Finding Context Paths for Web Pages

Yoshiaki Mizuuchi † Keishi Tajima

Department of Computer and Systems Engineering
Kobe University, Japan

(† Currently at NTT Data Corporation)

Background (1/3)

Aceess to Web Pages

There are three ways to access Web pages:

1. direct access by known URLs,
2. navigation from other pages, and
3. content-based access via search engines. $\Leftarrow$ currently, very important!

However, many page authors assume only 1 and 2.

Background (2/3)

Structure of Web Sites

Many page authors set up "**standard routes**" to each page.

Most Web sites are consisting of:

- entrance pages for direct access,
- other pages,
- hierarchical paths from entrance pages to other pages, and
- other links.

Most Web sites have hierarchical standard routes.

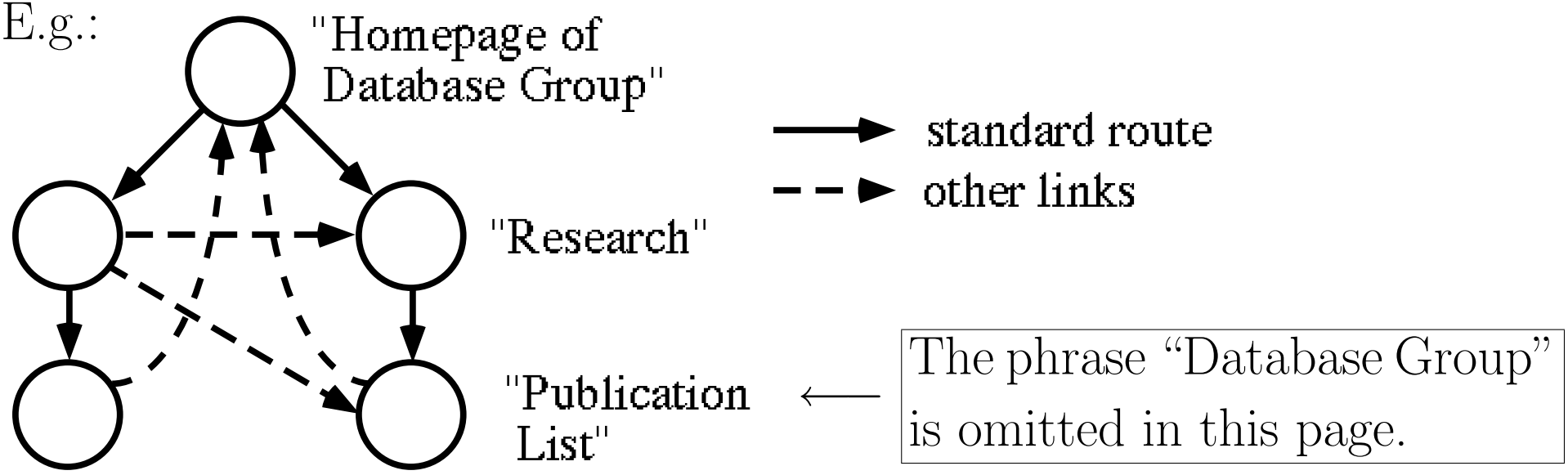
Background (3/3)

Omission of Already Known Information

A Page author sometimes:

- assumes every reader comes through the same path,
- assumes they already know information written on that path, and
- omit such information in the current page.

E.g.:



Problem

Sometimes Web pages are not self-contained.

$\Downarrow$

Indexes only with the words in each page itself are sometimes incomplete.

$\Downarrow$

Queries with multiple keywords sometimes fail to find appropriate pages.

E.g.:

Query {"Database Group", "Publication List"} cannot retreive the page in the last example.

Goal of This Research

In order to complement incomplete indexes:

- we detect paths assumed by page authors, and
- we extract appropriate keywords from the pages on the paths.

We call such a path **entrance path** or **context path**.

The main goal of this research is:

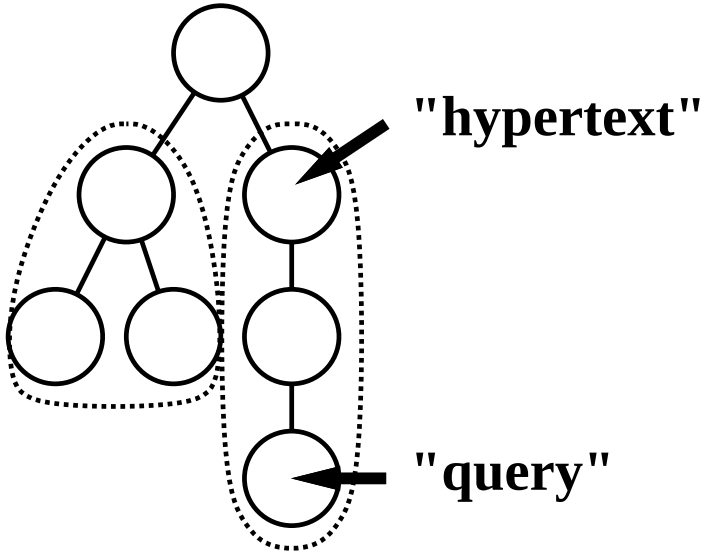
> to develop a method of automatically detecting an entrance path for each page.

Related Work (1/3)

Subgraph-based Retrieval of Web Pages

Our previous research [ACM Hypertext 98]

- We detect documents consisting of series of connected Web pages, and
- use those documents as a unit of query.
- We can retrieve documents including given multiple keywords even if they appear in different pages.



There are other kind of cases where query with multiple keywords fails.

Related Work (2/3)

Indexing with keywords in anchors [Li 98]

They use words in an anchor of a link for indexing its destination page.

We extract words from both direct and indirect ancestor pages while they consider only direct parent pages

Related Work (3/3)

Structual Queries

Many researches have proposed structural queries like:

select p
where appear("Database Group", p),
appear("Publication List", p'),
$p \rightarrow^* p'$

They may retrieve pairs of unrelated pages connected through very long links.

Our Basic Strategy

To detect entrance paths, we use following information:

- directory structure encoded in URL
- file names (currently, only "`index.*`")
- graph structure

We determine entrance paths by using heuristics based on those information.

Preliminaries (1/3)

Link Patterns

We classify links into 5 link patterns:

- **intradirectory links**:
  between pages in the same directory
- **downward links**:
  from pages in a directory to pages in its direct/indirect subdirectory
- **upward links**:
  from pages in a directory to pages in its ancestor directory
- **sibling links**:
  between pages in incomparable directories on the same site
- **intersite links**:
  between pages on different sites

Preliminaries (2/3)

Link Roles

Based on link patterns and other information, we also classify links into 3 link roles:

- **entrance links**:
  constituting the standard routes to their destination pages.
- **back links**:
  going back to pages on the entrance paths
- **jump links**:
  the others

Preliminaries (3/3)

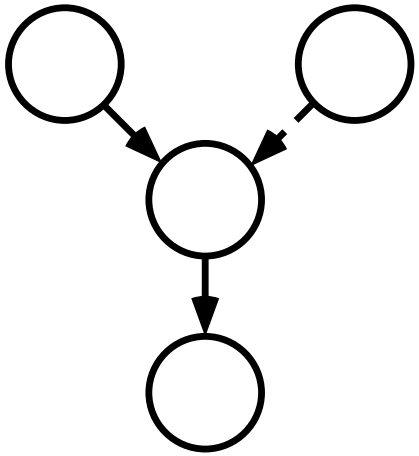
Basic Assumptions

We adopt the following assumptions for simplicity:

1. Every page has at most one entrance path.

   We cannot perfectly determine entrance paths. Instead, we select one most likely path for each page.

2. Entrance paths are concatenation of entrance links.



This will greatly reduce the computation cost. Discovery of entrance paths can be reduced to discovery of an entrance link.

Discovery of Entrance Links (1/6)

Overview

We determine the entrance link for each page in the following steps:

1. collect all incoming links,
2. exclude those that cannot be the entrance link,
3. select the most likely one, and
4. examine whether that is really the entrance link or not.

Discovery of Entrance Links (2/6)

Step 2: Excluded Cases

Those are never entrance links, and are excluded in Step 2:

- upward links, and
- intradirectory links to a page named "`index.*`".

Discovery of Entrance Linss (3/6)

Step 3: Priority for Link Selection

In step 3, first we use the following priority orders:

1. **downward** links have the highest priority,
   - among them, links from upper directory have higher priority
2. **intradirectory** links next,
   - among them, links from `index.*` have higher priority
3. **sibling** links next, and
4. **intersite** links have the lowest priority

Discovery of Entrance Links (4/6)

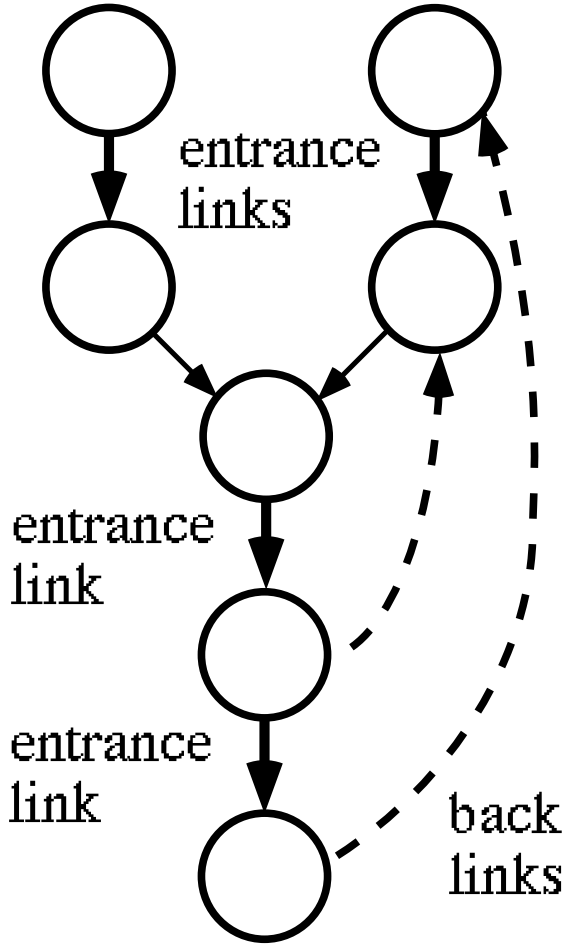
Step 3: Priority for Link Selection (cont'd)

For candidates with equal priority, we select one

> so that there would be the largest number of **back links**

over that link.

However, this produces mutual dependency among the decision of entrance links of pages.



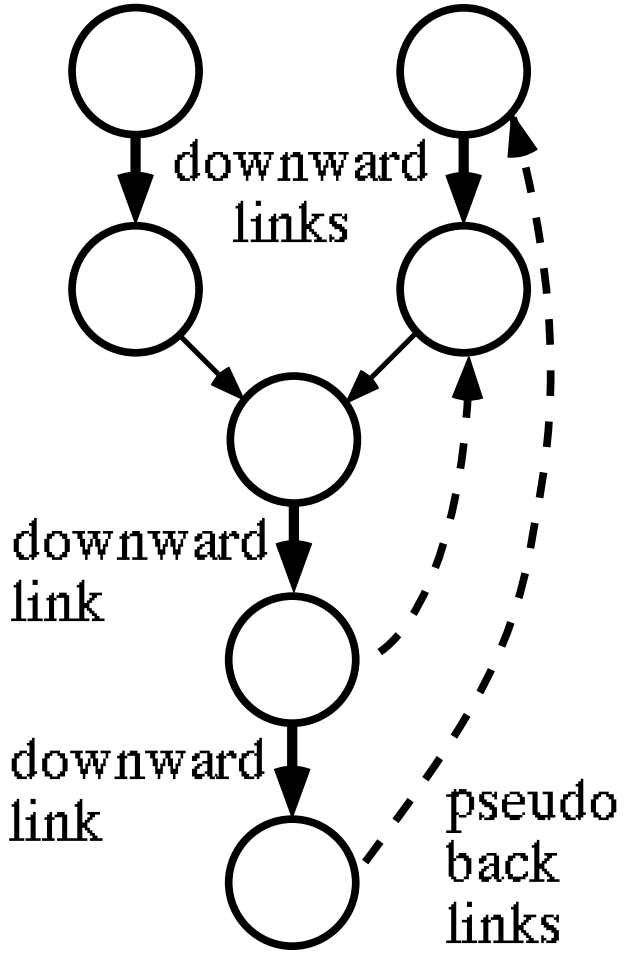
Discovery of Entrance Links (5/6)

Step 3: Priority for Link Selection (cont'd)

To avoid that mutual dependency, we use downward links instead of entrance links.

This method:

- can be computed easily because downward links can be determined independently, and
- can approximate the previous method



Discovery of Entrance Links (6/6)

Step 4: Examining the Selected Candidate

We examine whether the selected one is really the entrance link by the following criteria:

- if the selected one is not a intersile link, $\Longrightarrow$ YES
- if the selected one is an intersite link,
  - if the number of (pseudo) back links is greater than a threshold $\Longrightarrow$ YES
  - if it is less than a threshold $\Longrightarrow$ NO

  (But we need some better method.)

Experimental Results (1/2)

Precision Ratio

We applied our method to our research group Web site with:

- 840 pages,
- 1400 links within the site, and
- 2200 links to other Web sites,

and find correct entrance paths for 78% pages.

However, the precision ratio greatly changes depending on the style of page organization of each Web site.

Experimental Results (2/2)

Failing Cases

- directories with no "`index.*`" and no incoming downard links.
  (about 60 pages)
- 3 documents each of which is organized into a series of pages in the same directory and connected via "next" and "previous" links. We cannot distinguish "next" links and "previous" links.
  (about 120 pages)
- entrance pages, for which our method deterimne some links as their entrance links.
  (7 pages)

Conclusion

- We show the first step for discovery of entrance paths for Web pages.
- However, discovery of entrance paths heavily depends on the way of page organization, and there can be no unique general solution.
- Therefore, we need to
  - determine the organization style of each Web site,
  - determine the organization style of each group of Web pages, and
  - adopt a method appropriate to that style.

  Currently, we are studying this approach.