Discovery and Retrieval of Logical Information Units in Web

Keishi Tajima Kenji Hatano Takeshi Matsukura
Ryouichi Sano Katsumi Tanaka

Department of Computer and Systems Engineering
Kobe University

Background (1/3)

Access 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.**

   Each page is indexed based on the words appearing in it.

Background (2/3)

Two Types of Queries

1. retrieving an already-known page out of huge Web data.
   - We use keywords that we saw in that page.
   - The querying unit may be a **page**.
2. looking for unknown documents concerned with a topic of one's current interest.
   - We use keywords that are likely to appear in documents concerned with that topic.
   - The querying unit is a **document**

Background (3/3)

Pages v.s. Documents

- Pages — physical data units designed for presentation.
- Documents — logical data units consisting of independent self-contained set of information.

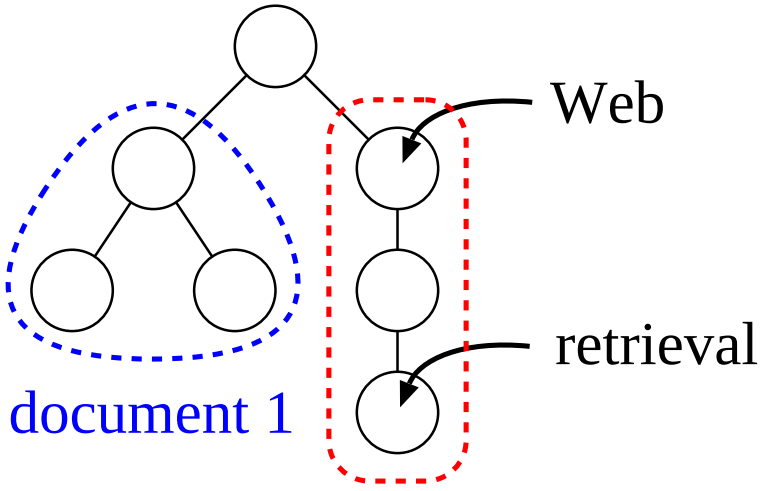
One complete document in Web is often composed of multiple pages.

Problem

Data Unit in Web Query

A Data unit in ordinary search engines is not a documents but a page. For this discrepancy,

conjunctive queries with multiple keywords may fail to retrieve an appropriate document.



query = {Web, retrieval}

Goal of This Research

The main goal of this research is:

to develop a framework for querying logical documents in Web data.

Related Work (1/4)

Classification (1): Static v.s. Dynamic

Static Approach:

> statically divides the Web graph into subgraphs corresponding to logical data units in advance.

- the most straightforward consequence of our observation of the problem.
- we can employ complex computation for data unit discovery without the response latency at query execution.

Related Work (2/4)

Classification (1): Static v.s. Dynamic

Dynamic Approach:

dynamically determines the data units when a query is given

- The units may vary depending on the given query.
- By combining with some ranking method, it can always return "the best n" candidates [Li98].

Related Work (3/4)

Classification (2): Information Used for Data Unit Detection

- page contents
  - text
  - tagging pattern
- graph structure
- directory structure in URL

Related Work (4/4)

- Static / Term Frequency [Mizuuchi97,Tajima98]

  repeatedly merging neighboring nodes based on the term frequency similarity

- Static / Link and Directory Structure [Nagafuji98]

  repeatedly merging and splitting directories based on the structures of links between directories.

- Dynamic / Graph Structure

  "**minimal subgraph approach**" [Li98,Hatano99]
  (the only dynamic approach published so far)

find minimal subgraphs including all the keywords

Our Approach (1/3)

We take **minimal subgraph approach**

Though the static approach is a straightforward solution,

- it is difficult to perfectly detect logical data units, and
- always returning "best n candidates" is very useful.

Our Approach (2/3)

Issues in the Minimal Subgraph Approach

1. The cost to dynamically compute minimal subgraphs is high.

   We detect links that are clearly not parts of a logical data unit, and does not traverse them.

2. Not all subgraphs are part of one logical document. Some ranking method is essential to give priority to likely ones.

   We designed a ranking methods based on how strongly the keywords appearing in a subgraph seem related to each other.

Our Approach (3/3)

Summary of Our Approach

Given query keywords, we approximate logical data units including all the given keywords by the following steps:

1. we determine links that may be parts of a logical document,
2. we find minimal subgraphs connected only by those links and including all the given keywords, and
3. we rank them by our formula.

Step 1: Determining Links to Traverse (1/3)

Typical Structure of Documents in Web

- sequence — pages corresponding to sections of a document linked via "`next`" and "`previous`" links, e.g.:
  - html versions of papers or slides.
- hierarchy — pages with hierarchical indexing structure, e.g.:
  - many homepages of people or organizations, or
  - database pages, such as product catalog pages.

Step 1: Determining Links to Traverse (2/3)

Three Kinds of Links

1. jump links — to pages in other documents by other authors,
2. route links — intended by page authors to be "standard routes" through which the readers go through the document, and
3. back links — going back to some prior pages along the routes.

a document $\equiv$ a subgraph connected by "**route links**"

Step 1: Determining Links to Traverse (3/3)

Excluding Non-Route Links

We exclude links that point to:

1. pages in ancestor directories,
2. "`index.*`" or the directory name of the same directory
3. pages in an incomparable, not sibling directory, or
4. pages on a different Web site

We use very simple analysis because:

- we only want to exclude clearly wrong ones, and
- our purpose is to reduce the cost

Step 2: Finding Minimal Subgraphs

Our current implementation uses the result of an existing search engine.

1. first we fetch pages including at least one keyword by submitting a disjunctive query to the search engine,
2. we fetch the pages in a distance 1 via "route links" from retrieved pages
3. repeat the step 2 until enough number of answers are found.

For a real system with its own database, we need to elaborate some algorithm to traverse a graph. See [Li98].

Step 3: Ranking Method

Cost Function $F(G)$

$F(G)$ calculates the "cost" of a subgraph G:

$$F(G) = \sum_{v \in V} (K(v) + C)^{-1}$$

where V: a set of pages in G
$K(v)$: the number of keywords in the page v
C: a constant parameter to adjust F

A "cost" of a page v is smaller when $K(v)$ is larger.

We rank G with smaller $F(G)$ higher.

Step 3: Ranking Method

Properties of $F(G)$

$$F(G) = n \cdot \frac{1}{n} \sum_{v \in V} (K(v) + C)^{-1}$$

where n is the number of pages in V.

1. $F(G)$ is proportional to n, i.e. the size of the subgraph.
2. roughly, $F(G)$ is inversely proportional to the average of $K(v)$

Step 3: Ranking Method

Properties of $F(G)$ (cont'd)

3. the larger the variation of $K(v)$ is, the larger $F(G)$ is.

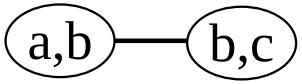
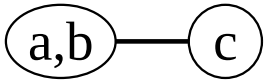
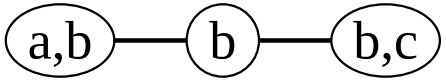
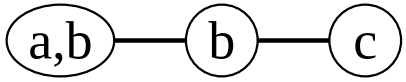
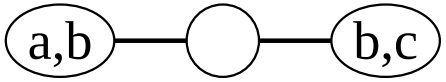
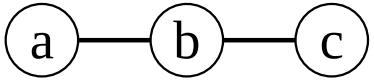
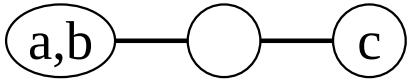
$$\frac{1}{n}\sum_{v \in V}(K(v)+C)^{-1} \geq \{\frac{1}{n}\sum_{v \in V}(K(v)+C)\}^{-1}$$

   "$=$" holds only when $\forall v_i, v_j . K(v_i) = K(v_j)$.

4. The smaller C is, the more significant the effect of $K(v)$ is. When $C = 0$, the cost of a page with no keyword is infinite.

5. $F(G)$ does not reflect the shapes of G, which heavily depends on the authoring style of authors.

Step 3: Ranking Method

G	n	avg($K(v)$)	$C=3$	$C=1$	$C=0$
	1	3	0.167	0.250	0.333
	2	2	0.400	0.677	1.000
	2	1.5	0.450	0.833	1.500
	3	1.67	0.650	1.167	2.000
	3	1.33	0.700	1.333	2.500
	3	1.33	0.733	1.677	∞
	3	1	0.750	1.500	3.000
	3	1	0.783	1.833	∞

Step 3: Ranking Method

Summary of Properties of $F(G)$

$F(G)$ reflects the distribution of locality of keywords.

- When keywords are distributed to many pages, n becomes large.
- The more the localities of keywords overlap, the larger the average number of $K(v)$ is.
- When keywords are continuously overlapping in every page, they are more likely to be one document related to those keywords.

Step 3: Ranking Method

Distribution within Pages

- $F(G)$ reflects the distribution of keywords across pages.
- We should also examine the distribution within each page.

e.g.

Links to My Favorite Pages

- 'xxx' Home Page — ... k_1 ...
- Pages on 'yyy' — ...

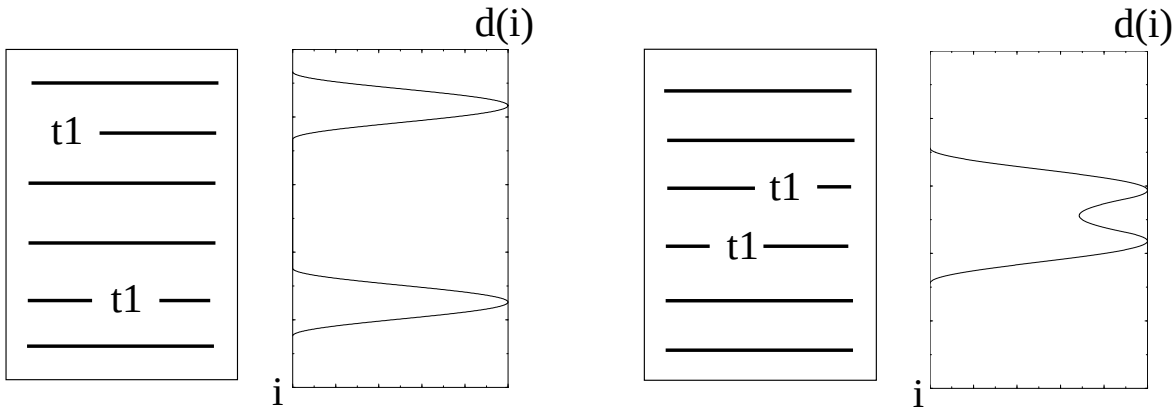
  ⋮

- Pages by 'yyy' — ... k_2 ...

Step 3: Ranking Method

Appearance Density of Words

- Each word occurrence has its influence around it.
- $d_t(i)$: appearance density of a word t at a position i is the sum of the influences of the occurrence of t around i.
- We normalize $d_t(i)$ so that $\max_i(d_t(i)) = 1$

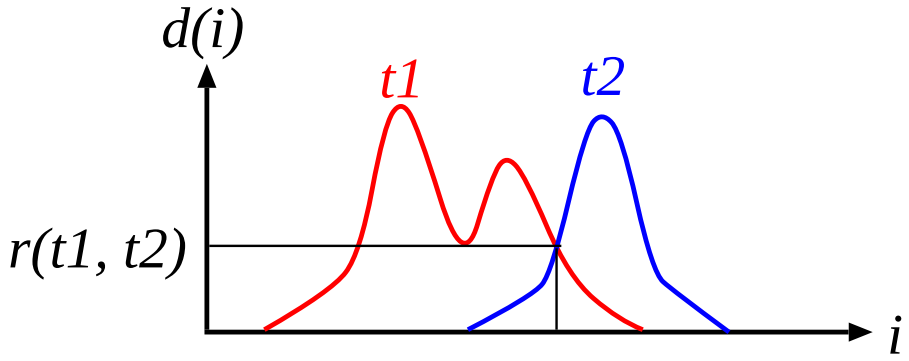


Step 3: Ranking Method

Appearance Density of Words

$r(t_1, t_2)$, the degree of the interrelation of t_1 and t_2:

$$r(t_1, t_2) = \max_{1 \le i \le L} \min(d_{t_1}(i), d_{t_2}(i))$$



For word t in the `<title>` tag or the main heading:

$$r(t, t') = 1 \quad \text{for all } t'$$

Step 3: Ranking Method

Appearance Density of Words

We redefine $F(G)$ so that it reflects the interrelation of keywords within pages:

$$F(G) = \sum_{v \in V} \{A \cdot K(v) + B \cdot \sum_{t_i, t_j \in T(v)} r(t_i, t_j) + C\}^{-1}$$

- We will find appropriate values for A, B, C by experiments.

Preliminary Experiments (1/2)

Query 1: {notebook, card, catalog}

"collect all the catalog pages of PC cards for laptop computers"

n	1	2	3	4	5
# of ans.	980	27	21	0	0
# of correct ans.	682	17	12	0	0
precision ratio	0.696	0.629	0.571	–	–
(accumulative)	0.696	0.694	0.691	0.691	0.691
recall ratio†	0.959	0.983	1.000	1.000	1.000

† We assume there is no answer with $n > 5$

Preliminary Experiments (2/2)

Query 2: {"Ryouichi Sano," "Kobe University"}

"find the homepage of Ryouichi Sano, who is a student of Kobe University"

n	1	2	3	4	5
# of ans.	0	0	1	1	0
# of correct ans.	0	0	0	1	0

Discussion (1/3)

When is the minimal subgraph approach useful?

- recall-oriented queries, e.g. “collect **all** pages that . . .”
- to find some specific page. Some keywords that is expected to be appropriate for the query may happen to appear only in neighboring pages.

Discussion (2/3)

Semantics of Conjunctive Web Queries

Conjunctions are used with various intentions:

also query — a page discussing t_1, and also t_2.

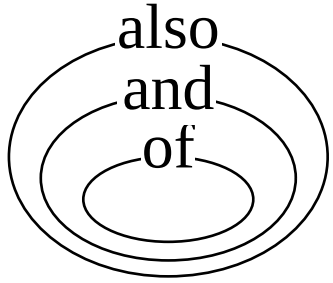
t_1 and t_2 may not be related to each other.

and query — a page discussing a topic related to t_1 and t_2.

t_1 and t_2 are cooperatively describing the topic.

of query — a page discussing t_1 of t_2.

General word t_2 and
specialized word t_1
narrow down the topic
stepwise.



Discussion (3/3)

Ranking Method for Each Semantics

- **also** query

  The interrelation between keywords may not be important, and therefore, $K(v)$ and $r(t_1, t_2)$ are not appropriate.

- **and** query

  We mainly focus on this. $K(v)$ and $r(t_1, t_2)$ must be important.

- **of** query

  The shape of the graphs and the order of the appearance of the words may also be useful.

Conclusion

- logical data unit in Web $\neq$ page
- minimal subgraph based retrieval for conjunctive queries
- distinction of "route links" and other links
- ranking subgraphs based on the overlap/distribution of query keywords