Defining Logical Domains in a Web Site

Wen-Syan Li, Okan Kolak[†]*, Quoc Vu*
C&C Research Laboratories, NEC USA, Inc.
110 Rio Robles, M/S SJ100, San Jose, CA 95134, USA
email: {wen, okan, qvu}@ccrl.sj.nec.com

Hajime Takano
Human Media Research Laboratories, NEC Corporation
4-1-1, Miyazaki, Miyamae-ku, Kawasaki, Kanagawa 216, Japan
email: gen@hml.cl.nec.co.jp

Abstract

Each URL identifies a unique Web page; thus, it is viewed as a natural choice to use for organizing Web query results. Web search results may be grouped by domain and presented to users as clusters for ease of visualization. However, it has a drawback: dealing with large Web sites, such as `Geocities`, `W3C`, and `www.cs.umd.edu`. Large Web sites tend to yield many matches that leads to a few large, flat structured, and unorganized clusters. As a matter of fact, these sites contain Web sites of other entities, such as projects and people. Many pages in these sites are actually "logical domains" by themselves. For example, Web sites for projects at a university or the XML section at W3C could be viewed as "logical domains". In this paper, we propose the concept of *logical domain* with respect to *physical domain* which is identified simply by domain name. We have developed and implemented a set of rules based on link structure, path information, document metadata, and citation to identify logical domain entry pages and their corresponding boundaries. Experiments on real Web data have been conducted to validate the usefulness of this technique.

Keywords: Logical domain, domain boundary, WWW, link structures, site map

Hypertext 2000, San Antonio, TX.

1. Introduction

With the explosive growth of the WWW, most queries yield a large number of matched Web pages. Without aggregate structure and organization, it is difficult for users to forage for relevant pages. A URL identifies a unique page, ignoring dynamically generated pages. Thus, it is usually viewed as a natural choice to use for organizing Web query results.

For some Web search engines, query results are grouped by URL domain name and the users are presented with a set of clusters. This has an advantage of better visualization. The users first locate the most relevant site and browse through matched pages in that site. However, organizing query results by domain has two limitations, especially when dealing with large Web sites.

First, large sites tend to yield a lot of matches due to vast number of documents that they contain. This leads to a few unorganized clusters, which makes it hard for the users to distinguish relevant documents from others. As a matter of fact, many large Web sites, such as Geocities[1], AOL[2], and NEC BIGLOBE[3], are either ISP sites or Web site hosting providers. Many pages in these sites are actually "logical domains" by themselves. *A logical domain is a group of pages that has a specific semantic relation and a syntactic structure that relates them.* For example, Web sites of "a user home page", "a project group", and "a tutorial on XML" can be viewed as logical domains.

[†] This work was performed when the author visited NEC, CCRL. The author is currently a Ph.D. student at Department of Computer Science, University of Maryland.

Second, for a query with the keyword XML, many portal sites specialized in XML, such as `www.xml.org` and `www.w3c.org`, tend to have a large number of matches. Grouping results by domain does not provide a well organized way for users to locate the most relevant page in these Web sites. This motivates us to organize the query results by logical domain. For example, we may identify the logical domains `/standard/`, `/proposal/`, and `/official/` in `www.xml.org`. It may be a better visualization to show users the entry pages of these logical domains, rather than showing hundreds of pages within that domain.

In this paper we propose the concept of *logical domain*. Logical domains are organized based on functionalities, such as project, seminar, and personal home page, with respect to *physical domain*, which is organized based on domain names. We develop a technique for defining logical domains by utilizing Web page metadata including titles, URL, anchors, and link structure as well as citation. The technique starts with identifying logical domain entry page candidates followed by defining boundary of each logical domain.

The rest of paper is organized as follows. In Section 2, we present the rules used to identify logical domain entry pages. In Section 3, we describe the methods for defining boundaries for logical domains. In Section 4, we present our experimental results on real Web data. In Section 5, we summarize related work in this area. Finally we give our concluding remarks.

2. Identifying Logical Domain Entry Pages

In this section, we describe a set of rules that we consider and use in identifying logical domain entry page candidates. We start with the definitions of physical domains and logical domains.

2.1 Definition and Criteria

A physical domain is defined as a set of pages with the same DNS domain name. For example, `www.ccrl.com` and `www.ccrl.com/dl99ws/` are hosted by a Web server (or Web servers) of a unique DNS domain name and thus they are in the same physical domain. On the other hand, a *logical domain* is defined as a set of Web pages in a physical domain which as a whole provides a particular function or is self-contained as an atomic information unit. The root page of a logical or physical domain is called *entry page*, which is meant to be the first page to be visited by the users navigating that domain. We identify and summarize some functions of logical domains as follows.

Entry page for navigation : a page with the name index.html is the default entry page of a directory for most Web servers (e.g. `www.ccrl.com/index.html` is the entry point for `www.ccrl.com`). It may have a site map or a number of links to assist users in navigating the site.

Personal site : Usually personal web sites are located in a physical domain rather than being physical domains by themselves. `www.ccrl.com/~wen/`, for example, is the entry page for a personal Web site. The personal Web sites by themselves are independent. We view them as logical domains and treat them as individual entities.

Topic site : Usually web pages related to a particular topic are grouped together under a directory. Such logical domains could be used for class information, seminar announcement, faculty directory, or project Web sites. For example, `www-db.stanford.edu/people/` and `www.cs.umd.edu/projects/amanda/` can be viewed as logical domains by themselves.

Popular site : Sometimes a page in a domain may be more popular than the entry page of the domain. Such a popular page, indicated by a large number of external incoming links (i.e. citation), may be viewed as an entry page of a logical domain. Some example pages of this kind include (1) publication pages of well-known researchers or professors; (2) "hobby" pages, such as `www.cs.umd.edu/~sibel/poetry/poetry.html`; and (3) tutorial, reference, or direction pages, such as `www.cs.umd.edu/~pugh/intro-www-tutorial`.

2.2 Rules for Identifying Logical Domain Entry Pages

We have developed a set of rules for identifying logical domain entry pages based on the available Web page metadata, such as title, URL string, anchor text, and link structures as well as popularity by citation. Each rule has an associated scoring function. We define an initial scoring function for each rule and then make adjustments based on the experimental results on $3,040$ URLs in `www-db.stanford.edu` and $18,872$ URLs in `www.cs.umd.edu`. Every page is assigned with a score. The higher is the score of a page, the more likely that a page is a logical domain entry page. After all

```
Rule#1 url                              : "/~[^/]*/?$"                    : +60
Rule#2 url  : "^[^~]*/(people|users?|class(es)?|projects?|seminars?)/$": +30
Rule#3 url                              : "^[^~]*/$"                      : +20
Rule#4 url                              : "/cgi\-bin/"                    : -100
Rule#5 title                            : "\bhome\b"                      : +10
       title                            : "\bweb\b.*\bpage\b"             : +10
       title                            : "\bwelcome\b"                   : +5
Rule#6 incoming link anchor text        : "^home$"                        : +5
       incoming link anchor text        : "\bgo\b.*\bhome\b"              : +5
       incoming link anchor text        : "\breturn\b.*\bhome\b"          : +5
Rule#7 outgoing link anchor text        : "^home$"                        : -10
       outgoing link anchor text        : "\bgo\b.*\bhome\b"              : -10
       outgoing link anchor text        : "\breturn\b.*\bhome\b"          : -10
Rule#8 title of the linked page         : "\bhome\b"                      : -10
       title of the linked page         : "\bweb\b.*\bpage\b"             : -10
       title of the linked page         : "\bwelcome\b"                   : -5
Rule#9 external incoming link count     : >0                              : +20
       external incoming link count     :                                 : +20%
Rule#10 outgoing link count             : >20                             : +5
Rule#11 internal incoming link count    : == 0                            : +20
```

Figure 1: Rules and Scoring Functions for Identifying Logical Domain Entry Pages

pages are scored, a portion of top scored pages are used as logical domain entry page candidates for boundary definition. The rules in regular expression and their scoring functions are summarized in Figure 1. Now we present the rules in detail.

Rule 1: If a URL ends with a user home directory in the form of `~user name/`, the score is increased by 60. It is because such a URL is most probably an entry page of a user home page and a personal Web site is viewed as a logical domain. Note that `~user name/` and `~user name/index.html` are the same. Before applying the rules for identifying logical domains, we remove `index.html` from all URLs. Note that `www.cs.umd.edu/users/candan/` does not match with this rule.

Rule 2: If path contains certain words given in the *topic word list*, such as `people` and `seminar`, and it is not under a user home page, then it is probably a logical domain. In our current implementation, a topic word list for the `.edu` domain contains `people`, `users`, `faculty`, `students`, `class`, `seminar`, and `project`. Other considered topic words include `FAQ` and `Information` for general purpose Web sites, such as `NEC` and `W3C`. If we identify a URL ending with a word in the topic word list, we increase the score. For example, `www-db.stanford.edu/people/` and `www.cs.umd.edu/users/` match with this rule while `www.cs.umd.edu/projects/omega/` does not.

Rule 3: If a URL ends with a "`/`", such as the URL `www.ccrl.com/dl99ws/`, there exists an index page (i.e. index.html) in that directory. Thus, this URL is designed to be an entry page for navigating that directory; we increase the score of the page. URLs such as `www.cs.umd.edu/projects/omega/` and `www-db.stanford.edu/lore/` match with this rule. Note that `/~crespo/publications/meteor/` does not match with this rule. The reason is that we would like to identify `/~crespo/` as a entry page instead of having both URLs as entry pages; which may consequently result in several smaller logical domains. However, we do not eliminate the fact that there can be more than one logical domain within a single user Web site. For example, in `www.cs.umd.edu` we identify `/users/sibel/poetry/` as a possible logical domain entry page in addition to `/users/sibel/` because that Turkish poetry portal site is very popular; indicated by a large number of external incoming links.

Rule 4: We do not consider dynamically created pages, such as `www.cs.umd.edu/cgi-bin/finger/`, as an entry page. Thus, we reduce the scores.

Rule 5: If the title of a page contains the phrase "home", "welcome", or "homepage" which indicate that page is a logical domain entry page, we increase its score. One frequently seen title matching this rule is "Welcome to my homepage".

Rule 6: If there is a link pointing to a page with the phrase "home", "go home", or "return home" in the anchor, there is a high possibility that the page being pointed to is a logical domain entry page.

Rule 7: This is the counterpart of *rule 6*. If a page A under B points to the page B with "home", "go home", "return home" in the anchor, then it is more likely that B is an entry page (based on *rule 6*). On the other hand, A which is under an entry page B is less likely to be an entry page too. Based on this observation, we reduce the score of A.

Rule 8:[1] This is the counterpart of rule 5. If a page A under B links to B whose title contains `home`, `Web page`, `welcome`, etc., then B is likely to be an entry page based on (*rule 5*). On the other hand, A is less likely to be an entry page too. Based on this observation, we reduce the score of A.

Rule 9: If a page has external incoming links from other physicial domains, then this page is likely to be an entry page. The reason is that people tend to link the entry page of a domain rather than pointing to a specific page. We increase the score of a page if it has an external incoming link. The higher the number of external incoming links, the higher the probabilty of the page being a logical domain entry page, so we add 20% of the number of external incoming links to the score of the page. The external incoming link information is extracted using AltaVista Connectivity Server[4, 5].

Rule 10: If the page has more than 20 outgoing links, it might be an entry page, pointing to several other pages within the logical domain. This is similar to the concept of "fan" proposed by R. Kumar et. al[6]. In [6], only those Web pages with more than 6 outgoing links are considered for topic distillation by assuming that in general a good page should not have less than 6 pages. Similarly, we observe that a page with very few outgoing links is probably not an entry page.

Rule 11: If there is no link from any other page in the same domain to this page, that means the page is designed to be accessed directly, and therefore probably an entry page to a logical domain.

The algorithm that scores the pages initially processes all the pages linearly, and assigns them a score. So any page, whether linked by some other page in the domain or not, are included in the scoring process. It is boundary detection that follows the links to find the pages in a candidate logical domain. Depending on the crawling method used, it is possible to crawl pages in a domain that are not linked by any other page in that domain. If they are crawled, scoring phase will process these pages too.

These rules and scoring functions perform well in the two Web sites we tested. Obviously, more sophisticated schemes and additional tuning can further improve the results. For example, we may develop techniques for identifying mirror sites and identical documents with symbolic links. If we can identify two URLs are actually identical, we should merge them for evaluation. We did perform pre-process tasks to merge URLs that can be identified identical. For example, the pages `~candan/` and `~candan/index.html` in `www.cs.umd.edu` are merged. Some more sophisticated techniques, such as [14], can be useful for further improvements of the experimental results.

Currently we are applying machine learning techniques to automatically derive scoring functions as listed in Figure 1 as well as evaluating our algorithms on large portal sites, such as NEC BigLobe[2], which is providing directory services.

3 Defining Boundary of a Logical Domain

After all pages are scored, a certain percentage or number of pages with higher scores are chosen as entry page candidates to form logical domains. The boundaries of logical domains are identified using path information and link structure. Our boundary definition tasks start with using path information to assign all pages under certain entry pages. Then, we make adjustments including checking accessibility through links, removal of small size logical domains, and reassignment of removed logical domain pages. In this section we present logical domain boundary definition methods.

3.1 Path-based Boundary Definition Approach

The main purpose of this approach is to assign *all* pages to a logical domain. The boundary of a logical domain is first defined as the pages under the entry page of that logical domain based on paths. Our intuition is that the pages in a logical domain would be in a directory, where the entry page is at the top level. After all the pages in all domains are determined, one more pass is

[1] This rule is not used in the experiments presented in Section 4.

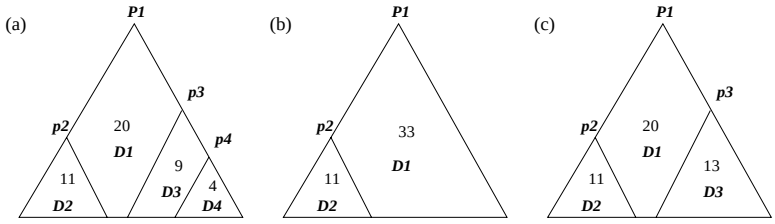

Figure 2: Path-based Approach (a) Initial Assignment; (b) Adjustment by Checking Min. Page Count and Removing Small Domains; and (c) Adjustment by Checking Min. Page Count with Dynamic Page Reassignment

performed to eliminate the ones with very few pages; which would not be suitable to be a logical domain by themselves. The pages in these removed logical domains are reassigned to other logical domains immediately above. Both passes performed on every logical domain in a bottom up fashion.

The detailed algorithm of this approach is described below. The algorithm takes the results from Section 2 (i.e. all n pages and their scores) and two parameters, the initial number of entry page candidates, k, and the minimum number of pages required in a domain, min_domain_size.

Step 1: Select k pages, $P_1 \ldots P_k$ with the highest score as entry page candidates.

Step 2: Build $Parent\_children\_List$ for $P_1 \ldots P_k$ based on the path. P_i is the parent of P_j if $P_i.hostdir$ is a longest prefix of $P_j.hostdir$. $P_i.hostdir$ = URL of P_i without the last file name.

Step 3: Assign $P_{k+1} \ldots P_n$ to be under one of the entry pages $P_1 \ldots P_k$ to form logical domains $D_1 \ldots D_k$. P_j is assigned to be under P_i if only if $P_i.hostdir$ is the longest prefix of $P_j.hostdir$. P_j is the entry page of the logical domain D_j.

Step 4: Merge D_j and P_j with D_i recursively from the bottom to the top if the size of D_j is less than min_domain_size, where P_i is the immediate parent of P_j

Step 5: Output all logical domain entry pages, P_i and their corresponding domains, D_i.

We first use *path information* for logical domain definition. That is, only pages that are under the same directory root can be in the same logical domain. A page must be under the same directory as the entry page or in a subdirectory under the entry page because this is how people organize HTML files in the directories. Say, if `www.cs.umd.edu/users/` is identified as a logical domain, `www.cs.umd.edu/projects/hcil/` cannot be in the logical domain `/users/` even there exists a link between these two pages. This is because `/projects/hcil/` is under a different directory.

Another design consideration is that we must perform the entry page definition task in a bottom up fashion in step 4. Let's use Figure 2(a) as an example for illustration. P_1, P_2, P_3, and P_4 are the entry pages for the logical domains D_1, D_2, D_3, and D_4 respectively. The numbers indicate the number of pages in each initial logical domain. We identify the $Parent\_children\_List$ as (P_1, P_2), (P_1, P_3), and (P_3, P_4). In step 4, we perform adjustments by removing those domains with very few pages. For example, we would like to consider only the domains with more than 10 pages. One way is to just remove all entry pages whose domains have less than 10 pages and reassign their pages to other domains. However, one drawback is that we will remove both D_3 and D_4 as shown in Figure 2(b). This scheme tends to generate domains as an hirarchy of many levels; that is, the domains closer to the root page will gather a lot of released pages from the bottom domains.

One better way (as in our implementation in step 4) is to reassign the pages dynamically in the bottom up fashion. As shown in Figure 2(c), we first remove D_4 and reassign the pages in D_4 and P_4 to its parent D_3. Now D_3 has 14 pages so that D_3 itself forms a logical domain and all domains have more than 10 pages.

3.2 Path+Link Boundary Definition Approach

One of the characteristics of the path-based approach is that all pages are assigned one logical domain. However, we observe that some logical domains may have isolated sub-domains. That is, we may not be able to navigate from a logical domain entry page to all the pages

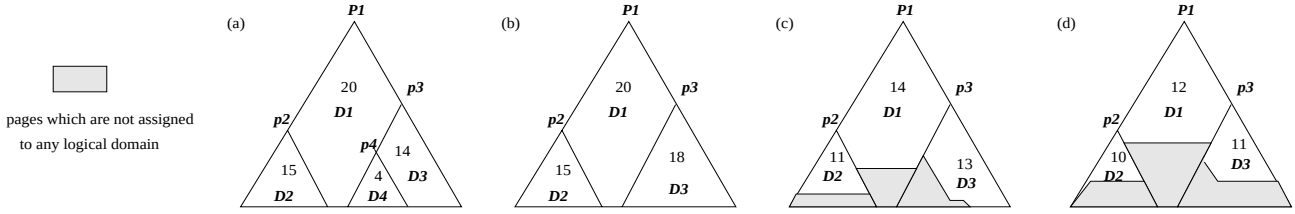

Figure 4: comparison of Approaches to Boundary Definition: (a) Original Assignment; (b) Path-based Approach; (c) Path+Link Approach with $radius$=3; and (4) Path+Link Approach with $radius$=2

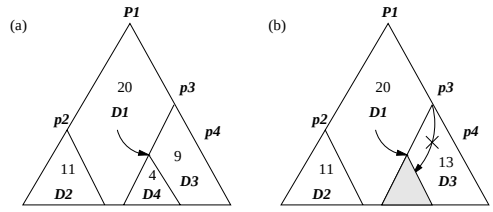

Figure 3: The Issue of Logical Domains with Isolated Sub-domains

in the same logical domain. In Figure 3(a), four logical domains are created. The pages in D_4 are crawled by following the links from a page in D_1 and there is no link from any page in D_3 to D_4. Using the path-based approach, the pages in D_4 would be released and reassigned to D_3. But users can not navigate from P_3 to all the pages in its logical domain. The path+link approach is developed for dealing with this drawback. Note that if P_3 has a link pointing to a page in D_1 and that page has a link pointing to a page in D_4, such a path is not considered because otherwise D_3 is not a tree by itself.

The path+link based approach is similar to the path-based approach except step 3 and step 4 need to be modified to consider page accessibility from logical domain entry pages. A new parameter $radius$ is introduced for specifying how many links to follow for verifying accessibility. The detailed algorithm is as follows (the additions to the path-based approach algorithm are underlined).

Step 1: Select k pages, $P_1 \ldots P_k$ with the highest score as entry page candidates.

Step 2: Build $Parent\_children\_List$ for $P_1 \ldots P_k$ based on the path. P_i is the parent of P_j if $P_i.hostdir$ is the longest prefix of $P_j.hostdir$. $P_i.hostdir$ = URL of P_i without the last file name.

Step 3: Assign $P_{k+1} \ldots P_n$ to be under one of the entry pages $P_1 \ldots P_k$ to form logical domains $D_1 \ldots D_k$. P_j is assigned to be under P_i if only if $P_i.hostdir$ is the longest prefix of $P_j.hostdir$ and P_j can be reached from P_i by following $radius$ hyperlinks within the union of D_i, P_j, and D_j. P_j is the entry page of the logical domain D_j.

Step 4: Merge the pages in D_j and P_j which can be reached from P_i by following $radius$ hyperlinks within D_i. with D_i recursively from the bottom to the top if the size of D_j is less than the value of min_domain_size, where P_i is the immediate parent of P_j

Step 5: Output all logical domain entry pages ,P_i, and their corresponding domains, D_i.

In Figure 4, we compare these two discussed approaches. Figure 4(a) shows an initial result based on only path information, in which some small domains may exist. We can add a constraint of minimum domain size and dynamically reassign pages. As shown in Figure 4(b), the size of each domain is now more desirable. Figures 4(c) and 4(d) illustrate the result by checking accessibility by link with different radius values. Note that in this two cases some pages may not be assigned to any domain. The larger is the radius, the more pages are contained in a logical domain. In the next section, we present the experimental results.

4. Experimental Results

We have implemented the rules and algorithms described in Sections 2 and 3. We have conducted experiments on `www-db.stanford.edu`, `www.cs.umd.edu`, and `www.w3c.org`. The pages collected from these domains were crawled from the root pages by following the links within the same domains. 3040, 18872, and 13356 pages were collected from these three sites respectively. The purposes of the experiments are observing (1) the performance of the rules presented in

```
Score           Document
===== ================================
488.8 www-db.stanford.edu/
150   ~crespo/publications/awareness/
124.2 ~ullman/
115.4 people/
108.2 ~cho/
105.6 ~vin/
105.4 ~widom/photos.html
105.2 ~vassalos/cs345_98/
100.8 ~wiener/
100.8 ~breunig/
100.4 ~crespo/publications/webwriter/
100.4 ~crespo/publications/meteor/
100.2 ~catherin/
100.2 ~ullman/pub/kdt.html
100.2 ~testbed/python/manual.1.3/lib/
97    ~ullman/fcdb.html
94    ~widom/
91.8  ~junyang/seven/
90    ~sergey/
90    ~ullman/ullman-books.html
89.2  ~echang/
88.6  ~zhuge/
87.2  ~zhuge/zhuge.html
87    ~ullman/ullman-papers.html
86.6  ~widom/widom.html
86.4  ~vassalos/
86.2  ~ullman/fcsc-notes.html
86.2  ~danliu/
85.8  ~tlahiri/
85.8  ~chaw/
=======================================
```

Figure 5: Logical Domain Entry Page Identification Results for `www-db.stanford.edu` (top 30)

Section 2; (2) the behavior of algorithms and parameters; and (3) the performance of logical domain definition algorithms. We now present experimental results in detail.

4.1. Performance of Entry Page Identification

The first experiment is to test the performance of rules. We believe that the results are satisfactory, given that we do not perform intensive tuning to make them "excellent". The effectiveness of the rules, especially rule 2, are really application domain dependent. The scores of the top 50 pages in `www-db.stanford.edu` are given in Figure 5.

In the experimental results on `www.cs.umd.edu` and `www-db.stanford.edu`, the logical domains identified are mostly Web sites for people, projects, and classes. The results indicate that most pages in these two Web sites are organized in such ways for users to navigate.

On the other hand, `www.w3.org` behaves more like a single entity. We observe that the logical domains in `www.w3.org` are defined based on "subjects" rather than "entities". Some representative logical domains identified are as follows:

```
www.w3.org/MarkUp/    www.w3.org/Protocols/
www.w3.org/XML/       www.w3.org/People/
www.w3.org/TR/        www.w3.org/Provider/
www.w3.org/Tools/     www.w3.org/RDF/
```

Our technique aims at considering both link structure and contents. Although we do not examine the document contents directly, our rules examine the number of external incoming links and use that information as an indicator to judge the importance of each page. This concept is similar to "topic distillation" for organizing Web query results proposed by J. Kleinberg. [7]. In the experiment results, many popular pages on particular topics outscore most of personal and project Web sites. Some of logical domains identified mainly by their popularity. Examples could be a manual site, an interesting portal site, or publications.

4.2. Behavior of the Algorithms

The second set of experiments are designed to observe the behavior of the algorithms by varying three parameters: (1) initial number of entry pages; (2) minimum domain size; and (3) radius for checking accessibility by link. Table 1 summarizes the experimental results on `www-db.stanford.edu` with different combinations of parameter values. Note that in the experiments, we do not consider the entry page candidates without any page under them. In experiment 1, we found 73 entry page candidates have no page under them. We reassign these pages to other logical domains. Thus, 27 logical domains are defined.

We observe that the selection of initial number of entry pages makes little difference as long as it is large enough to include all logical domains. One reason is that there are many small domains in the Stanford Web site and they are usually later removed. Another reason is that the logical domains have high scores and contain a large number of pages tend to be ranked at the top. Thus, they are selected as initial entry page candidates regardless whether 50 or 100 is used for the initial number of entry page candidates.

Experiment #	Initial Number of Entry Pages	Min. Domain Size	Link Radius	Logical Domains Identified	Avg Logical Domain Size	Total Number of Pages Included in Logical Domains
1	100	NA	NA	27	112.5	3040
2	100	5	NA	17	178.8	3040
3	100	10	NA	13	233.8	3040
4	100	NA	2	24	16.1	388
5	100	5	2	14	26.8	376
6	100	10	2	8	43.0	345
7	100	NA	3	24	24.2	582
8	100	5	3	14	38.2	574
9	100	10	3	8	68.1	545
10	50	NA	NA	24	126.6	3040
11	50	5	NA	16	189.93	3040
12	50	10	NA	13	233.8	3040
13	50	NA	2	20	16.3	326
14	50	5	2	12	26.2	315
15	50	10	2	7	40.4	284

Table 1: Comparisons of Results on `www-db.stanford.edu` Using Different Parameter Values

The second observation is that the minimum domain size does have a great impact on the total number of logical domains defined in all experimental setup. The larger is the minimum required domain size, the fewer are the logical domains defined.

The third observation is that the increase of the radius for checking accessibility by link results in the increase of average domain size. We notice that in two university sites tested in this paper, the link structure is quite shallow. For example, in `www-db.stanford.edu` 76% of the Web pages can be reached from the root page within 3 links. Thus, we believe that checking accessibility within a small number of links is sufficient.

4.3. Performance of Boundary Definition

In Section 4, we present two approaches to boundary definition. Note that the boundary definition is tunable in our algorithms. In Table 1, we show how the number of domains identified can be adjusted using the parameters of minimum domain size and radius for checking accessibility by link. Thus, the average size of each domain can be tuned as well. For the bound definition results, on the left hand side of Figure 6, we show the partial boundary definition for `www.w3c.org` in the format of Web site maps.

5. Related work

Most related work are in the area of Web query result organization. The motivation is that since WWW encourages hypertext and hypermedia document authoring (e.g. HTML or XML), authors might prefer to create Web documents that are composed of multiple pages connected with hyperlinks. Therefore, indexes used by search engine based on individual pages are not sufficient. Many schemes have been investigated to organize either query results for better presentation or to cluster pages for better indexing.

Keishi et al.[8] proposed a query framework in which hypertext is divided into connected sub-graphs corresponding to individual topics. These sub-graphs are used as the data units for queries. Both contents and link topology are used for finding connected sub-graphs of documents associated with a topic. This approach is also used as base for visualization, as in[9]. Li and Wu[10] introduced the concept of *information unit*, which can be viewed as a logical Web document consisting of multiple physical pages as one atomic retrieval unit. A framework of query relaxation by structure is proposed. In this framework, a set of connected physical pages that, as a whole, contains all query terms can be retrieved. This framework supports desirable progressive processing for Web queries, i.e. it generates the best k results in the order of ranking. This method aims at dynamically organizing query results. The work presented in this paper can be combined with the query relaxation by structure scheme by Li and Wu[10] in the way that the structure relaxation is limited to logical domains rather than physical domains as used in [10]. The work by Tajima et al.[11] extends the concept of information units by considering keyword occurrence

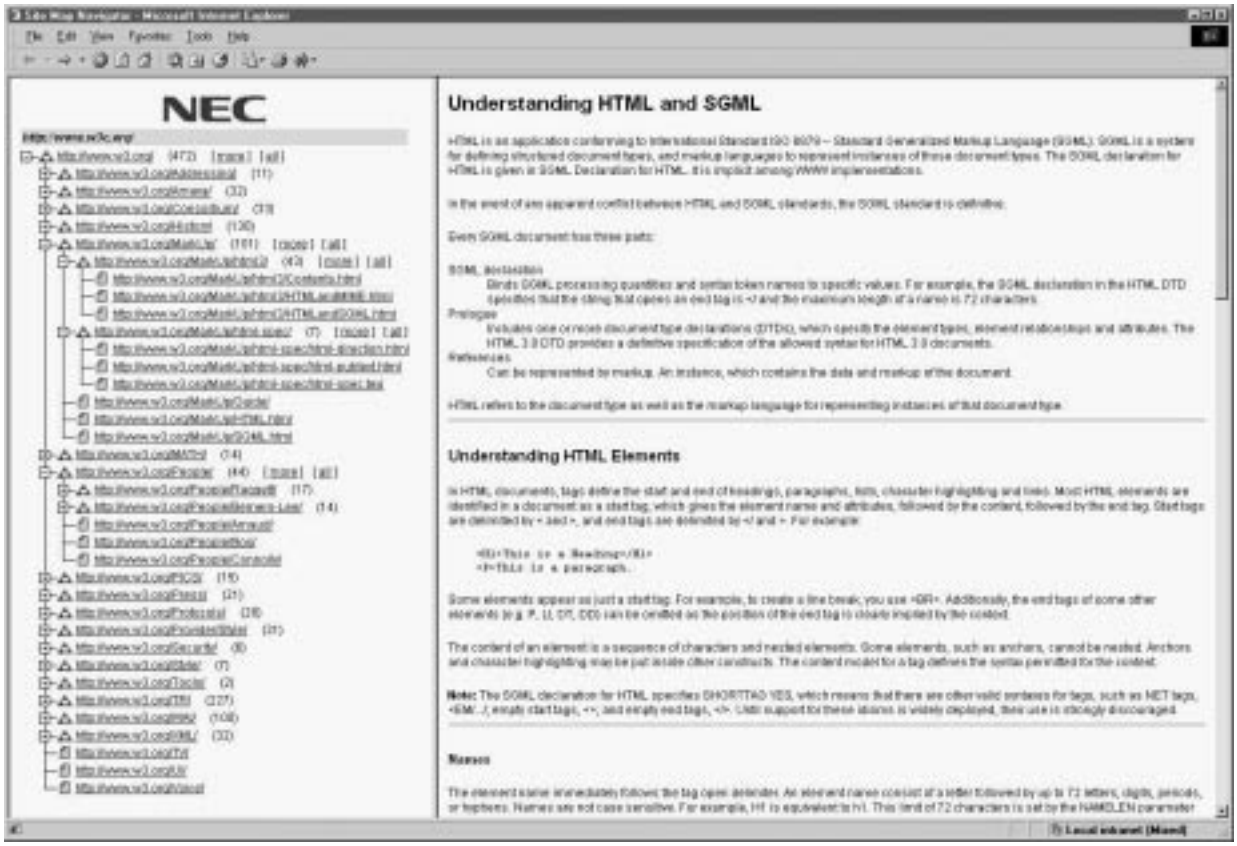

Figure 6: Web Site Map for Navigating `www.w3c.org`

frequency and distribution. This work focuses on for a given set of documents as query results, how to rank them based on contents and link structures.

WWW Dynamic Bookmark (WDB)[12] is a management tool for supporting revisiting WWW pages. WDB watches and archives a user's navigation behavior, analyses the archive, and shows analyzed results as clues for revisiting URLs. It integrates link analysis and user behavior analysis to evaluate WWW page importance. WDB presents a list of sites that a user has visited, in the order of importance, via a landmark list in each site, and showing relationships among sites. In this work, only a few simple rules are applied.

The contents of Web pages are often not self-contained given they are in hypertext. A page author often assumes all the readers of the page come through the same path (i.e. from root page). However, search engines only return the hit pages rather than giving all pages on the navigation path. Mizuuchi and Tajima[13] propose a method of these paths within a physical domain. Our methods for finding logical domains and their boundaries can be used to enhance the work by Mizuuchi and Tajima[13] in the way that we only need to return the pages in the path from the hit documents to the logical domain entry page, instead of the root page.

Compared with the mentioned existing work, the work presented in this paper is novel in defining logical domains in a Web site. Another novelty of our work is that in addition to page metadata and link structures, we also utilize external incoming links for ranking quality of logical domains so that more important and higher quality entry pages are selected as representatives.

6. Concluding remarks

In this paper, we present the concept of *logical domain* with respect to *physical domain* identified by domain name. We develop a set of rules for identifying logical domain entry pages and methods for defining logical domain boundaries. These methods have been experimentally evaluated on three large, real Web sites, including two university sites and one organization site. The preliminary experimental results show our approach is promising in producing good results.

The idea was to develop a general system that may identify the logical domain in any web site. In case

of W3, the site is well organized using directories for logical domains, so overlap is normal. Although it is true that for that site one could simply use the directory structure to organize the site into logical domains, one would first need to know the fact that directory structure is in parallel to logical domain structure for that site. So we can say the system was still useful in that it helped finding that fact.

Finally, we would like to note the fact that these algorithms are designed to work off-line. This is more obvious for creating a sitemap, but may be subtle when we talk about organizing query results. We can summarize the procedure as follows: (1) All the pages in the domain(s) that is indexed are crawled, (2) Logical domains are extracted off-line using crawled pages, (3) The results of off-line logical domain extraction can be used to do on-line query result organization and sitemap creation.

Acknowledgement

The authors would like to express their appreciations to the Web sites `www.w3c.org`, `www.cs.umd.edu`, and `www-db.stanford.edu` for their data used in the experiments described in this paper. Selecting the Web sites is based on the considerations (1) the authors need to be familiar with the contents so that the authors can evaluate the results; and (2) the pages in the Web sites must not be dynamically generated pages. The second consideration restricts the authors from using most of corporation sites. The experimental results presented in this paper are for the purposes of scientific research only.

References

[1] Yahoo Inc. *Information available at http://www.geocities.com/*.

[2] America Online, Inc. *Information available at http://www.aol.com/*.

[3] NEC Corporation. *Information available at http://www.biglobe.ne.jp/*.

[4] AltaVista Technology, Inc. of California. *Information available at http://www.altavista.com/*.

[5] Krishna Bharat, Andrei Broder, Monika Henzinger, Puneet Kumar, and Suresh VenKatasubramanian. The connectivity server: fast access to linkage information on the web. *Computer Networks and ISDN Systems*, (30):469–477, 1998.

[6] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the Web for Emerging Cyber-Communities. In *Proceedings of the 8th World-Wide Web Conference*, Toronto, Canada, May 1999.

[7] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, January 1998.

[8] Keishi Tajima, Yoshiaki Mizuuchi, Masatsugu Kitagawa, and Katsumi Tanaka. Cut as a Querying Unit for WWW, Netnews, e-mail. In *Proceedings of the 1998 ACM Hypertext Conference*, pages 235–244, Pittsburgh, PA, USA, June 1998.

[9] Kenji Hatano, Ryouichi Sano, Yiwei Duan, and Katsumi Tanaka. An Interactive Classification of Web Documents by Self-Organizing Maps and Search Engines. In *Proceedings of the Sixth International Conference on Database Systems for Advanced Applications (DASFAA),*, pages 35–42, Hsinchu, Taiwan, April 1999.

[10] Wen-Syan Li and Yi-Leh Wu. Query Relaxation By Structure for Document Retrieval on the Web. In *Proceedings of 1998 Advanced Database Symposium*, Shinjuku, Japan, December 1999.

[11] Keishi Tajima, Kenji Hatano, Takeshi Matsukura, Ryoichi Sano, and Katsumi Tanaka. Discovery and Retrieval of Logical Information Units in Web. In *Proceedings of the 1999 ACM Digital Libraries Workshop on Organizing Web Space*, Berkeley, CA, USA, August 1999.

[12] Hajime Takano and Terry Winograd. Dynamic Bookmarks for the WWW. In *Proceedings of the 1998 ACM Hypertext Conference*, pages 297–298, Pittsburgh, PA, USA, June 1998.

[13] Yoshiaki Mizuuchi and Keishi Tajima. Finding Context Paths in Web. In *Proceedings of the 1999 ACM Hypertext Conference*, Darmstadt, Germany, February 1999.

[14] Krishna Bharat and Andrei Z. Broder. Mirror, Mirror, on the Web: A Study of Host Pairs with Replicated Content. In *Proceedings of the 8th World-Wide Web Conference*, Toronto, Canada, May 1999.