Using Link Types in Web Page Ranking and Filtering

Zhanzi Qiu, Matthias Hemmje, Erich J. Neuhold
FHG-IPSI, Dolivostr. 15, 64293 Darmstadt, Germany
{qiu, hemmje, neuhold}@ipsi.fhg.de

Abstract

Corresponding to the evolution of the Web from the poorly structured towards more structured, semantic rich network, search methods that apply to it are also evolving from using little structural and semantic information to using more such information that is available. This paper proposes a ranking and filtering mechanism that makes use of link types that is representable with new Web standards. We suggest that page ranking can be propagational through links and the propagational rates depend on the types of the links and users' specific set of interests. Page filtering can be decided based on link types combined with some other information relevant to links. For either a ranking or filtering task, a profile containing a set of ranking or filtering rules to be followed in the task can be specified to reflect users' specific interests. Technical issues in implementing the mechanism in a search system are also discussed.

Keywords: *link types, ranking, filtering, search system, the Web*

1. Introduction

The traditional Web lacks explicit structural meta information [28] and the search engines on it are typically keyword-based. To improve the ability of expressing structures and semantics on the Web, the W3C [29] has been making an effort to develop and promote new Web standards for several years. Accompany this effort, the Web is undergoing an evolution from the poorly structured towards more structured, semantic rich network. And search methods that apply in the Web space are also evolving from using little structural and semantic information to using more such information that is available.

Adhering to this trend, we have explored several hyperstructure-based search methods, such as using hypertext contexts as Web search boundaries [19; 20] and applying hypertext composites in structured query and search [22]. We also have presented an idea of making use of link-based domain models, which are hyperstructures that have domain specific semantics, in formulating structured queries [21] so that Web users can get more specific results relevant to their information needs.

This paper describes another part of our effort in this direction. In this work, we focus on exploring the value of the most basic semantic information in the Web space, i.e. link types, in search activities. Link types are used for page ranking and filtering purpose. The mechanism will help users to get more rational page ranking in a long list of search hits and provides users more specific search results relevant to their information needs through filtering.

The rest of the paper is organized as follows. Section 2 gives a formal description of links and link types. Section 3 answers the question how links and link types can be represented with new Web standards. Section 4 and 5 propose schemas for using link types in page ranking and filtering. Section 6 discusses several technical issues in implementing the schemas while Section 7 talks about the experimental issues. Finally, Section 8 mentions related work and Section 9 concludes this work and outlines our future activities.

2. Links and Link Types

In a hypertext/hypermedia environment, the basic information components are nodes and links. A node is a unit of information [30]. It may contain any kind of data, such as a fragment of text, a graph, a picture, sound and motion video sequence. A node type specifies a concept, i.e. the nature of the kind of information contained in the node.

A link is a relationship between nodes. The meaning of a link is often indicated by the semantic type of the link. A link with a specified semantic type is a semantic link.

2.1. Semantic Links

Formally, a semantic link $\lambda \in \Lambda$ consists of the following components: [9]

$$\lambda = \langle t_\lambda, sn_\lambda, dn_\lambda, A_\lambda, \sigma_\lambda \rangle$$

where

$t_\lambda \in T_\Lambda$ is the link type

$sn_\lambda \in N$ denotes the source node of the link λ

$dn_\lambda \in N$ denotes the destination node of the link λ

$A_\lambda = \{a_1, a_2, \ldots, a_{k\lambda}\} \subseteq \mathsf{A}_\Lambda$ is a set of structured link attributes

$\sigma_\lambda \in \Sigma$ is a free text annotation to the link.

The link type t_λ serves to discriminate between various types of semantic links. It can be used to restrict a domain community to a few link types so that their semantics can be understood fully by both information providers and users in the domain. If we assume a closed hyperstructure, the source node sn_λ and the destination dn_λ make part of the set of nodes N that constitutes this structure. The link information is composed of a structured part (the link attributes A_λ) and of an unstructured part σ_λ.

In reality, for a link λ, the components sn_λ and dn_λ are imperative to exist, while t_λ, A_λ and σ_λ are not. A link with no specification of t_λ, A_λ and σ_λ is a non-semantic link, or can be seen as a special kind of semantic links.

As the value of A_λ and σ_λ in ranking and filtering Web pages is not explored in this work, in the rest of the paper a semantic link is simply denoted as $\lambda = \langle t_\lambda, sn_\lambda, dn_\lambda \rangle$.

2.2. Link Types

Starting from early research on semantic net [31], typed links have long been a topic of interest. Many people have advocated typed links [26; 17; 12; 5; 6]. In 1993, Trigg [27] even listed a set of 80 classes of link types. Link types can help users and computers to distinguish various kinds of links [28].

In the context of the Web, the typical nodes are Web pages or fragments in the pages (thus we use the terms *node* and *page* alternatively later in the paper). Links may exist between any of these nodes. Most of the links on the current Web are not typed. However, HTML 3.2 [13] and HTML 4.0 [14] define a set of generic link types to describe relationships of a given node with other nodes. Dublin Core [7] can also be seen as an ongoing effort to define a relation type system for this global hypertext system. It defines an element "RELATION" in its element set to provide a means to express relationships among resources (metaphor of nodes in RDF [23]) that have formal relationships to others, but exist as discrete resources themselves, e.g. images in a document, chapters in a book, or items in a collection.

In most cases, link types are domain-specific and usually exist in certain domain models. It is anticipated that in addition to the above mentioned generic link type systems various domain-specific link type systems will also come into the Web as new Web standards become more adopted and some browsing and search systems that take into account this kind of information appear. This anticipation motivates our work.

3. Standard Representation of Links and Link Types

Like any other information, links and link types can be shared and reused in the Web space only when they are represented in a standard way. To our pleasure, several Web standards - HTML, XLink and RDF - have provided the possibility.

HTML

In HTML 2.0 and its later versions [13; 14], a special kind of *LINK* element, which may only appear in the head of a document, is defined to specify a relationship between the current document and another resource. The types of the links are represented in their *rel* (specifying a forward link) and *rev* (specifying a reverse link) attributes. These types are defined in HTML language itself or by users (Only HTML 4.0 and above permit user-defined link types).

The following example illustrates how link types are encoded in *LINK* definitions:

```
...the rest of the document...
<HEAD><TITLE>Chapter 2</TITLE>
<LINK rel="Index" href="../index.html">
<LINK rel="Next"  href="Chapter4.html">
<LINK rel="Prev"  href="Chapter2.html">
</HEAD>
...the rest of the document...
```

XLink

In XLink [32], links are encoded in linking elements. The types of links can be encoded via the *role* attributes of linking elements. Following is an example of an out-of-line extended xlink:

```
<mylink xml:link=”extended” inline=”false”
   role=”commentary”>
  <locator href=”smith2.1” role=”Essay”/>
  <locator href=”jones1.4” role=”Rebuttal”/>
  <locator href=”robin3.2” role=”Comparison”/>
 </mylink>
```

RDF

In RDF [23], a statement may specify a named relationship between two resources. In a broad sense, such a named relationship can be regarded as a typed link. The link types can be encoded in *propertyElt* element via its tag (property name) and *resource* attribute. As in the following example, the link type IsPartOf is represented by the tag *<dc:relation>* and the *rdf:resource* attribute in *<dcq:RelationType>* :

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/TR/WD-rdf-
   syntax#"

 xmlns:dc="http://purl.org/metadata/dublin_core#"
 xmlns:dcq="http://purl.org/metadata/dublin_core
 _qualifiers#">
   <rdf:Description about="http:doc2">
     <dc:Relation>
     <dcq:RelationType
   rdf:resource="http://purl.org/metadata/dublin_
   core_qualifiers#IsPartOf>
       <rdf:value resource="http:doc1/>
      </dc:Relation>
  </rdf:Description>
 </rdf:RDF>
```

Note that the property names must always be associated with a schema, which represents a domain model or ontology, such as Dublin Core [7]. This is consistent to what we have mentioned before, i.e., link types are often domain specific.

4. Using Link Types in Ranking

There are some systems that make use of links in their ranking mechanisms. Most of them however only count the number of links associated with a search hit [4; 16; 3].

In our work, we agree with Brin et al. [3] that page ranking can be propagational through links. Furthermore, we take into account not only the number of links associated to search hits but also the types of the links. In another word, we suppose that the rank of a page depends on the sum of the ranks of the pages linking to it and the types of those links.

4.1. A Simplified Consideration

Intuitively, a simplified ranking mechanism that uses link types can be described like this: Links that are not relevant to the current search are filtered out and not counted in the ranking formulas.

For example, in Figure 1, the node at the lowest position (n5) has 2 incoming links, one is of type *a*, and the other is of type *b*. If link types are not considered, the number of incoming links that will be counted in the ranking is 2, while if link types are considered, say, to select only *a* type links, the number of incoming links counted in the ranking is 1.

It is clear that with respect to some users' specific interests, by taking into account the link types may improve the ranking of search hits.

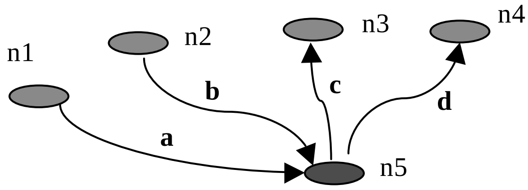

Figure 1. Typed links

4.2. A More Formal Consideration

Considering more carefully, the meaning of link types in ranking is not simply as “take it” or “ignore it”. It seems necessary to introduce a factor called *ranking propagational rate* (RPR) into the mechanism. This factor will blur the absolute distinction between different link types and may help to produce more reasonable ranking results with respect to users' specific interests.

Formally, the page ranking mechanism using link types is described as:

Let u be a page, $V = \{v_1, v_2, ..., v_k\}$ be a set of pages linking to u, t_i be a type of links, $f(t_i,v_i,u)$ be a value describing the *ranking propagational rate* from v_i to u, then the rank of u $R(u)$ can be defined as:

$$R(u) = \sum_{v_i \in V} f(t_i,v_i,u) * R(v_i)$$

$f(t_i,v_i,u)$ reflects a kind of view or interest on the page u from the page v_i (in which this link type is very important or unimportant). It also depends on a user's specific set of interests.

For instance, in Figure 1, $R(n5) = f(a,n1,n5) * R(n1) + f(b,n2,n5) * R(n2)$.

4.3. Ranking Propagational Rate

Where can a ranking propagational rate for a link come from? This is a critical issue to be addressed in order to make the mechanism really work.

There should be a few factors that may affect the value of a ranking propagational rate for a link. In our work up to now, we take into account the following information relevant to links:

- the type of the link
- the type of the source node of the link
- the type of the destination node of the link
- the content similarity of the two nodes of the link

Stated in a formal way, the value of a ranking propagational rate corresponding to the combined condition of link type *T-link*, source node type *T-sn*, destination node type *T-dn* and content similarity between the two nodes *Sim* is denoted as

RPR (T-link, T-sn, T-dn, Sim)

For a link $\lambda = <t_\lambda, sn_\lambda, dn_\lambda>$, **IF**

$(t_\lambda ==$ T-link) AND
(TypeofNode (sn_λ) == T-sn) AND
(TypeofNode (dn_λ)== T-dn) AND
(Similarity (sn_λ, dn_λ) >= Sim)

THEN its ranking propagational rate

$f(t_\lambda, sn_\lambda, dn_\lambda)$ = RPR(T-link, T-sn, T-dn, Sim).

A ranking profile specified for performing a ranking task may consist of a set of ranking propagational rates and their conditions. Such a profile can be organized in a table as:

Type of Link	Type of Source Node	Type of Destina-tion Node	Similarity of Source Node and Destination Node	Ranking Propaga-tional Rate
T-link	T-sn	T-dn	Sim	RPR

Each row in the table represents the expression for computing a ranking propagational rate (the last column) and its conditions (the other columns). The expression can be a constant, or built based on some symbol agreements, e.g. *1/n*, where *n* is supposed to be the number of the links of a certain type in the source node of the links. During performing the ranking task on a hypertext document collection, a system just needs to go through the rows, get corresponding ranking propagational rate expressions and compute the rate for each link in the collection.

The profiles for performing ranking tasks in a certain domain can be specified by users before they ask for the ranking or predefined and contained as parts of the domain model, which defines node types and link types used in the domain. In the latter case, the profiles transfer the view or interest of the authors or document providers about the nodes and links in their documents to the users and should be helpful suggestions for users to get a high-quality ranking.

4.4. Computing Page Ranks

To enable computing page ranks in a document collection, a set of pages to be used as the sources of ranking need to be specified and assigned with certain page ranks. These pages usually reflect users' specific information interests and personalized views on them and even their neighborhood pages. With the page ranks of these source pages and a certain ranking profile, the ranks for all the other pages in the collection can be computed fairly straightforward if the issues of scale are ignored.

5. Using Link Types in Filtering

As for ranking, link types can also be used in page filtering to improve the efficiency of browsing and searching on the Web. A simple example to support this argument is that a browsing system may hide (or in other way) the links of certain types from a page so that the users don't need to go to the pages that are linked from the page but are not relevant to users' interests. In another word, the system filters out those pages for users by taking advantage of link types.

In a deliberate consideration, a filtering mechanism making use of link types may be more efficient when it takes into account some other information relevant to links. Whether a node (page) will be filtered out in a browsing or searching activity can be decided based on one or more kinds of the following information:

- type of the node to be filtered out
- information related to the incoming links of the node, including
  - the types of those links
  - the types of the ancestor nodes, i.e. the nodes which are the source nodes of the

links (the node is the destination node of the links)
  - the content similarity of the node and the ancestor nodes
- information related to the outgoing links of the node, including
  - the type of those links
  - the types of the descendant nodes, i.e. the nodes which are the destination nodes of the links (the node is the source node of the links)
  - the content similarity of the node and the descendant nodes

Formally, a filtering rule considering all the above kinds of information can be denoted as:

toFilterOut (T-node, INC, OUT), in which

$INC = \{inc_1, inc_2, \ldots, inc_n\}$
$inc_i = < T\text{-}incLink_i, T\text{-}incNode_i, Sim\text{-}inc_i >$

$OUT = \{out_1, out_2, \ldots, out_n\}$
$out_i = < T\text{-}outLink_i, T\text{-}outNode_i, Sim\text{-}out_i >$

In a filtering following this rule, given a node u, suppose B_u is a set of ancestor nodes of u and F_u is a set of descendant nodes of u, u will be filtered out

IF

(TypeofNode (u) == T-node) ***AND***
(Foreach $inc_i \in INC$*, there exists the semantic link* $<T\text{-}incLink_i, b_u, u>$
where $b_u \in B_u$ ***AND*** *TypeofNode* (b_u) == $T\text{-}incNode_i$ ***AND*** *Similarity* $(b_u, u) < Sim\text{-}inc_i$*)*
AND
(Foreach $out_i \in OUT$*, there exists the semantic link* $<T\text{-}desLink_i, u, f_u>$
where $f_u \in F_u$ ***AND*** *TypeofNode* (f_u) == $T\text{-}outNode_i$
AND *Similarity* $(f_u, u) < Sim\text{-}out_i$*)*

.

Note that the content similarity value contained in the rule will play an important role in handling the cases in which there are more than 1 link between two nodes, say node A and node B.

In practice, a filtering profile used for performing a filtering task may need to contain a set of filtering rules. It can be represented in a table as:

Type of Node	**Incoming Links**	**Outgoing Links**
T-node	(INC)	(OUT)

Each row in the table represents a filtering rule. In fulfilling a filtering task with such a profile, all nodes that meet one or more rules in the profile will be filtered out.

Like ranking profiles, filtering profiles may also be specified by either document users or providers. In the latter case, the profiles may be provided as parts of certain domain models and they are helpful suggestions from the document providers to the users for filtering.

6. Implementation

A search system that intends to implement the ranking and filtering mechanisms presented above can be built upon a normal keyword-based search system. In a nutshell, it consists of at least the following engines that represent different aspects: link information gathering engine (crawler & extractor), ranking/filtering profile editor, keyword indexing engine, query engine (user interface), retrieval engine (for deriving search results) and DB Manager for storing data in the system.

In this section we will not describe the mature keyword-based indexing and searching technology but discuss several of the special issues to be addressed in implementing such a system.

6.1. Link Information Gathering and Storing

At first, the system needs to have a crawler and an extractor to gather link information from the Web. The crawler downloads the Web pages (nodes) specified by users or parsed out from the pages during the processing and stores them in the document repository. Every page gets an associated ID number called nodeID that is assigned whenever a new URI is parsed out of a page. The extractor (e.g. JEDI [15]) extracts all the link information represented with Web standards (as described in Section 3 above) in the collection and stores the information in the link base of the system. The database schema for the link base is shown in Figure 2.

The node type table stores all the node types used in the collection. The node table is for storing the types and URIs of the nodes gathered in the collection. The link type table is for storing the link types existing in the collection. The link table contains the links between the nodes in the collection.

6.2. Ranking and Filtering Profiles Editing and Storing

As described in Section 5, any ranking or filtering task will be performed by the system based on a profile specified by users or document providers. For editing such a profile, a user-friendly interface needs to be provided. Such an interface presents all the ranking or filtering rules contained in a profile and enables users to insert new rules into the profile (as shown in Figure 3 and 4).

Node Type Table	nodetypeID	nodetype		
Node Table	nodeID	nodetypeID	URI	
Link Type Table	linktypeID	linktype		
Link Table	linkID	source_nodeID	linktypeID	target_nodeID

Figure 2 Database schema for link Information – a link base

To enable the reuse of the defined profiles in the system, the database schema as shown in Figure 5 can be designed for storing the profiles. The database tables in the schema implement the tables for representing ranking or filtering profiles described in Section 4 and 5. In each table, all columns for describing node, node type, link and link type contain certain id values referring to the corresponding column values in the link base described in last subsection (6.1). Every time when a profile stored in the database is to be modified, the system gets the data of the profile from the database and constructs the interface for editing (as shown in Figure 3 and 4) dynamically.

6.3. Form-Based Interface for Specifying Ranking or Filtering Profiles in Queries

In the system, the interface for specifying a ranking or filtering profile can be as simple as a selection field contained in a query form (varies a lot with respect to complexity in different systems). The options listed in the selection field are all the ranking or filtering profiles defined in the system.

As a simple example, Figure 6 shows a query page that contains only one field for specifying query criteria other than the selection field for profiles. Figure 7 shows a result page that indicates which ranking or filtering profile has been used in deriving the search. The result page also provides a link for going back to the query page.

6.4. Deriving Search Results

To derive results for queries with a specified ranking and/or filtering profile, the retrieval engine of the system needs to contain at least 3 components: keyword-based search component, filtering component and ranking component. Each component performs a part of the work for deriving search results, as its name indicates.

Concretely, the process of deriving the search results for a query can be divided into 3 steps:

1. The keyword-based search component derives

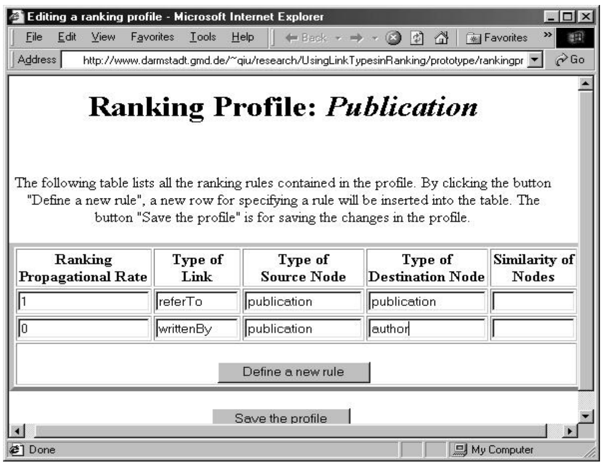

Figure 3 Editing ranking profile

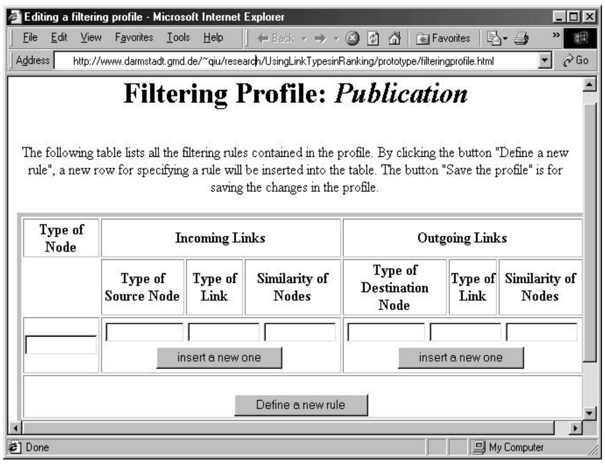

Figure 4 Editing filtering profile

keyword-based search results corresponding to the input query terms. *(Let r1 denotes the result from this step).*

2. If a certain filtering profile has been selected in the query, the filtering component performs filtering to the result from the first step (i.e. *r1*) according to the selected filtering profile. (*(Let r2 denotes the result from this step. If no filtering has been performed, r2=r1).*
3. If a certain ranking profile has been selected in the query, the ranking component performs ranking to the result from the second step.

appropriate user interfaces for specifying profiles and submitting ranking or filtering tasks.

For instance, in a system that supports hypertext context-based search [19; 20], the ranking and filtering mechanism can help users to define hypertext contexts when they browse on the Web. More clearly, a set of Web pages that meet a certain ranking or filtering rule can be added into or filtered out from a user-defined hypertext context. This may speed up the definition of hypertext contexts that reflect users' specific interests well and furthermore improve the quality of searches performed within the boundary of the hypertext contexts.

Ranking Profile Table	R_ruleID	linktypeID	snodetypeID	dnodetypeID	similarity	RPR
Filtering Profile Table	F_ruleID	nodeID				
Incoming Links Table	inclinkID	F_ruleID	snodeID	linktypeID	similarity	
Outgoing Links Table	outlinkID	F_ruleID	dnodeID	linktypeID	similarity	

Figure 5 Database schema for ranking and filtering profiles

After these 3 steps, the search results corresponding to a query have been derived. If users are not satisfied with the results, they can go back to the query page, select other profiles and submit the query again (as shown in Figure 7).

6.5. Discussion

In a search system, page filtering and ranking may be performed not only to search results corresponding to various queries but also in the process of specifying search criteria. What needs to be done is to design

7. Experimental Issues and Results

While we can not get from the current Web a large document collection with rich explicitly represented link type information that is necessary for performing a thorough evaluation, we have done several preliminary experiments with an improved version of DELITE-Webglimpse [19]. This improved DELITE-WebGlimpse supports the interactive annotation of link types and implements the ranking and filtering mechanism

Figure 6 An exemplary query page

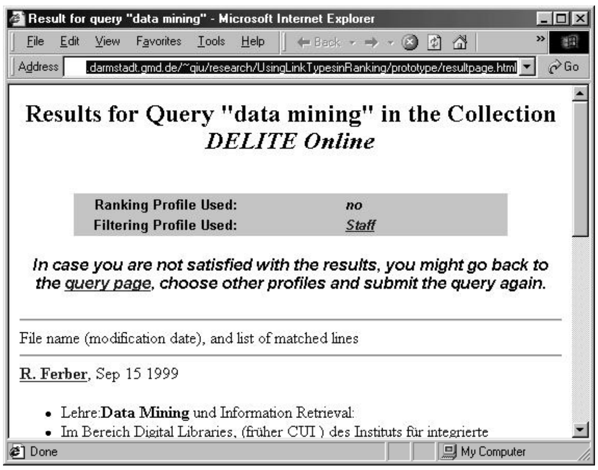

Figure 7 An exemplary result page

presented in this paper.

In one experiment performed in the collection of GMD Darmstadt site, we once tried to find out the DELITE (a research division in GMD) experts in a certain research field, e.g. "retrieval", by taking into accounts their publications and projects. To attain this, we first annotated the links from the pages for publications to the pages for staff members with the type "hasAuthor", and the links from the pages for projects to the pages for staff members with the type "hasContributor". Then we specified that the ranking propagational rate would be *1* if a link was of the type "hasAuthor" or "hasContributor" and the source and destination node of the link contained a certain keyword, e.g. "retrieval" (a kind of content similarity). Otherwise, the ranking propagation rate was set to *0*.

Furthermore, the pages for DELITE staff members, publications and projects were grouped distinctly into 3 hypertext contexts, "DELITE staff", "DELITE publications" and "DELITE projects", based on certain filtering profiles. The rank of each page containing "retrieval" in the contexts was set to 1.

Finally, with the ranking profile specified, searching "retrieval" in the context "DELITE staff" got a quite rational ranking list for the DELITE experts in the IR research field. The first 3 persons in the list are all senior scientists and mentors of other staff members in the field.

When another ranking profile, say the ranking propagation rate for the selected link types and source and destination nodes was *1/n* (*n* is the number of the links of a certain type in a source node), was specified for the task, the resulting ranking list looked even better. The first 3 persons got the same positions, while some other later got more reasonable positions in the list. An interpretation about this difference may be that the first ranking profile doesn't take into account the different number of a certain typed link in a page. It should be real that the impact of a linked page p*1*to a page *p2* is less if the page *p1* has links of a certain type to many other pages. And this impact may vary in different applications.

8. Related Work

Link structures has been studied in hypertext research that predates the Web [1; 24; 10; etc]. Especially, Frei and Stieger [8; 9] have discussed how knowledge of the adjacency of nodes via hyperlinks can be used to help a user navigate or find the answer to a query.

In the context of the Web, some systems count the number of links associated with search hits in their ranking mechanism [4; 16; 3]. The success of the Google system [3] proves the great value of link information in improving search quality including page rank. There is also work that uses link information to propagate rewards from interesting pages to those that point to them [2].

From the IR research community, Justin Picard has shown a technique derived from logic and probability for integrating the evidence provided by document links [18]. Norbert Fuhr has presented a probabilistic version of Datalog (pD) and shown its suitability for IR [11]. Rölleke & Blömer have shown an application of the pD. They use a probabilistic link type space and uncertain links that yield a ranking of retrieved documents according to their links to documents which have been retrieved with respect to the content criteria of the query [25]. The CACM collection has been used in all their experiments and the link types mainly considered were "citing" and "cited" relationship between documents. No indication has shown how the case will be when the models are applied in the context of the Web, in which a large number of pages and links of various types exist.

Comparatively, the mechanism we present in this paper is quite a general one that makes use of not only the number of links but also the link types that can be explicitly represented on the Web for page ranking and filtering purpose. The system interface designed for enabling the users to edit and specify various profiles for ranking/filtering tasks makes the mechanism easily to be applied to various domains on the Web.

9. Summary

This paper presents a ranking and filtering mechanism that makes use of link types that are representable with new Web standards. We suggest that page ranking can be propagational through links and the propagational rates depend on the types of the links and users' specific set of interests. Page filtering can be decided based on link types combined with some other information relevant to links. For either a ranking or filtering task, users or document providers can define or specify a certain profile that contains a set of ranking or filtering rules to be followed in the task.

There is a great significance that a search system implementing this mechanism will save users effort in getting specific search results relevant to their information needs. However, a thorough evaluation about the method and the system is still to be done. Such an evaluation will include not only the quality of search results with respect to precision and recall but also the technical system performance and scalability with respect to indexing and searching. After the evaluation is done, some refinements of the mechanism will be further made.

Finally, it should be a great challenge to define models or standards of link types for various domains and ask document creators to follow the rules. It is useful to study how the existing information about link types (as mentioned in section 3) would be helpful to automatically compute ranking and filtering, thus leveraging the users' work in defining the ranking and filtering profiles.

Reference:

[1] Botafogo, R., Rivlin, E., Shneiderman, B., "Structural analysis of hypertext: Identifying hierarchies and useful metrics," ACM Transactions on Information Systems, 10(1992), pp. 142-180.

[2] Boyan, J., Freitag, D. and Joachims, T., "A machine learning architecture for optimizing Web search engines," in the AAAI-96 Workshop on Internet-based Information Systems.

[3] Brin, S. and Page, L., "The anatomy of a large-scale hypertextual Web-search engine," Proc. 7th International World Wide Web Conference, 1998.

[4] Carriere, J., Kazman, R., "WebQuery: Searching and visualizing the Web through connectivity," Proc. 6th International World Wide Web Conference, 1997.

[5] Conklin, J., "Hypertext: An introduction and survey," Computer, September 1987, vol. 20, no. 9, IEEE Publications, pp. 17-41.

[6] DeRose, S. J., "Expanding the notion of links," Proc. of Hypertext'89, pp. 249-257.

[7] Dublin Core: http://purl.oclc.org/dc/

[8] Frei, H. P. and Stieger, D., "Making use of hypertext links when retrieving information", Proc. 3th ACM Conference on Hypertext, pp. 102-111, 1992.

[9] Frei, H. P. and Stieger, D., "The use of semantic link in hypertext information retrieval," Information Processing and Management, Vol. 31, No. 1, 1995, pp. 1-13.

[10] Frisse, M. E., "Searching for information in a hypertext medical handbook," Communications of the ACM, 31(7), pp. 880-886, 1988.

[11] Fuhr, N., "Probabilistic datalog: Implementing logical information retrieval for advanced applications," Journal of the American Society for Information Science (JASIS), Volume 51, Number 2, 2000, pp. 95-110.

[12] Halasz, F. G., "Seven issues," Revisited (keynode address), Proc. 3rd ACM Conference on Hypertext, San Antonio, TX, 1991.

[13] HTML 3.2: http://www.w3.org/TR/REC-html32.html.

[14] HTML 4.0: http://www.w3.org/TR/REC-html40/.

[15] Huck, G., Fankhauser, P., Aberer, K., Neuhold, E., "Jedi: Extracting and synthesizing information from the Web," Proc. 3rd IFCIS International Conference on Cooperative Information Systems (CoopIS'98), New York City, August 1998, pp. 32-43.

[16] Kleinberg, J., "Authoritative sources in a hyperlinked environment," Proc. ACM-SIAM Symposium on Discrete Algorithms, 1998.

[17] Mohageg, M. F., "The influence of hypertext linking structures on the efficiency of information retrieval," Human Factors, 1992, 34(30), pp. 351-367.

[18] Picard, J., "Modeling and Combining Evidence Provided by Document Relationships Using Probabilistic Argumentation Systems," Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998), August 24-28 1998, Melbourne, Australia. ACM 1998, pp. 182-189

[19] Qiu, Z., Hemmje, M., Neuhold, E. J., "Towards Supporting User-Defined Hypertext Contexts in Web Searching," in: Proc. IEEE Advances in Digital Libraries 2000 (pp. 48-57). Los Alamitos, CA, IEEE Computer Society.

[20] Qiu, Z., Hemmje, M., Neuhold, E. J., "ConSearch: Using Hypertext Contexts as Web Search Boundaries," in: Etzion, Opher & Scheurmann, Peter (Eds.), International Conference on Cooperative Information Systems (CoopIS 2000), (pp. 42-53). Berlin [etc.]: Springer, 2000 (Lecture notes in computer science; 1901).

[21] Qiu, Z., Hemmje, M., Neuhold, E. J., "Using Link-Based Domain Models in Web Searching," in Proc. 2000 KYOTO International Conference on Digital Libraries: Research and Practice, November 13-16, 2000, Kyoto University, Kyoto, Japan. Los Alamitos, CA, IEEE Computer Society.

[22] Qiu, Z., Hemmje, M., Neuhold, E. J., "Using Hypertext Composites in Structured Query and Search," in: Bauknecht, Madria & Pernul (Eds.), 2nd International Conference on Electronic Commerce and Web Technologies (EC-Web 2001), (pp. 326-336). Berlin [etc.]: Springer, 2001 (Lecture notes in computer science; 2115).

[23] RDF: http://www.w3.org/RDF/

[24] Rivlin, E., Botafogo, R. and Shneiderman, B., "Navigating in hyperspace: designing a structure-based toolbox," Communications of the ACM, 37(2), pp. 87-96, 1994.

[25] Rölleke, T., Blömer, M., "Probabilistic Logical Information Retrieval for Content, Hypertext, and Database Querying," Proc. Hypertext - Information Retrieval - Multimedia '97 (HIM'97), Sept. 29 – Oct. 2, 1997, Dortmund, Germany.

[26] Shum, S. B., "The missing link: hypermedia usability research & the Web," Interfaces, British HCI Group Magazine, Summer, 1996. Reprinted in ACM SIGCHI Bulletin 28 (4), 68-75.

[27] Trigg, R., "A network-based approach to text handling for the online scientific community," Ph.D. thesis, Dept. of Computer Science, Univ. of Maryland, Nov. 1983.

[28] Trigg, R. H., "Hypermedia as integration: Recollections, reflections and exhortations," in Keynote Address in Hypertext'96 Conference. Xerox Palo Alto Research Center.

[29] W3C: http://www.w3.org/.

[30] W3C Terms: http://www.w3.org/Terms.html.

[31] Woods, W. A., "What's in a link: foundations for semantic

networks," in Ronald J. Brachman and Hector Levesque, eds., Readings in Knowledge Representation, Morgan Kaufmann, 1985.

[32] XLink: http://web3.w3.org/TR/xlink/.