combined IR systems. in 21st annual international ACM SIGIR Conference on Research and Development in Information Retrieval, (Melbourne, Australia, 1998), 190-196.

[23] Wen, J.-R., Nie, J.-Y. and Zhang, H.-J. Query Clustering Using User Logs. ACM Transactions on Information Systems (TOIS), 20 (1). 59-81.

8. APPENDIX

Proof of convergence for the calculation of unified matrix A

In the appendix, we will prove the convergence of iterative calculation method of unified matrix A defined by (5). The proof of convergence would be given, after the proofs of 3 lemmas.

Lemma A: The matrix A defined by (5) is non-negative, row-stochastic.

Proof: Based on (4), we know that matrices L'_M and L'_{NM} are non-negative, row-stochastic. And we also know the constraint of parameter α, β: $\alpha_M + \sum_{\forall N \neq M} \beta_{NM} = 1, \alpha_M > 0, \beta_{NM} > 0$. Thus, each element in matrix A is non-negative, and sum of each row of matrix A is 1. That means the matrix A defined by (5) is a non-negative, row-stochastic matrix. ■

Lemma B: If A defined by (5) is also reducible, there exist a permutation matrix P, such that $PAP^T = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$. Here, A_1 is a non-negative, row-stochastic and irreducible matrix.

Proof: Actually, if A is reducible, there exist a permutation matrix P, such that $PAP^T = \begin{bmatrix} A_1 & 0 \\ B & A_2 \end{bmatrix}$. A_1 is a non-negative, row-stochastic and irreducible matrix.

As metioned in the construction of the unified matrix A, we know, if $\beta_{MN}>0$, then $\beta_{NM}>0$. That means if L'_{MN} is not zero matrix then L'_{NM} is not zero matrix too. Also, L'_{MN} and L'_{NM} are all positive matrices. So A has somewhat symmetry character. That is, if A_{ij} is non-zero then A_{ji} is non-zero too.

Notice that the transformation of A, PAP^T, doesn't change the symmetry couple relation of A. It mean that the transformed matrix PAP^T has the same feature as original matrix A: if element (i,j) is non-zero then the element (j,i) is non-zero. So PAP^T has the format of $\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$ ■

Lemma C If one matrix A is non-negative, row-stochastic matrix, and irreducible, then iterative calculation $x^{i+1} = A^T x^i$ converge to the principle eigenvector of A. (Assume x^0 is positive and normalized vector).

Proof: A is non-negative, row-stochastic matrix also irreducible, thus, A is an ergodic transition matrix of a Markov chain MC.

According the ergodic theory of Markov chain, if we can prove that the MC has one and only one stationary probability vector x^S, then the iterative calculation $x^{i+1} = A^T x^i$ can converge to the stationary vector x^S for any initial vector x^0. Here, we assume norm of x^0 is normalized to 1, and x^0 is positive.

To prove the Markov chain has only one stationary vector x^S, we get the following 2 points firstly:

1) For A is non-negative, row-stochastic matrix, $\rho(A)$, the spectral radius of A, is equal to 1.

2) For A is non-negative and irreducible matrix, $\rho(A)$ is an eigenvalue of A with multiplicity 1, and $\{x \mid x>0, Ax = \rho(A)x\} = \{x \mid x>0, A^T x = \rho(A)x\}$ [2]. Based on 2), there exists one and only one vector $x \geq 0$ (considering scaling) satisfying $xA = \rho(A)x$. From 1), $\rho(A)$=1. Hence, there exists one and only one vector $x \geq 0$ (considering scaling) satisfying $xA = x$.

If we scale x to make the sum of x is 1, it's easy to know the equation $xA^k = x$ existed for any k=1, 2… So x is the stationary vector of Markov chain MC. Also, x is the principle eigenvector of A.

Hence, if A is non-negative, row-stochastic matrix, and irreducible, then iterative method $x^{i+1} = A^T x^i$ converge to the principle eigenvector of A. ■

Theorem: For the unified matrix A defined by (5), iterative method $w = A^T w$ converge to the principle eigenvector of A.

Proof: Firstly, A is a non-negative, row-stochastic matrix. If A is irreducible, then according to lemma C, we know the iterative method $w = A^T w$ converge to the principle eigenvector of A.

If A is reducible, let $w' = Pw$, here P is the permutation matrix fitting $PAP^T = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$. Then the iterative method turns to be $w = \begin{bmatrix} A_1^T & 0 \\ 0 & A_2^T \end{bmatrix} w$.

By the lemma B, A_1 is a non-negative, row-stochastic and irreducible matrix. And A_2 is non-negative, row-stochastic. If A_2 is reducible, we can apply lemma B on it and transform it to block-like diagonal matrix, with sub-matrix being irreducible. So, without loss of generality, we assume A_1, A_2 are irreducible. Hence, we rewrite w' to be $\begin{pmatrix} w'_1 \\ w'_2 \end{pmatrix}$, then we get two sub-iterative methods: $w'_1 = A_1^T w'_1$, and $w'_2 = A_2^T w'_2$. Based on lemma C, these 2 methods all converge. Taking limitation on the original iterative method: $w = A^T w$, we know w is an eigenvector of A associated with eigenvalue equals to 1. Also, we know spectral radius of A is 1, so w is the principle eigenvector of A. ■