A Path-based Approach for Web Page Retrieval[*]

Jianqiang Li, Yu Zhao
NEC Labs, China
{li_jianqiang, zhao_yu}@nec.cn

Hector Garcia-Molina
Stanford University
hector@cs.stanford.edu

ABSTRACT

Use of links to enhance page ranking has been widely studied. The underlying assumption is that links convey recommendations. Although this technique has been used successfully in global web search, it produces poor results for website search, because the majority of the links in a website are used to organize information and convey no recommendations. By distinguishing these two kinds of links, respectively for recommendation and information organization, this paper describes a path-based method for web page ranking. We define the Hierarchical Navigation Path (HNP) as a new resource for improving web search. HNP is composed of multi-step navigation information in visitors' website browsing. It provides indications of the content of the destination page. We first classify the links inside a website. Then, the links for web page organization are exploited to construct the HNPs for each page. Finally, the PathRank algorithm is described for web page retrieval. The experiments show that our approach results in significant improvements over existing solutions.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Design, Experimentation.

Keywords

Navigation Path, Web Search, Web information retrieval

1. INTRODUCTION

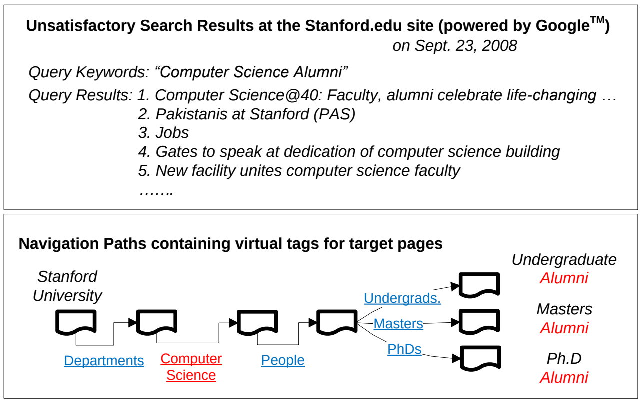
Links created by the web page authors provide the basis for web browsing and are also critical for successful web page retrieval. From the similar nature of academic citations and links [35], a basic assumption that links convey human recommendations can be derived directly. Based on this underlying assumption, many have conducted research on link analysis (e.g., PageRank [22] and HITS [17]) to exploit the Web structure to capture the relative importance of a web page. This approach has demonstrated significant improvement of the performance of global web search engines compared with those from text-only techniques [29].

However, this basic assumption on the recommendation semantics of links is "close enough" to the truth [26] only at the level of the global Web. In general, this assumption does not hold at the local level of the Web, i.e., a publicly accessible website or an intranet. The main reason why the assumption fails to hold is that typically a large amount of links at a local website are utilized mainly for organizing the web pages at the website. Such links are, by nature, "well-structured" and have little semantics of recommendation. In addition, for the case of small web, the number of the inter-site links is too small. This fact causes the websites that adopt the same search technologies as those used in global search engines to fail to provide satisfactory search results [15]. In fact, the TREC-8 Small Web Task [16] has shown that almost no benefits can be obtained from link based methods. The research [30] also suggested that PageRank from intranet link analysis cannot provide effective discriminative information among web pages. At present, employing the links with their semantics of recommendation to enhance web search quality has been widely studied [17][22]. However, to the best of our knowledge, little work has been done on utilizing those links that are primarily for local web page organization to help improve Web information retrieval.

[*] A preliminary version of this paper appeared in [18]. This submission includes more complete and formal description of the algorithms and experiments.

This paper describes our research in exploiting the "well-structured" links in local websites to improve web page ranking. PageRank algorithm assumes that links convey recommendations and computes a query-independent feature. Our research assumes that "well-structured" links propagate the topics of web pages, i.e., the destination page of a link inherits the topics of the source page. Since the page ranking exploits the words appeared in the multiple hops of links, it is a query-dependent measurement. The idea originates from the following observation: The builder of a website generally employs an explicit or implicit hierarchy of links to help organize or categorize the website's collection of pages; Readers generally utilize the information appearing in multiple levels of such links as a guide in navigating through the web site; When taken as a whole, the anchor text associated with the links, and the URL, page titles, etc. of the source and destination pages in a reader's navigation path often give clear indication of the nature or purpose of the destination page. We surmise that by utilizing information inherent in such Hierarchical Navigation Paths (HNPs), we can improve the accuracy of web page retrieval.

To characterize the usefulness of navigation path in Web search applications, let's suppose a user wants to obtain the alumni list of the Department of Computer Science at Stanford University. To get his required information, he might submit a query with keywords "computer science alumni" to the stanford.edu website search engine. However, the top-100 hits (queried on Sept. 23, 2008) do not contain links to any alumni list pages directly. Instead, a user can find three alumni list pages for "Undergrads", "Masters" and "PhDs" respectively by manual navigation from the homepage of the Department of Computer Science at stanford.edu. All these three pages do not contain any textual clues about "computer science"; however, such implicit sense could be deduced semantically from the navigational contexts, i.e., the department's homepage. That is to say, from the perspective of the user, because there is a navigation path from department's homepage to each alumni page, the alumni pages are virtually tagged with the topic of homepage, "computer science", as shown in Figure 1.



Unsatisfactory Search Results at the Stanford.edu site (powered by Google^{TM})
on Sept. 23, 2008

Query Keywords: "Computer Science Alumni"

Query Results: 1. Computer Science@40: Faculty, alumni celebrate life-changing …
2. Pakistanis at Stanford (PAS)
3. Jobs
4. Gates to speak at dedication of computer science building
5. New facility unites computer science faculty
…….

Figure 1. Website search scenario

Since HNPs can give important indicative or contextual information about the content of its associated web pages, they can be considered as a kind of new resource that can help raise the precision of the similarity calculation between query and a web page. To exploit HNP for improving web page ranking, this paper mainly addresses two problems: (1) to identify all the "web-structured" links and construct corresponding HNPs for each web page; (2) to makes use of the resulting HNPs as a kind of new resource for web page ranking. This paper proposes HNP discovery and PathRank algorithms for problems (1) and (2), respectively. Through these algorithms, the power of links for recommendation and links for information organization are combined together to improve web page retrieval. The proposed algorithms are evaluated on the web page collections of stanford.edu website and WT10g test collection of TREC. The experimental results reveal that our proposed approach outperforms existing solutions significantly.

The rest of the paper is organized as follows. Section 2 discusses the roles of links. Some basic concepts are given in Section 3. Section 4 and 5 describes the HNP discovery approach, where the algorithm for link classification is given in Section 4, and the HNP construction algorithms are presented in Section 5. The PathRank algorithm for exploiting HNPs to rank web page is given in Section 6. Section 7 describes the experimental results. The related work is considered in Section 8. Section 9 concludes the paper with discussions.

2. The roles of links

We informally identify two roles that links can play:

- A structural link *l* from page *s* to page *d* indicates that *s* and *d* have a definite organizational relationship. In particular, *d* may represent a component or an instance of what *s* describes. For example, a product list page *s* may point to individual products through structural links, or a university page may identify its departments via structural links.

- A reference link *l* from page *s* to page *d* tells *s*'s reader that page *d* may be of interest. For example, the author of page *s* may be identifying his favorite band or preferred search engine through *l*.

Notice that it is hard to ascertain the role of a link by examination, as we need to infer the intention of the link author. In this paper we will look for evidence that indicates that a link is likely to be reference or likely to be structural. Also notice that in practice, a link *l* may both be structural and reference, e.g., a products page may point to the most popular of its items. However, our goal will be to identify the *principal* role of a link.

We further classify structural links into two sub-roles:

- A hierarchical link *l* from page *s* to page *d* is mainly used to organize the content of a website into a hierarchy. For example, a university site uses hierarchical links to identify its schools (or divisions), which in turn identify departments, which in turn identify faculty and students, and so on.

- The main role of a navigational link *l* from *s* to *d* is to provide a shortcut from *s* to *d*, to assist in navigating a web site. For example, links that point back to the home page, or that point to the next item in a list, are navigational.

Again, our sub-classification is based on the *principal* role a link plays: if the main role is to organize a website, the structural link is hierarchical; if its main role is to facilitate quick navigation, it is navigational. And again, our goal here is to look for evidence that tells us what this main role is.

Generally, a hierarchical link is embodied as a subsumption relation (e.g., whole-part, parent-child, class-subclass, or class-property) between web pages. If the subject of a source page of a link subsumes that of its destination page in certain way, it is a hierarchical link, otherwise, it is a navigational link.

Our heuristic for distinguishing between structural and reference links is simple but very natural: we will say a link *l* from *s* to *d* is mainly reference if it is an inter-site link. If *l* points to a page *d* on another web site, then it is unlikely that *d* is closely related to *s*, and hence we assume is it mainly a reference. Our heuristic for separating hierarchical from navigational links in more complex and will be discussed in Section 4.

3. Basic concepts

In this section we present the basic notions to be used in this paper.

A website *W* can be represented as a triple *W*=<*P*, *L*, Δ>, where *P* is a set of web pages (files); *L* is the set of links defined inside the web pages of *P*; and Δ is the domain of the website (e.g., *stanford.edu* or *cs.stanfand.edu*). As we discuss below, the pages in *P* are constrained to be in domain Δ.

A *link* is displayed as an underlined or highlighted phrase or word in an HTML page and can direct the reader to another web page or another part of the same page when it is clicked. Formally, a link *l* has source page *s*(*l*) (where *l* occurs) and destination *d(l)* (the page that *l* points to). We use *at*(*l*) to denote the anchor text displayed for the link in *s*(*l*). When *l* is understood, we will write *s*, *d* and *at* for simplicity. Some pages contain multiple links to the same destination; in such a case we consider them to be a single link, where *at(l)* is the concatenation of all the anchor text.

Table 1. The example of domain and directory of a URL

p's URL	http://library.stanford.edu/depts/mathcs/research_help/guides/techreports.html
dom(*p*)	library.stanford.edu
dir(*p*)	library.stanford.edu/depts/mathcs/research_help/guides/

A web page p has two basic attributes: $url(p)$ is the page's URL, and $title(p)$ denotes the title. From $url(p)$ we can determine the domain of p, $dom(p)$, and a directory where p is located, $dir(p)$. Table 1 illustrates the domain and directory of a sample page.

Domains and directories form a hierarchy, and we will exploit this hierarchy when identify the role of a link. We say that domain X is a subdomain of Y, written $X \subset Y$, if X has Y as a postfix. For example, the domain *xyz.abc.com* is a subdomain of *abc.com*. Similarly, we say that directory X is a subdirectory of Y, $X \subset Y$, if X has Y as a prefix. For example, the directory *abc.com*/*d*1/*d*2/ is a subdirectory of *abc.com*/*d*1/. We use super-domain and super-directory to refer to the inverse relationships. For two domains (directories) X and Y, $X \subseteq Y$ means that $X \subset Y$ or $X=Y$.

Every domain X has an associated home page $home(X)$ that denotes the start or index page of the domain. When we analyze links it will be useful to refer to the home pages of all super-domains of a domain X, i.e., $superhome(X)=\{p \mid \exists\ Y \text{ s.t. } X \subset Y \text{ and } p=home(Y)\}$. We can define analogous functions for directories.

Table 2 summarizes the notation we have introduced.

Table 2. Basic notation

l	A link
$d(l)$	The destination of link l
$s(l)$	The source page of link l
$at(l)$	The anchor text of link l
p	A web page
$url(p)$	The URL of web page p
$title(p)$	The page title of p
$dom(p)$	The domain of web page p
$dir(p)$	The directory of web page p
$home(dom)$	The home page of domain *dom*
$superhome(domY)$	The set of homepages of the super-domains of *domY*
$superhome(dirY)$	The set of homepages of the super-directories of *dirY*

As we analyze a web site $W=<P, L, \Delta>$, we assume it is well formed. That is, P contains all pages in the domain Δ, and for every $p \in P$, $dom(p) \subseteq \Delta$. Similarly, for every $l \in L$, $s(l) \in P$.

Hints about the relationships between the source and destination of a link are syntactically encoded in their URLs. For example, say that for a given link $url(s(l))$= http://www.abc.com/d/file.html and $url(d(l))$ = http://www.abc.com/. In this case we can see that d is the home page of a superdomain of $dom(s)$. Hence, it appears that l is a navigational link that takes the user back into the web site hierarchy.

Using this intuition, we identify two types of links, based on purely syntactic hints. These properties will be used in the next section for link classification.

- Given a web site $W=<P, L, \Delta>$, we say that a link $l \in L$ from page s to page d is *syntactic-navigational* if one or more of following conditions holds:
  1: $d=s$;
  2: $d=home(dom(s))$;
  3: $d \in superhome(dom(s))$;
  4: $d \in superhome(dir(s))$;

- Given a web site $W=<P, L, \Delta>$, we say that a link $l \in L$ from page s to page d is *syntactic-reference* if $dom(d)$Ø Δ.

A syntactic-reference link leads outside the current website W, i.e., $d \notin P$.

Note that the syntactic-reference property of a link depends on the website definition, and in particular the top domain Δ. For example, if Δ= *stanford.edu*, a link *l* from page *s* with *dom*(*s*)=*cs.stanford.edu* to page *d* with *dom*(*d*)=*www.stanford.edu* is not a syntactic-reference link. However, if Δ= *cs.stanford.edu*, then *l* is a syntactic-reference link.

4. Link classification

To exploit the "well structured" hierarchical links for web information retrieval, first we need to discovery the links with the role of organizing the website. Following is the pseudo code of our heuristic hierarchical link discovery algorithm.

Heuristic HL discovery algorithm:

```
Input: $W$=<$P$, $L$, $\Delta$>
Output: Classify all links in $L$ into the sets of hierarchical links $HL$, reference links $RL$, syntactical navigational links $NL\_syn$ (obtained by syntactical URL analysis) and semantic navigational links $NL\_sem$ (obtained by the semantic link analysis)

1. $HL=\phi$, $RL=\phi$, $NL\_syn=\phi$, $NL\_sem=\phi$
2. For each $l \in L$ do
3.     If $l$ is syntactical-reference then assume the role of $l$ is reference link and $RL= RL \cup \{l\}$
4. For each $l \in L-RL$ do
5.     If $l$ is syntactical-navigational then assume the role of $l$ is navigational link and $NL\_syn= NL\_syn \cup \{l\}$
6. For each $p \in P$, extract its link collections and obtain $coll(p)=\{lc_1, lc_2, \ldots, lc_m\}$
7. For each $p \in P$ with $|coll(p)| \neq 0$ do
8.     For each $lc \in coll(p)$ do
9.         $OP(lc)=\{d(l)|l \in lc\}$; $OP(p)=\{d(l)|s(l)=p\}$
10.        Calculate $C(OP(lc))$ with Formula (2)
11.        For each $l \in \{l'|\ l' \in L-(RL \cup NL\_syn) \wedge s(l') \in OP(lc)\}$ do
12.            If $d(l) \in C(OP(lc)) \cap (OP(p) \cup \{p\})$ then assume the role of $l$ is navigational link and $NL\_sem = NL\_sem \cup \{l\}$
13. $HL=L-(RL \cup NL\_syn \cup NL\_sem)$
```

In Lines 2-3, the roles of the links with the syntactical-reference property are classified as reference link. After this step, all the remaining links in $L-RL$ are intra-site hierarchical links or navigational links. Lines 4-5 classify the roles of the links with syntactic-navigational property as navigational links and obtain corresponding set *NL_syn*.

Note that Lines 2-3 and 4-5 mainly employ the syntactical information implied in the URLs to detect reference links and navigational links. We call these lines the syntactical analysis phase.

However, in many websites, especially for the dynamic web pages generated from databases, the URLs cannot provide enough information for link classification. So in Lines 6-12, we employ a so-called semantic link analysis phase to exploit the page layout information together with the linkage relations between web pages to differentiate between navigational links and hierarchical links.

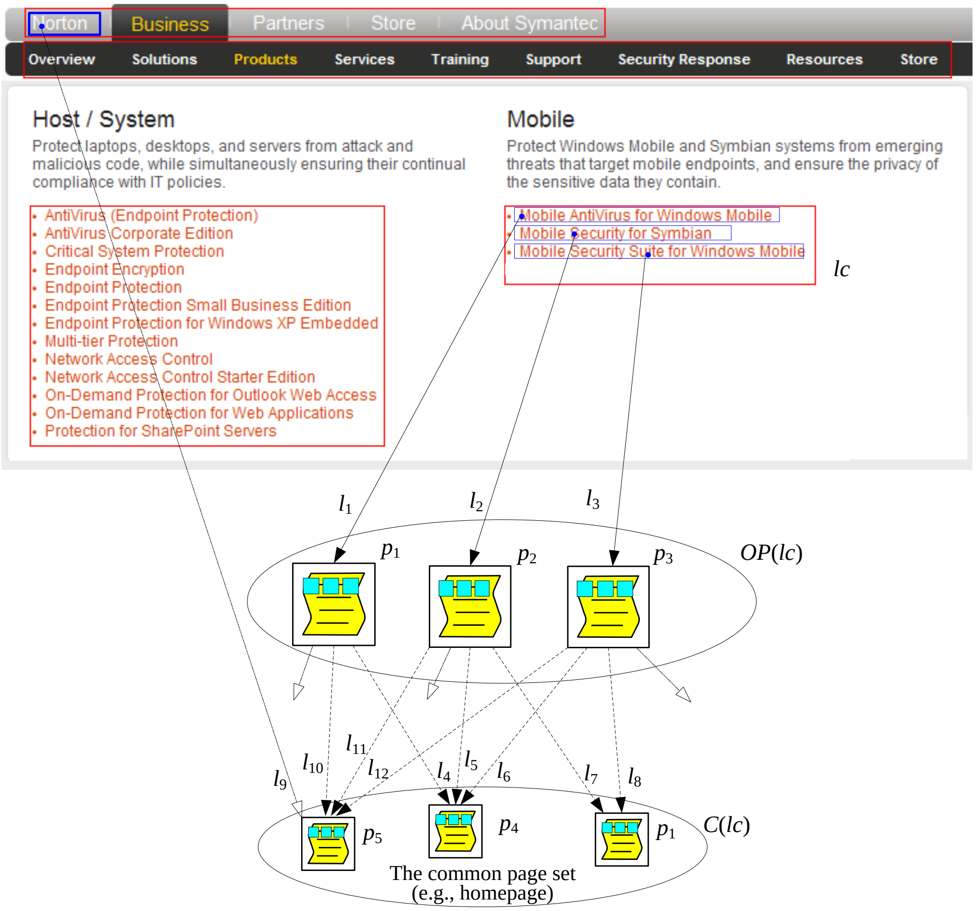
In this phase, we identify the navigational links first, and then the remaining links are hierarchical. In the remainder of this section, we describe the details of the navigational link detection.

The method adopted in this phase is based on a concept of Link Collection (LC). A LC is a set of links contained in a semantic block [8] of a web page. Examples of semantic blocks are shown in the web page of Figure 2. The links in each semantic block compose a LC.

Given a website W=<P, L, Δ>, the set of outbound links of a web page $p \in P$ can be represented as $link(p)=\{l'|l' \in L \wedge s(l')=p\}$. The set of LCs in p can be represented as $coll(p)=\{lc_1, lc_2, \ldots, lc_m\}$, where $lc_i \cap lc_j=\phi$ for $i \neq j$. The set $coll(p)$ is a partition of $link(p)$, i.e.,

$$\bigcup_{lc \in coll(p)} lc = link(p)$$

By assuming that a LC contains multiple links, Section 4.1 introduces the intuition and the basic rule for navigational link detection. In Section 4.2, the basic rule is extended to deal with a LC with only one link. Section 4.3 refines the basic rule by adding a new constraint derived from improved intuition.



Host / System

Protect laptops, desktops, and servers from attack and malicious code, while simultaneously ensuring their continual compliance with IT policies.

Mobile

Protect Windows Mobile and Symbian systems from emerging threats that target mobile endpoints, and ensure the privacy of the sensitive data they contain.

Norton
Business
Partners
Store
About Symantec
Overview
Solutions
Products
Services
Training
Support
Security Response
Resources
Store
AntiVirus (Endpoint Protection)
AntiVirus Corporate Edition
Critical System Protection
Endpoint Encryption
Endpoint Protection
Endpoint Protection Small Business Edition
Endpoint Protection for Windows XP Embedded
Multi-tier Protection
Network Access Control
Network Access Control Starter Edition
On-Demand Protection for Outlook Web Access
On-Demand Protection for Web Applications
Protection for SharePoint Servers
Mobile AntiVirus for Windows Mobile
Mobile Security for Symbian
Mobile Security Suite for Windows Mobile
lc
l1
l2
l3
p1
p2
p3
OP(lc)
l9
l10
l11
l12
l4
l5
l6
l7
l8
p5
p4
p1
C(lc)
The common page set
(e.g., homepage)

Figure 2. The semantic linkage analysis for hierarchical link identification

4.1 Basic Rule

Assuming y (e.g., a link collection lc or page p) contains a set of outbound links, we define $OP(y)$ is a set of outbound pages of y, i.e., $OP(y)= \{d(l)|l\in y\}$. Each page $p'\in OP(y)$ is an outbound page of y.

If p' is an outbound page of p, and p belongs to certain page set P_x, i.e., $p\in P_x$, p' is also called an outbound page of P_x.

In general, we say p' is a *common* outbound page of P_x iff each page in P_x has a link pointing to p'. But for the case that $p'\in P_x$, if all pages in P_x except p' itself have a link pointing to p', we also say p' is a *common* outbound page of P_x.

Formally, p' is a *common* outbound page of P_x iff any one of the following conditions holds:

$$(1)\ p' \in P_x: \quad \forall p \in P_x - \{p'\} : \exists l, s(l) = p \wedge d(l) = p'$$
$$(2)\ p' \notin P_x: \quad \forall p \in P_x : \exists l, s(l) = p \wedge d(l) = p'$$

For example, in Figure 2, $lc=\{l_1, l_2, l_3\}$ is a LC in the example web page. For lc, $OP(lc)= \{p_1, p_2, p_3\}$. The link l_4 is an outbound link of p_1 and points to page p_4, then p_4 is an outbound page of p_1. Since $p_1\in OP(lc)$, p_4 is an outbound page of $OP(lc)$. Also, $p_4\notin OP(lc)$, and it is the outbound page of each page in $OP(lc)=\{p_1, p_2, p_3\}$, then p_4 is a common outbound page of $OP(lc)$. Similarly, p_5 is a common outbound page of $OP(lc)$. For $p_1 \in OP(lc)$, each page in $OP(lc)–\{p_1\}=\{p_2, p_3\}$ has a link pointing to it. Although there is no link p_1 pointing to itself, p_1 is also a common outbound page of $OP(lc)$.

Now, we use the example to introduce the basic intuition for semantic link analysis: In Figure 2, the *lc* contains three links, l_1, l_2, and l_3. Page p_4 is a common outbound page of $OP(lc)$. Intuitively, pages p_1, p_2, p_3 are likely to be siblings in a hierarchically organized website, since the links in a single LC point to these pages. Furthermore, since all these pages points to p_4, it is likely that p_4 is an ancestor of pages p_1, p_2, p_3 in the hierarchy, i.e., links l_4, l_5, l_6 are all back links. Therefore, we can label l_4, l_5, l_6 as navigational links. The basic rule for navigational link detection can be derived directly from this intuition.

Given *p* and $lc \in coll(p)$ with $|OP(lc)|>1$, all the common outbound pages of $OP(lc)$ make up $OP(lc)$'s common outbound page set $C(OP(lc))$, i.e., $C(OP(lc))=\cap_{p\in OP(lc)}OP(p)$, we have following rule to detect navigational links:

$$\text{link } l\in TL(lc)=\{l' \mid l'\in L-(RL\cup NL')\wedge s(l')\in OP(lc)\} \text{ is a navigational link if } \exists\, p'\in C(OP(lc)) \text{ such that } p'=d(l) \qquad (1)$$

For the given $lc\in coll(p)$, $TL(lc)$ is the target link set of the rule (1), where $l'\in L-(RL\cup NL\_syn)$ means that this rule is applied after the syntactical analysis phase, and $s(l')\in OP(lc)$ means that this rule only considers the outbound links of the pages in $OP(lc)$.

By applying rule (1) in our example, we can determine that l_{10}, l_{11}, and l_{12} are navigational links, since they share a common destination page p_5. Similarly, l_4, l_5, l_6 (pointing to common outbound page p_4) and l_7, l_8 (pointing to common outbound page p_1) are all navigational links (Actually, l_7 and l_8 are the links between sibling pages).

4.2 Extension for LC with only one link

We should notice that the assumption to make the basic intuition reasonable is that $|OP(lc)|>1$, i.e., there are multiple links in the *lc*. However, there are some cases that *lc* contains only one link, i.e., $|OP(lc)|=1$ (when *lc* contains multiple links to the same destination page, we consider them to be a single link).

For example, in Figure 3, lc_2 in page *p* contains only one link pointing to p_3, i.e., $OP(lc_2)=\{p_3\}$. Page p_3 has three outbound links l_8, l_9, l_{10} pointing to p_5, p_6, p_7, respectively. Then the common outbound page set $C(OP(lc_2)) =\{ p_5, p_6, p_7\}$, i.e., it contains all the outbound pages of p_3. By rule (1), all the outbound links of p_3, i.e., l_8, l_9, l_{10}, should be navigational.

However, given a *lc* with only one link *l*, it is unlikely that all outbound links of page $d(l)$ are navigational.

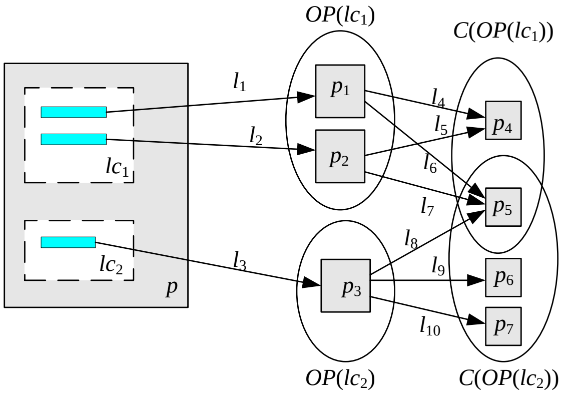


Figure 3. The schematic diagram on the link collection with only one link

To make rule (1) still applicable in such cases, given *p* with $lc\in coll(p)$, our implementation uses following formulas to calculate $C(OP(lc))$:

$$C(OP(lc))=\begin{cases} \bigcap\limits_{q\in OP(lc)} (OP(q)\cup\{q\}) & |OP(lc)|>1 \qquad (a) \\ \bigcap\limits_{q\in OP(p)} (OP(q)\cup\{q\}) & |OP(lc)|=1, |link(p)|>1 \qquad (b) \\ \phi & |OP(lc)|=1, |link(p)|=1 \qquad (c) \end{cases} \qquad (2)$$

Formula (*a*) is defined for our earlier case when $|OP(lc)|>1$. Formula (*b*) and (*c*) are defined for the cases that there is only one link in *lc*, i.e., $|OP(lc)|=1$.

- When there are multiple links in *p*, i.e, $|link(p)|>1$, Formula (*b*) is adopted. We use the example in Figure 3 to illustrate the basic idea of Formula (*b*). The link collection lc_2 in page *p* has only one link l_3. We assume all the links in *p* make up a virtual link collection $lc_v = link(p)$. In our example, $lc_v=\{l_1, l_2, l_3\}$. For lc_2, we use lc_v to replace lc_2 to calculate corresponding $C(OP(lc_2))$, i.e., $C(OP(lc_2))= C(OP(lc_v))=\{p_5\}$.

- Otherwise, $|link(p)|=1$, Formula (c) is adopted, i.e., $C(OP(lc))=\phi$. It means the intuition for semantic hyperlink analysis is not applicable for the page with only one outbound link.

The introduction of the virtual LC will set conflicting signal on the navigational links judgment. For example, in Figure 3, when lc_1 is analyzed, l_4, l_5 , l_6, l_7 are classified as navigational links; when analyzing lc_2, l_6, l_7, l_8 are classified as navigational, and l_4, l_5 are not classified as navigational. For such a case, since lc_v was introduced to analyze lc_2, we use the lc_1 analyzing for the links related to lc_1. Therefore, we determine that l_4, l_5 are navigational links.

In the implementation, for all the LCs with one link in a page p, we only need to calculate one $C(OP(lc_v))$ and scan one time the links in the target link set $TL(lc_v)$. The reason is that, for all these LCs, i.e., $\{lc|lc\in coll(p)\wedge|lc|=1\}$, they share the same lc_v, $C(OP(lc_v))$, and $TL(lc_v)$.

4.3 Refinement

We notice that in some real websites, the intuition for semantic link analysis doesn't hold. For example, in a company website, a list of software products might share a common child page about "system requirements". For such a case, since the "system requirements" page is one part of the description on each of the software product, it doesn't make sense to say that the links from different product pages to the "system requirements" page are navigational. Then, an additional constraint should be employed.

Given p and $lc\in coll(p)$, say p' is a common outbound page of $OP(lc)$ (e.g., in Figure 2, p' might be p_5 or p_4). Each page in $OP(lc)$ has a link pointing to p' (e.g., l_4, l_5, l_6 pointing to p_4, and l_{10}, l_{11}, l_{12} pointing to p_5). There are mainly two reasons for these links pointing to p': (A) p' contains general/important information and is a page higher in the website hierarchy; (B) The content of p' is semantically close only to the topic presented by the pages in $OP(lc)$. In general,

i) If the reason is (A), i.e., p' is a page higher in the hierarchy and contains general/important information, there should also be a link from p to p' or $p=p'$. For this case, the main role of the links pointing to p' is to provide shortcut to p', i.e., they are navigational links. E.g., for page p_5 in Figure 2, not only l_{10}, l_{11}, l_{12} pointing to it, but also there is a link l_9 from p to it. Then, page p_5 is more likely to be a page higher in the hierarchy, and links l_{10}, l_{11}, l_{12} pointing to p_5 are navigational.

ii) If the reason is (B), i.e., p' is only relevant to the pages in $OP(lc)$ and might not be a page higher in the hierarchy, there would be no link from p to p'. For this case, the main role of these links pointing to p' is to organize p' with the pages in $OP(lc)$ together, i.e., they are not navigational links. For example, for page p_4 in Figure 2, only l_4, l_5, l_6 pointing to it, and there is no link from p to it. Then, page p_4 might not be a page higher in the website hierarchy, and links l_4, l_5, l_6 pointing to p_4 are not navigational.

In our example shown in Figure 2, pages p_5 or p_4 are both common outbound pages of $OP(lc)$. Link l_9 is the key evidence to differentiate the links pointing (from pages in $OP(lc)$) to them as navigational or not.

Based on this observation, we can say that, given p and $lc\in coll(p)$, the pages in $OP(lc)$ share a common outbound page p', if there is a link from p to p' or $p=p'$, the links from the pages in $OP(lc)$ to p' are navigational links. Correspondingly, the rule (1) is modified as:

$$\text{link } l\in TL(lc)=\{l'|\, l'\in L-(RL\cup NL')\wedge s(l')\in OP(lc)\} \text{ is a navigational if } \exists\, p'\in C(OP(lc)) \text{ such that } d(l)=p'\wedge(p'\in OP(p)\vee p'=p) \quad (3)$$

In this example, since there is link l_1 pointing to $p_1\in OP(lc)$, l_7 and l_8 pointing to p_1 are actually the links between sibling pages in $OP(lc)$. It also implies that l_7 and l_8 assume the role of navigational links.

Note that, given p and $lc\in coll(p)$, the semantic link analysis only determine whether the outbound link of the pages in $OP(lc)$ is navigational link or not. So, for all the outbound links of the homepage $home(\Delta)$, if they are not inter-site links (i.e., syntactical-reference links) and don't point back to $home(\Delta)$ (i.e., syntactical-navigational link), they are considered as hierarchical links by default.

5. HNPs construction

So far, given a website $W=<P, L, \Delta>$, we have classified the links in L into reference links, navigational links, and hierarchical links. Since our final goal is to use the HNP to improve web page retrieval, this section will use the hierarchical links to construct HNPs for the web pages in P.

A path can be formally represented as $\tau=<l_1, l_2, \ldots, l_n>$, where l_i ($1\leq i\leq n$) is a hierarchical link, $d(l_i)=s(l_{i+1})$, $s(l_1)=home(\Delta)$, and $d(l_n)$ is the destination page of τ. Each HNP is associated to its destination page $d(l_n)$.

Since a path is constructed from a sequence of hierarchical links, no cycle is allowed in a HNP τ, i.e., $l_i \neq l_j$ for any $i\neq j$. If a HNP contains n links, we say its length is n, i.e., for $\tau=<l_1, l_2, \ldots, l_n>$, $length(\tau)=n$.

Our goal is to build all possible hierarchical paths, starting at $home(\Delta)$. Then for each page $p\in P$, we use all paths that end at p as “description” of p, as we discuss later.

We adopt a two-step approach to construct all possible paths and to guarantee that each page in P has at least a path associated with it.

In the first step, the basic path construction algorithm uses the output HL of the link classification algorithm in Section 4 to construct as many paths as possible.

One page might have multiple associated paths (with different lengths). In general, the longer path contains lower quality indications of the page content. In the basic path construction algorithm, we give a parameter k as the predefined maximum length for the constructed paths. The bigger k means lower quality but more constructed paths for a page and longer execution time of the algorithm. The smaller k means higher quality but less constructed paths for a page and shorter execution time of the algorithm.

After the running of the basic path construction algorithm, some pages in P may have no paths associated with them.

The reason is that the unexpected errors from web page crawling and parsing or the incorrect judgments of the hierarchical link discovery algorithm might cause some pages not to have an identified hierarchical path leading to them.

Therefore, the second step relaxes the criteria of hierarchical links and considers some navigational links as hierarchical links to make sure each web page has at least one path associated with it.

5.1 Basic path construction algorithm

The pseudo code of the first step of the path construction algorithm is described as below:

Path construction algorithm: *Step* 1

Input: $W=<P, L, \Delta>$, and HL= the set of hierarchical links outputted by the link classification algorithm.
Output: a set of path *HNP*, and a set of hierarchical links *HL_unused* that not employed for path construction in the algorithm running

```
 1. $HNP'=\phi$; $HNP=\phi$; $HL\_unused=\phi$; $HL\_used=\phi$
 2. For each $l$ with $s(l)=home(\Delta)$ do
 3.      $HNP= HNP\cup\{<l>\}$ and $HL\_used = HL\_used \cup\{l\}$
 4. $HNP'= HNP$;
 5. while $HNP'\neq\phi \wedge length(\tau\in HNP')<k$ do
 6.      $HNP''=\phi$;
 7.      For each $\tau=<l_1, l_2, \ldots, l_n>\in HNP'$ do
 8.           For each $l\in HL$ do
 9.                if $s(l)=d(l_n)$ and $l$ not in $\tau$, $HNP''= HNP''\cup\{<l_1, l_2, \ldots, l_n, l>\}$ and $HL\_used = HL\_used \cup\{l\}$
10.      $HNP= HNP\cup HNP''$
11.      $HNP'=HNP''$
12. end while
13. $HL\_unused=HL-HL\_used$
14. return $HNP$
```

In this algorithm, *HL_used* is the set of hierarchical links employed for path construction.

Lines 2-3 use outbound links of home page $home(\Delta)$ to initialize *HNP*. During each iteration from line 4 to 11, a set of paths HNP'' is generated from the set of paths HNP' generated from last iteration. Each path in HNP'' actually is the extension of a path in HNP' by a hierarchical link whose source page is the destination page of the HNP in HNP'. *HL_used* is the set of hierarchical links used for HNP construction. The maximum path length *k* could be set to ∞. In that case, because no cycle is allowed in a path, the iteration will terminate by itself.

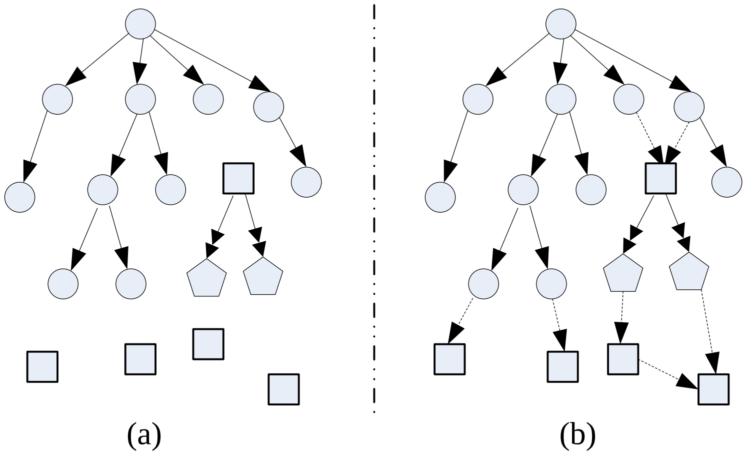


Figure 4. The schematic diagrams on the input and output of the compensation algorithm

For example, Figure 4(a) is the processing result of the basic path construction algorithm, where, for each web page, if there is a path starting from root node (i.e., the homepage) and ending by it, we say there is a path associated with it. Obviously, the squares and pentagons are web pages without associated paths. The squares are web pages without hierarchical link pointing to them. The double-head arrows are the hierarchical links not used for path construction.

5.2 Compensation for path construction

Assume *P_nohl* is the set of web pages without hierarchical link pointing to them (the squares in Figure 4), *P_nop* is the set of web pages without paths associated with them (the squares and pentagons in Figure 4), and *HL_unused* is the set of hierarchical links not employed for paths construction in the first step of path construction algorithm (the double-head arrows in Figure 4), the pseudo code of the second step of path construction algorithm are given as below:

Path construction algorithm: Step 2

Input: *HNP*, *P_nohl*, *P_nop*, *HL_unused*, *NL_syn*
Output: $\overline{HNP}$

```
 1. $\overline{HNP}$=$HNP$; $EHL=\phi$
 2. For each $p \in P\_nohl$ do
 3.      $EHL = EHL \cup \{l | d(l)=p \wedge l \in L - NL\_syn\}$
 4. $\overline{HL} = HL\_unused \cup EHL$
 5. While $P\_nop \neq \phi$ do
 6.      For each $l \in \overline{HL}$ with $d(l)=p \in P\_nop$
 7.           For each $\tau=<l_1, l_2, \ldots, l_n> \in HNP$
 8.                if $d(l_n)=s(l)$ and $l$ not in $\tau$, $\overline{HNP} = \overline{HNP} \cup \{<l_1, l_2, \ldots, l_n, l>\}$
 9.           $P\_nop = P\_nop - \{p\}$
10. end While
11. return $\overline{HNP}$
```

In the second step of path construction algorithm, Lines 2-3 deals with the web pages without hierarchical links pointing to them. All the inbound links of these web pages that not been judged as navigational links by the syntactical URL analysis are considered as an extended set *EHL* of hierarchical links. Line 4 merges the *HL_unused* and *EHL* into $\overline{HL}$. Then, an iteration process is utilized to remove the web page from *P_nop*. During each iteration, if the source page $s(l)$ of the l in $\overline{HL}$ have associated paths, the paths of *l*'s destination pages can be constructed from the paths of $s(l)$. Since no cycle is allowed in a path, the iteration will terminate. An empty *P_nop* means that each page has at least one associated path.

As an example, Figure 4(a) is the input of second step path construction algorithm. Figure 4(b) is corresponding output, where the broken line arrows are the set of links *EHL* generated by Line 2-3. $\overline{HL}$ is the set of broken line and double-head arrows.

Since the notions of hierarchical and navigational links depend on the intent of the website creator, the output of our algorithm can only be considered an “educated guess”. However, as we will see in the following sections, these educated guesses can indeed improve web search performance.

6. PathRank for Web Page Ranking

Until now, the paths are constructed for each page in *P*. This section presents our PathRank algorithm to exploit these paths for web page retrieval.

Since a path will be utilized for discovering the semantic meaning of its destination page, the linguistic contents of the path, including the URLs, anchor text, and web page titles along it, are concatenated. Given a page *p*, we define $\delta(p)=title(p)\cdot url(p)$, where the symbol “·” represents concatenation. For a path $\tau=<l_1, l_2, \ldots, l_n>$, which is associated to page $d(l_n)$, we define $\delta(\tau)=\delta(s(l_1))\cdot at(l_1)\cdot\delta(s(l_2)\cdot at(l_2)\cdot\ldots\cdot\delta(s(l_n)\cdot at(l_n)\cdot\delta(d(l_n))$.

Assume there are *m* paths ending at page *p*, i.e., paths $\tau_1, \tau_2, \ldots, \tau_m$. Given query $q=\{term_1, term_2, \ldots, term_u\}$, our PathRank algorithm (using the *m* paths as the “description” of *p*) employs three steps to calculate the similarity scores between *p* and *q*: 1) query-independent score calculation; 2) query-dependent score calculation; 3) the combination of query-dependent and query-independent scores.

6.1 Query-independent score calculation

This step uses the reference links to obtain an importance ranking of paths based on page-rank.

Firstly, we construct a Web graph, where each node is a website, and the links between nodes are reference links (website *x* links to website *y* if any page in *x* has a reference link to any page in website *y*). Secondly, we compute the rank value R_W of each website *W* by applying the conventional PageRank algorithm. Then, we assign value R_W to all of the paths in website *W* to represent their page-rank importance.

6.2 Query-dependent score calculation

This step calculates the relevance score $R(\delta(\tau), q)$ between a query *q* and a path τ.

Intuitively, the text in different positions of the sequence $\delta(\tau)$ should have different weights. We weight the text in $\delta(\tau)$ that come from the same page equally. Thus, we divide $\delta(\tau)$ as text nodes $tn_1, tn_2, \ldots, tn_n, tn_{n+1}$, where $tn_1=\delta(s(l_1))$, $tn_i=at(l_{i-1})\cdot\delta(s(l_i))$ for $2\le i\le n$, and $tn_{n+1}=at(l_n)\cdot\delta(d(l_n))$.

For simplicity, we define tn_i’s weight as $w(tn_i)= 1/(n-i+2)$, meaning that the closer is tn_i to tn_{n+1}, the higher is the weight of tn_i. For tn_{n+1}, the weight is 1.

Given $\delta(\tau)= tn_1\cdot tn_2\cdot,\ldots\cdot tn_n\cdot tn_{n+1}$ and $q=\{term_1, term_2, \ldots, term_u\}$, the formula for computing $R(\delta(\tau), q)$ is:

$$R(\delta(\tau),q)=\alpha\frac{\sum_{i=1}^{n+1}w(tn_i)sim(tn_i,q)}{n+1} \qquad (4)$$

where $w(tn_i)$ is the weight of tn_i; $sim(tn_i, q)$ is the BM25 score [33] between tn_i and *q*; α is the proportion of *q*’s terms appearing in $\delta(\tau)$. Assuming there are *u* terms in *q* and *v* of them appear in $\delta(\tau)$, $\alpha=v/u$.

6.3 Score combination

After R_W and $R(\delta(\tau), q)$ are obtained, we can combine them to obtain the similarity score between page *p* and *q*.

We use R_W to weight $R(\delta(\tau), q)$ to combine the query-dependent and query-independent scores. Assume there are *m* paths ending at page *p* (in website *W*), i.e., $\tau_1, \tau_2, \ldots, \tau_m$. Then the similarity scores between *q* and *p* is given by the normalized similarity between *q* and all the paths pointing to *p*:

$$R(p,q) = R_W \frac{\sum_{i=1}^{m} R(\delta(\tau_i), q)}{m} \tag{5}$$

Note that each path associated with the web page makes a contribution to the final rank value. If the information appearing on each path is viewed as a summary of the content of the destination web page, multiple points of view from multiple authors or contexts can be reflected through this set of paths.

7. EXPERIMENT AND EVALUATION

In this section, we empirically verify the benefits of the proposed path-based approach.

7.1 Experiments on website search

This study first evaluates the quality of the constructed paths, and then evaluates informational and navigational queries [2] results.

7.1.1 Data set and Query set

We have studied dozens of publicly accessible websites and select stanford.edu as one of the most representative websites to simulate web search, where the inter-subdomain links are treated as reference links and utilized to rank the subdomain site within stanford.edu. A breadth-first crawling strategy [24] is configured in our crawler. The maximum link hops followed by the crawler is set to 15. We collected about 2 million web pages from 2,980 subdomains of stanford.edu. After pruning away web pages from the subdomains containing fewer than 20 pages and duplicate web pages, about 1.4 million unique pages from 768 distinct subdomains remain.

We hired 18 students to help us create a query set. They were divided into two groups respectively for the navigational and informational queries.

For navigational queries, we asked the students to browse the Stanford.edu website and to select 10-20 topics of interest to them (e.g., persons, services, projects, etc.). All the proposed topics are ranked by voting based on the topic's popularity among our subjects. (Given a list of topics, if a student knows a topic or finds the topic interesting, he will give a positive vote. The topic with the highest number of votes has the top rank.) The top 50 topics are selected. For each topic, the students collectively created a query phrase for it. (A query phrase is a set of terms that someone might use as a query to find corresponding page.) First, each student gives one or several terms to describe the query phrase that he may use to find the page; then, all the query phrases are ranked by voting, and the query phrase with highest voting rank is selected as the query phrase for the topic. For each of these topics, its homepage (i.e., the main page of the topic that generally contains a brief overview and navigational links to more detailed descriptions of the topic) is found and defined as ground truth.

For the informational queries, we first asked each student to specify 5-10 topics related to Stanford University that he is interested in, e.g., the topics on their disciplines or personal interests. Then, these topics are ranked on their popularity among our subjects and whether the relevant descriptions/answers of the topics can be found in the standford.edu website. (If a student thinks a topic is interesting and the relevant information of this topic is available in Stanford.edu website, he will vote for it.) For each of the top 50 topics, the query phrase is created using the same method mentioned above for navigational queries.

7.1.2 Evaluation criteria

Because paths play a key role in the subsequent web page retrieval, in this experiment, we evaluate not only the search results, but also the quality of the constructed paths.

To evaluate the constructed path, two subjective perspectives are considered:

1) Whether the identified paths are Entrance Paths (EPs) that are potentially assumed by website builder to guide the reader (or commonly used by the reader to navigate) from the website's homepage to the path's destination page. For example, the homepage of a university website contains a link to identify one of its departments, which in turn contains a link to identify a faculty member. Such a sequence of links from this homepage to the faculty member's page is a potential EP.

2) Whether the paths are Useful Paths (UPs) that contain useful summary information on the content of the paths' destination pages. For example, if a path ending at the main page of a Computer Science faculty member contains text such as "computer science" and/or "professor", it is a UP.

Correspondingly, following measures are employed:

Precision of EPs: The precision for EPs is defined as the proportion of the constructed paths which are exactly the EPs confirmed by a human.

Recall of EPs: the proportion of the EPs which are contained in the constructed path results.

F-measure of EPs: conjunction with Precision and Recall for EPs, as a harmonic average of them:

$$F-measure=\frac{\left(\beta^2+1\right)\Pr ec*\operatorname{Re}c}{\left(\beta^2\operatorname{Re}c\right)+\Pr ec},$$

where β reflects the weighting of Precision vs. Recall. For simplicity, it is set to 1 here.

Precision of UPs: Precision for UPs is defined as the proportion of the constructed paths which are UPs as determined by a human evaluator. UPs would have positive effectiveness for path-based web page retrieval.

We randomly sample 500 pages from the data set to evaluate the path quality. The precision of the EPs and UPs can be measured by human’s judgement on whether a path is EP/UP or not. To measure the recall of EPs, we need to collect all the possible EPs: 1) we first asked the students to browse the stanford.edu site to find the EPs (starting from the homepage of stanford.edu site) as many as possible for each of the 500 pages; 2) We asked the students to select all the EPs (ending at each of the 500 pages) from the set of paths automatically constructed by the algorithm; 3) The union of the two sets of EPs respectively from 1) and 2) serves as the ground truth for EPs. Note that the resulting ground truth may not be a complete set of EPs (some EPs might not be found by students and our algorithm), thus the evaluation of the recall of EPs is an approximate measure.

For the web search, precision evaluation is the main goal of the experiment. The following criteria are utilized in our experiment:

S@5 (S@10) for navigational query: the proportion of queries for which the correct answer (i.e., the homepage for the corresponding topic defined in the ground truth for navigational queries) is ranked in the top 5 (10) in the ranked list returned for the query

P@10 (P@20) for informational query: it is the proportion of relevant web pages (Whether a returned page is relevant to the query is determined by a human) in the top 10 (20) web pages in the ranked list returned for the query.

SP: is used to evaluate the overall quality of the approach for the website search. It’s an average of the precision on navigational and informational queries:

$$SP=\frac{\gamma(S@5)+(1-\gamma)(P@10)}{2},$$

where γ reflects the weighting of navigational query vs. informational query. For simplicity, it is set to 0.5 here.

7.1.3 Experiment results

1) Quality evaluation of the constructed path

The link classification and path construction algorithms are applied on the page set with domain stanford.edu (The entire stanford.edu is considered a website). The setting of the maximum length of paths plays an important role in making a tradeoff between the quality (shorter path means high quality) and quantity (longer path means larger quantity and more running time) of the resulting paths. In this experiment, we run the path construction algorithm six times, with maximum length 4, 5, 6, 7, 8 and unlimited, respectively. The resulting paths of the 500 sample pages are selected, which have the amount of 2,253, 2,310, 2,671, 3,225, 4,009 and 5,516 paths, respectively. Then, the corresponding quality measures of these resulting paths are calculated.

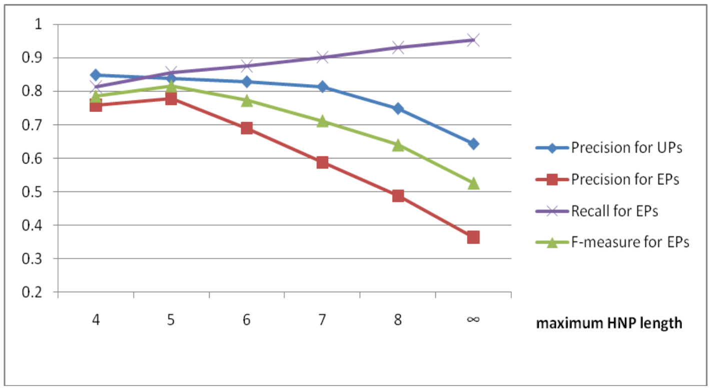


Figure 5. The evaluation of path extraction

Figure 5 shows the evaluation result of path discovery. Basically, the accuracy of the resulting paths of the link classification and path construction algorithms is satisfactory. Futhermore, it can be observed that the quality is closely related to the set maximum length of paths. Because the stanford.edu website is quite large, many pages can only be visited by multi hops from the root page. Therefore, as the results show, the recall for EPs are higher when the maximum path length is set larger. On the other hand, because the longer paths are generated from the shorter ones in path construction, the error of the shorter paths might be propagted to the longer ones. We can observe that when the maximum length is greater than 7, the quality of paths decreases quickly. Therefore, in the following experiment, we set the maximum length as 7 for path construction.

Note that due to the compensation step, even though the maximum length of path is set to 7, the length of some paths is longer than 7.

2) Web search evaluation

We compare our approach against two existing state-of-the-art methods for web page retrieval:

BM25: The similarity between queries and web pages is calculated based on the BM25 formula [33]. Given a page, we extract information and store the result in two fields: content and metadata. A page's content is represented by all text within the <body> tags. The anchor text of the links pointing to this page and the page title constitute its metadata. Their BM25 scores are combined as BM25= 0.7×BM25_content + 0.3×BM25_metadata.

BM25+PageRank: The query-independent measure is considered. We constructed a graph by considering each page in Stanford.edu is a node and a link between any two pages as an edge. Then the conventional PageRank algorithm is applied to obtain the importance ranking of the page. The linear combination of BM25 and PageRank is 0.8×BM25+0.2×PageRank.

In addition, we also consider the results from the website search engine at stanford.edu, which is powered by Google.

We implemented three versions of our PathRank approach for web page retrieval:

PathRank1 (path-FullText): PathRank1 means that our PathRank method is used to rank the pages, where the entire stanford.edu site is considered as a website. (The importance values R_W for all paths are set to 1.) path-FullText serves as a comparison method for our PathRank algorithm, where all the texts in the constructed paths of a page compose a representative document, and then the BM25 similarity between the query and the page's representative document is utilized for web page ranking.

PathRank1+BM25 (path-FullText+BM25): The page content and metadata (BM25= 0.7×BM25_content + 0.3×BM25_metadata) are considered. For PathRank1+BM25, the linear combination is 0.5×PathRank1+0.5×BM25. For path-FullText+BM25, the linear combination is 0.4×path-FullText + 0.6×BM25.

PathRank2 (path-FullText): PathRank2 means that our PathRank method is used to rank the web pages, where each subdomain (e.g., cs.stanford.edu) of stanford.edu is considered as a website (The paths are constructed within each subdomain, and the importance values R_W of the paths are calculated from the reference links across the subdomains).

PathRank2+BM25 (path-FullText+BM25): The page content and metadata are considered. Their linear combination is 0.5×PathRank2+0.5×BM25 (0.4×path-FullText +0.6×BM25).

Note that the parameters for the above linear combination (in both the existing state-of-the-art methods and our PathRank approach) were obtained empirically by tuning not described here.

Table 3. The evaluation of stanford.edu website search

	Navigational queries		Informational queries		Overall
	S@5	**S@10**	**P@10**	**P@20**	**SP**
BM25	0.43	0.52	0.79	0.71	0.61
BM25 + PageRank	0.59	0.68	0.80	0.72	0.70
stanford.edu website search	0.64	0.74	0.82	0.79	0.73
PathRank1 (path-FullText)	0.78(0.73)	0.86(0.77)	0.75(0.71)	0.69(0.64)	0.76(0.72)
PathRank1 +BM25 (path-FullText+BM25)	0.81(0.75)	0.90(0.79)	0.83(0.79)	0.72(0.69)	0.82(0.77)
PathRank2 (path-FullText)	0.85(0.79)	0.91(0.82)	0.77(0.73)	0.71(0.69)	0.81(0.76)
PathRank2+BM25(path-FullText+BM25)	0.91(0.85)	0.92(0.87)	0.89(0.81)	0.79(0.72)	0.90(0.83)

For stanford.edu site, the number of the extracted paths is 8.6M and 7.8M by PathRank1 and PathRank2, respectively. On average, there are about 6 and 5.5 paths for one page.

Given the 50 navigational and 50 information queries created by the students, the web pages in the data set from Stanford.edu are ranked by using the baseline techniques and our proposed PathRank approach. The following evaluation is conducted by the students: 1) For a navigational query, whether the homepage defined in the ground truth for the corresponding navigational query is returned in the top 5 (10) hits; 2) For an informational query, whether a returned page in the top 10 (20) is relevant to the corresponding informational query. Then, the values of **S@5 (S@10)** for navigational queries, **P@10 (P@20)** for informational queries, and overall search result quality **SP** are calculated. Table 3 summarizes the evaluation results of the web page retrieval experiments. The figures in the shaded rows are the only query-dependent measures. For others, the query-independent features are adopted. Among the three baselines (i.e., BM25, BM25+PageRank, website search engine), the website search engine has the best results.

Among the multiple version implementations of our PathRank approach, PathRank2+BM25 has the best performance. It can improve the search quality remarkably (utilizing two-tailed t-test with p-value=0.01), especially for the navigational queries. From the **SP** values, quantitatively, our approach shows improvements by as much as 14% compared with that of the website search engine. Actually, for the example query "computer science alumni" given in Section 1, the web pages on "Undergraduate Alumni", "Masters Alumni", and "Ph.D Alumni" from cs.stanford.edu are ranked in Top 5 by our approach. In contrast, the top 100 hits returned by the stanford.edu website search engine contain no direct link to any alumni list pages.

The subdomains of stanford.edu are relative independent of each other. Since PathRank2 incorporates the website rank R_W inside, it performs better than PathRank1. The experiments also show that, comparing with navigational queries, page contents play a more important role in informational queries.

An interesting phenomenon found from our experiments is that, as the number of keywords contained in a query increases, the performance of PathRank improves relative to the baseline solutions. In general, the contained keywords of a query are distributed across several nodes of one or more paths, where the inference is incorporated implicitly into the web page ranking process (e.g., the three keywords of the query "computer science alumni" are distributed in two adjacent nodes of multiple paths, the semantic "*the alumni information of computer science department*" can be inferred connotatively from these paths).

Generally, the experiment results in Table 3 demonstrate that the contextual information propagated across multi-steps of hierarchical links can improve the web search quality. The two kinds of links, structural and reference, should be distinguished when exploiting them for web page ranking.

7.2 Experiments on TREC WT10G

This section uses the WT10G, a public test collection used by TREC, to evaluate the performance of the proposed PathRank approach.

7.2.1 Tasks

WT10G contains 10GB of data, 1.69M web pages and 11.68K sites. We conducted two experiment tasks.

The first task was to retrieve topic relevant pages from the test collections. The 100 queries used in the experiment are from two web search specific tasks conducted on WT10G: 50 of them are the topics 451-500 of the TREC-9 main web task [43]; and another 50 are topics 501-

550 of the TREC-2001 topic relevance task [44]. The goal of this experiment is to evaluate the performance of our PathRank algorithms on informational web search (we have checked the descriptions and narratives of the 100 topics for querying, almost all of them are informational queries).

Another task was to retrieve the homepage (the main page) for a specific topic, where the topics used were from the 145 topics of the TREC-2001 homepage finding task [44], and the queries were generated from the <desc> field of each topic. The goal of this task is to evaluate the performance our PathRank approach on navigational web search.

7.2.2 Evaluation criteria

We selected standard TREC measures as evaluation criteria of our algorithms. For the topic relevance task, the precision of an algorithm is measured by the average precision over all relevant pages **AveP**; and document-level averages **P@5**, **P@10**, and **P@20**, which represents the precision of the top-5, top-10 and top-20 results, respectively. For the homepage finding task, three measures are used: **MRR**, the mean reciprocal rank of the first correct answer for each topic query; **%top10**, the proportion of queries for which a right answer was found in the top 10 retrieval results; and **%fail**, the proportion of queries in which no right answer was found in the top 100 retrieval results.

7.2.3 Experiment results

The linear combination 0.5×PathRank + 0.5×BM25 is used here. The maximum length of paths is set to 7. By the definition, a text node in a path is composed of three elements: page title (denoted by *t*), URL string (denoted by *u*) and anchor text (denoted by *a*). In order to investigate their influences on the search results, we conducted three experiments each of which uses only one kind of element (*t* or *u* or *a*) to represent path nodes. TREC best results and the existing state-of-the-art BM25+PageRank method (0.8×BM25 +0.2×PageRank) are selected as baselines.

Table 4. The evaluation on WT10G

	TREC-9 main web task				**TREC-2001 topic relevance task**				**TREC-2001 homepage finding task**		
	AveP	**P@5**	**P@10**	**P@20**	**AveP**	**P@5**	**P@10**	**P@20**	**MRR**	**%top10**	**%fail**
PathRank*(t)*	0.1721	0.3840	0.3540	0.3460	0.1894	0.2320	0.2348	0.2343	0.522	66.4	23.9
PathRank*(u)*	0.3075	0.5002	0.4813	0.4574	0.2938	0.2936	0.2775	0.2517	0.515	80.4	7.2
PathRank*(a)*	0.4529	**0.5812**	0.5723	0.5569	0.4027	**0.4710**	0.4503	0.4459	0.797	89.2	5.1
PathRank*(t+u+a)*	**0.4784**	0.5651	**0.5919**	**0.5881**	**0.4152**	0.4066	**0.4799**	**0.4703**	**0.826**	**91.6**	**4.3**
TREC best result	0.3519	/	0.5180	/	0.3324	0.4320	0.3620	0.3130	0.774	88.3	4.8
BM25+PageRank	0.1862	0.4012	0.3769	0.3251	0.1510	0.2302	0.2239	0.2175	0.732	85.2	6.8

The evaluation results of topic relevance task and homepage finding task are shown in Table 4. Basically, our PathRank method considering all the three elements (i.e., page title, URL, and anchor text) has the best performance. It outperforms TREC best results and existing BM25+PageRank for both tasks. And also we observe that anchor texts in paths play the most important role for the tasks (this result is consistent with the observation in [27]), while the page titles and URL strings could also bring positive influence when combined.

This experiment shows again that the two kinds of links, respectively for recommendation and information organization, should be distinguished when using them for web page ranking.

8. RELATED WORK

While in traditional information retrieval only the document content is of concern, in web page ranking, the link structure of the Web also plays an important role. Current popular models for web page retrieval are mainly combinations of content-based and link-based approaches.

A content-based approach, which is derived directly from traditional information retrieval technology, mainly employs the internal information of a web page to measure the similarity between the query and a web page. Since a web page represents information in its own way, the web page usually comprises much more diverse information compared with that in a plain-text document. Based on page segmentation algorithms such as those in [6][23], the web page is partitioned into blocks, and the layout structure inside of a web page is investigated to improve retrieval performance [8]. Similarly, the textual content of a web page together with its structural characteristics are employed in rule-based [21] or learning-based [12] classifiers for web page type classification. Reference [36] proposed a method using the page titles to help improve web page retrieval. The Web provides additional tagging information to extend the textual content of a web page. Several approaches have been reported: The impact and contribution of anchor text in web search is studied in [27]. In [10], the anchor text together with its surrounding words, i.e., the so-called extended anchor text, is utilized to improve the web page classification. Recently, [32] investigated the capability of social annotations [9] in improving the quality of web search.

Almost all of these content-based approaches treat each web page as an independent document, i.e., use single-page based ranking. Thus the returned page must include all the query keywords. These approaches ignore the fact that the internal content of a web page, even including anchor text from other pages, is often not self-contained. Although the (extended) anchor text in some adopted models [10][22] is propagated from one page to another, this propagation is only across the link.

Link-based approaches utilize the location of a web page in the Web's graph structure to determine its importance. PageRank [22] is one of the most popular algorithms. It utilizes the stationary state of a Markov chain, which is abstracted from an assumed “random surfer” model, to assign each page a score. The intuition is that a page has high rank if the sum of the ranks of its incoming links is high. Reference [8] presented a method to extend the link analysis from page level to block-level. Also, [39] proposed an object-level link analysis model. To integrate the web page content directly into the PageRank calculation, personalized and topic sensitive PageRank schemes are reported respectively in [34] and [13]. Similarly, the HITS algorithm [17] also uses content information to enhance the link analysis. In addition to an authority score, it also assigns a hub score to a page. Recently, to handle the problem that explicit links are sparse in the global Web and local website, the work on extending the explicit links to the implicit ones for improving the quality of web page classification in the global Web and small web is reported in [14] and [7], respectively.

A basic assumption underpinning link-based approaches is that links in the global Web have the semantics of recommendation. However, this assumption does not hold in general in local websites, since a large number of links exist in a specific website for organizing the web site. Therefore, back-links cannot really reflect the importance of a web page. In fact, the Web graph analysis [1] has shown that the global structure of the Web is totally different from its local structure, and the links of the local Web are more regular than those in the global Web. In addition, the random surfer model might have reduced validity. During navigation, the anchor text co-occurred with the link as well as corresponding context appearing in the navigation path (within the reader’s mind) also provides an important guides for web browsing.

This paper mainly focuses on exploiting the huge number of hierarchical links created in the local websites to improve the web search quality, where the contextual information from one-step link is extended to multi-steps of links. Several approaches for using hierarchies have been reported: [5] utilizes the shortest paths in the intranet to organize web search results in order to reflect the underlying structure of the intranet. Also, under an assumption that each page has at most one entrance path, a entrance path extraction algorithms are presented in [4][37]. A stochastic model is given in [11] to compute the probability that a user (with a certain information need) navigates along some paths from a web page to another. This path-based model is mainly applied for user behavior predication and usability analysis of a website. Recently,a navigation-aided retrieval model was proposed in [28]. However, all these approaches are different from our work as they have not directly addressed how these links with inherent “regular” features can be effectively exploited to improve the accuracy of web page ranking.

Related to the differentiation of hierarchical links and navigational links, [38] proposed an algorithm to distinguish the *navigational* and *semantic links* (which are similar to the navigational links and hierarchical links defined in this paper) for web thesaurus building. Its basic idea is to use the directory structure embed in the URL and *navigation list* [19] to identify navigational links. However, in many cases, there is no directory information in the URLs (especially for a web page that is dynamically generated from a database). Also, this work is different from our approach as it does not address how structural links can be exploited to improve web page retrieval.

9. CONCLUSION

Despite the success of link analysis based algorithms in global web search, they have poor performance for website and small web search [14][16]. The main reason is that a large number of hierarchical links in local websites are created for web page organization and have no recommendation semantics, which make the basic assumption underlying these link analysis algorithm invalid in a local website. This paper identifies the hierarchical links for improving web page retrieval. Through our path-based technique, the multi-level link structure together with the co-occurring textual information (e.g., anchor text, page titles, URL, etc.) in the local web are exploited for high quality web page retrieval. The experimental results showed that the proposed approach can improve the accuracy of web page retrieval significantly.

The research results presented in this paper represents a beginning to exploit links with the functionality of information organization in local websites as a novel resource for web page retrieval. One limitation of the PathRank approach is that it is not appropriate for ranking the pages from the "deep Web". With the increase of the path length, the efficiency of the path extraction becomes the bottleneck to block its application in such a scenario. In the future, more sophisticated algorithms for improving paths construction and path-based web page ranking could be designed. In addition, additional experiments are needed to validate our conclusion in different application scenarios.

REFERENCES

[1] A. Broder, R. Kumar, F. Maghoul, etc., Graph structure in the web. Proc. of WWW2000.

[2] A. Broder. A taxonomy of web search. SIGIR Forum, 36(2):3–10, 2002.

[3] B. Berendt, A. Hotho, etc., A Roadmap for Web Mining: From Web to Semantic Web, LNAI, Vol. 3209. 2004, pp.1-22

[4] C. E. Dyreson. A Jumping Spider: Restructuring the WWW Graph to Index Concepts that Span Pages. Proc. of WWW1998.

[5] M. Chen,, M. Hearst, etc., Cha-Cha: A System for Organizing Intranet Search Results, Proc. of USENIX USITS, 1999

[6] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, VIPS: A vision based page segmentation algorithm, Microsoft Technical Report, MSR-TR-2003-79, 2003.

[7] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. A comparison of implicit and explicit links for web page classification. Proc. of WWW2006, pp. 643–650.

[8] Deng Cai, Xiaofei He, Ji-Rong Wen and Wei-Ying Ma, Block-Level Link Analysis, Proc. of SIGIR2004, pp.440-447.

[9] Delicious: http://del.icio.us

[10] E. J. Glover, K. Tsioutsiouliklis, S. Lawrence, D. M. Pennock, and G. W. Flake. Using web structure for classifying and describing web pages. Proc. of WWW2002, pp562-569.

[11] Ed. H. Chi et al., Using Information Scent to Model User Information Needs aand Actions on the Web, Proc. of SIGCHI, 2001

[12] Eric J. Glover etc., Improving Category Specific Web Search by Learning Query Modifications, Symposium on Applications and the Internet, 2001, pp. 23-32.

[13] G. Jeh and J. Widom. Scaling personalized web search. Proc. of WWW2003, pp. 271-279.

[14] G. Xue, H. Zeng, Z. Chen, W. Ma, etc., Implicit Link Analysis to Small Web Search, Proc. of SIGIR2003, pp.56-63.

[15] P. Hagen, H. Manning and Y. Paul, Must search stink? The Forrester report, Forrester, June 2000.

[16] D. Hawking, E. Voorhees, P. Bailey and N. Craswell, Overview of TREC-8 web track. Proceeding of TREC-8, 1999, pp. 131-150.

[17] J. Kleinberg, Authoritative sources in a linked environment, Journal of the ACM, 46(5): 604-622, 1999.

[18] J. Q. Li, Y. Zhao, PathRank: Web page retrieval with navigation path, Proc. ECIR2009, pp. 350-361.

[19] J. L. Chen, B. Y. Zhou, J. Shi, H. J. Zhang, and Q. F. Wu. Function-based object model towards Website Adaptation, In Proc. of WWW01.

[20] K. Sepandar, H. Taher, M. Christopher, G. Gene. Exploiting the Block Structure of the Web for Computing PageRank, Stanford University Technical Report, 2003.

[21] K. Matsuda, T. Fukushima, Task-Oriented World Wide Web Retrieval by Document Type Classification, Proc. of CIKM1999. pp.109-113.

[22] L. Page, S. Brin, R. Motwani, and T. Winograd, The PageRank citation ranking: Bringing order to the web, Technical Report, Stanford University, 1998.

[23] Lin, S.-H. and Ho, J.-M., Discovering Informative Content Blocks from Web Documents, Proc. of SIGKDD2002.

[24] M. Najork and J. Wiener, Breadth-First Search Crawling Yields High-Quality Pages, Proc. of WWW2000, pp. 114-118.

[25] M. F. Tsai, T. Y. Liu, T. Qin, H. H. Chen, W. Y. Ma, FRank: a ranking method with fidelity loss. SIGIR2007: 383-390

[26] Monika Henzinger, Link analysis on the world wide web, Proc. of ACM Hypertext 2005, pp.1-3.

[27] N. Eiron and K. S. McCurley. Analysis of anchor text for web search. Proc. of SIGIR 2003, pp. 459–460.

[28] Pandit, S.; Olston, C. Source, Navigation-Aided Retrieval. Proc. of WWW2007, pp. 391-400.

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

[30] R. Fagin, R. Kumar, K. S. McCurley, J. Novak, D. Sivakumar, J. A. Tomlin, and D. P. Williamson. Searching the workplace web. Proc. of WWW2003, pp. 366-375.

[31] R. Kraft and J. Zien. Mining anchor text for query refinement. Proc. of WWW2004, pp. 666-674.

[32] S. Bao, X. Wu, B. Fei, G. Xue, Zhong Su, Y. Yu: Optimizing Web Search Using Social Annotation, Proc. of WWW2007

[33] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, M. Lau. Okapi at TREC. In:Text REtrieval Conference, 1992.

[34] T. H. Haveliwala. Topic-sensitive PageRank. Proc. of WWW2002, pp.517-526.

[35] Vaughan, L., & Thelwall, M., Scholarly use of the Web: what are the key inducers of links to journal web sites? Journal of the American Society for Information Science and Technology, 54 (1): 29-38, 2003.

[36] Y. Hu, G. Xin, R. Song, G. etc., Title Extraction from Bodies of HTML Documents and Its Application to Web Page Retrieval. Proceeding of SIGIR 2005, pp. 250-257.

[37] Y. Mizuuchi and K. Tajima. Finding context paths for Web pages. Proc. of ACM Hypertext, 1999, pp. 13–22.

[38] Z. Chen, S. Liu, W. Liu, G. Pu and W.Y. Ma. Building a. Web Thesaurus from Web Link Structure. Proc. of SIGIR 2003.

[39] Z. Nie, Y. Zhang, J. Wen, W.-Y. Ma: Object-level ranking: bringing order to Web objects. Proc. of WWW2005, pp. 567-574.

[40] T. Upstill, N. Craswell and D. Hawking, Query-independent evidence in home page finding, ACM Trans. Inf. Syst. 21 (3) (2003), pp. 286–313.

[41] Nick Craswell, Stephen E. Robertson, Hugo Zaragoza, Michael J. Taylor: Relevance weighting for query independent evidence. SIGIR 2005: 416-423

[42] Tsai, M.-F., Liu, T.-Y., Qin, T., Chen, H.-H., & Ma, W.-Y. (2007). Frank: A ranking method with fidelity loss. Proceedings of SIGIR 2007.

[43] Hawking, D., Overview of the TREC-9 Web Track, in TREC 2000.

[44] Hawking, D., Craswell, N., Overview of the TREC 2001 Web Track, in TREC 2001