Cut as a Querying Unit for WWW, Netnews, and E-mail

Keishi Tajima   Yoshiaki Mizuuchi †   Masatsugu Kitagawa †   Katsumi Tanaka

Department of Computer and Systems Engineering
Kobe University, Japan

(† Currently at NTT DATA Corporation)

Background (1/4)

WWW, Netnews, E-mail

- Main means for information exchange on the internet.
- They all are hypertext data.

	node	link
WWW	page	hyper link
Netnews	article	reference
E-mail	mail	reference

$\boxed{\text{Background (2/4)}}$

<u>Querying Tools for WWW, Netnews, E-mail</u>

Many querying tools:

- WWW search engines
- intelignet news/mail readers

Those systems:

- regard those hypertext data just as collections of nodes, and
- retrieve nodes satisfying a given condition.

$\boxed{\text{Data unit in querying is a node}}$

Background (3/4)

Two Types of Query

- retrieving an already-known page/article out of a huge data bunch.

The querying unit is a **node**.

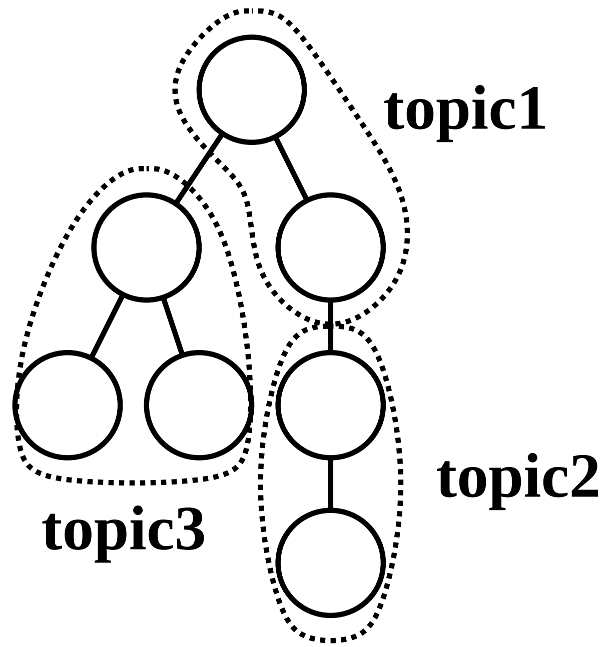
- looking for unknown pages/articles concerned with a topic of current interest. E.g.:
  - find a document on WWW concerned with the design of hypermedia
  - find a discussion in a newsgroup concerned with the design of hypermedia

The querying unit is a **topic** or a **discussion**.

Problem

a document/discussion $\neq$ a node

- A single document on WWW is often divided into multiple pages for browsing.
- A single discussion in a newsgroup (or a mailing list) extends over many following-up articles (or mails).



A document/discussion = a **connected subgraph**.

We call those connected subgraphs **cuts**.

Goal of This Research

The goal of this research is:

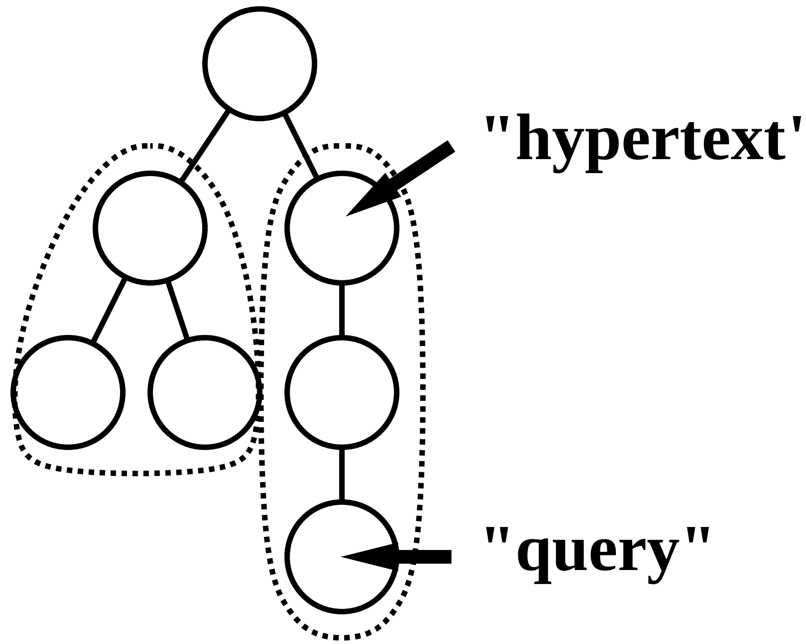
> to develop a query framework for WWW, Netnews, and E-mail which regards a cut as a querying unit.

Nodes v.s. Cuts

how are they different?

Consider a boolean retrieval:

- retrieve all nodes including both the keyword `hypertext` and `query`
- retrieve all cuts including both the keywords `hypertext` and `query`



Those two keywords in a single document may appear in different nodes.

Our Basic Strategy

We first need to develop

a method to detect precise cuts in those hypertext data.

Basic Strategy of cut detection:

- We detect edges where the contents of the neighboring nodes greatly change.
- To compute a similarity between nodes, we use feature vectors of them based on the term frequency.

Similarity between Nodes

Vector Space Model for Documents [Salton 68]

- the feature vector of a document d_i:

$$V(d_i) = (f_{d_i}^1 \cdot F_1, \ldots, f_{d_i}^m \cdot F_m)$$

where

$$\begin{aligned} f_{d_i}^j &= \text{the frequency of the word } j \text{ in } d_i \\ F_j &= \log(\text{the ratio of the documents including } j) \end{aligned}$$

- the similarity of two documents d_i and d_k:

$$\angle(d_i, d_k) = \frac{V(d_i) \cdot V(d_k)}{|V(d_i)|\ |V(d_k)|}$$

$\boxed{\text{Cut Detection (1/6)}}$

A Simple Approach

A well-known algorithm for edge-weighted graph partitioning:

Input: a graph and a number n

- repeat below until the number of nodes become n
  1. compute the similarity of all pairs of neighboring nodes.
  2. merge two nodes with the highest similarity into one node.

Output: a graph whose nodes represent cuts

Cut Detection (2/6)

Problems of the Simple Approach

Problems found in our experiment with the simple approach are:

- many nodes include more than two topics, and
- nodes with multiple children tend to be merged with all children, and that merging often blocks the merge of those children with their descendant.

To solve those problems,

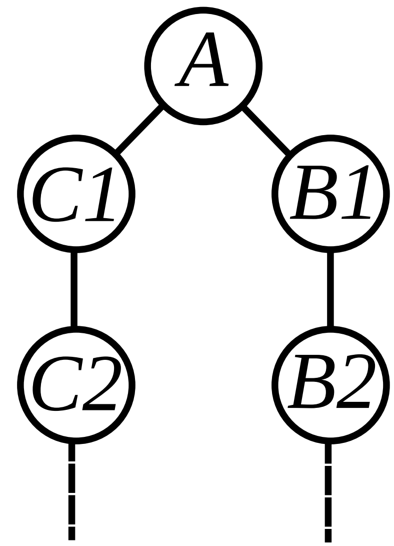
- we should allow overlap of cuts, and
- when selecting nodes to merge, we give priority to leaves.

Cut Detection (3/6)

An Example of the Problem

Experimental data: a tree consisting of 46 articles in `fj.soc.smoking`.

- A tells an experience of the death of a kin.
- $B_1, B_2, \ldots$ discuss the misery of terminal patients of cancer.
- $C_1, C_2, \ldots$ discuss the criticisms to hospitals.



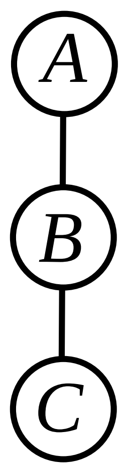
A is merged with B_1 and C_1 in early phases, and it blocks the merge of B_1 with B_2 or C_1 with C_2.

Cut Detection (4/6)

Our Algorithm

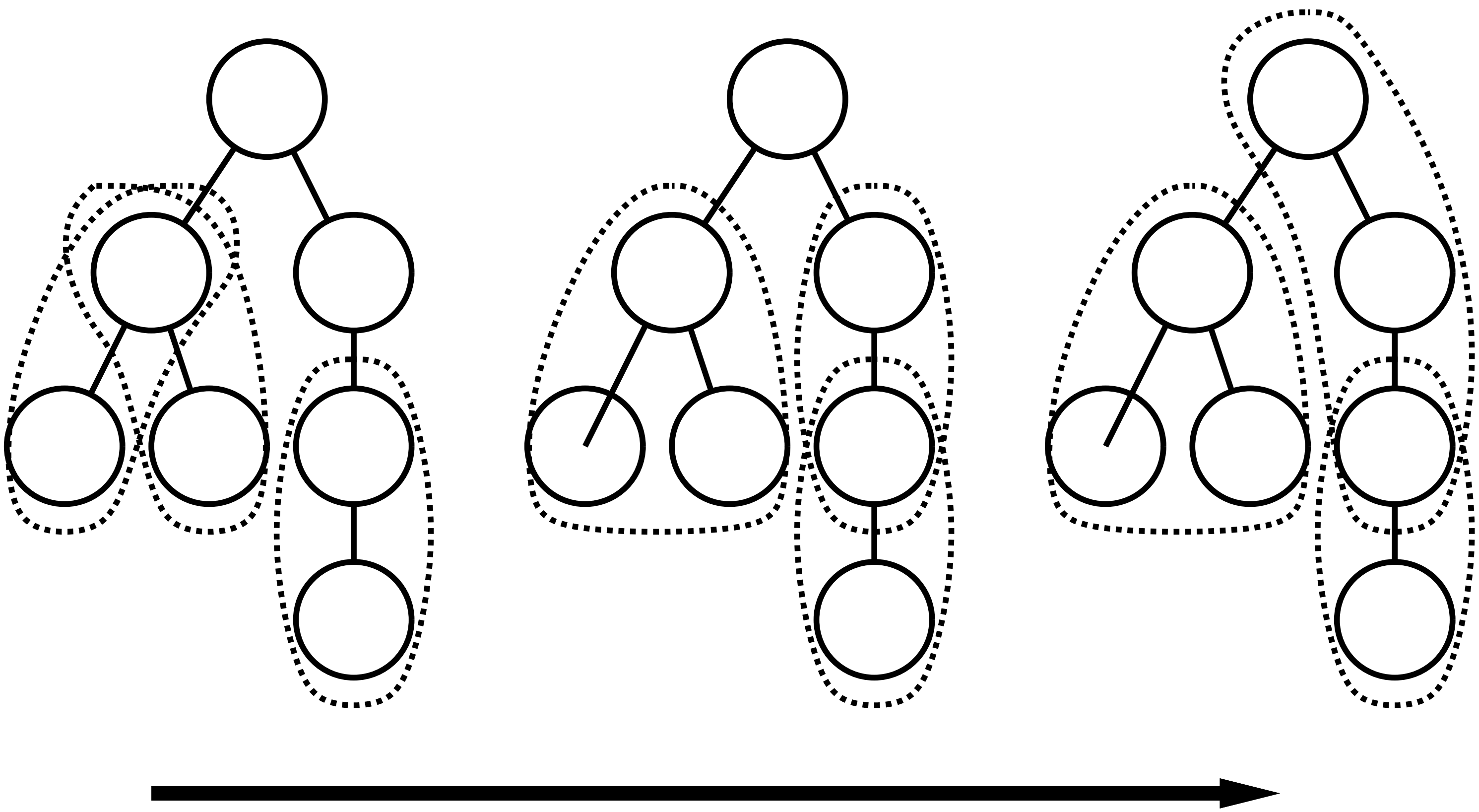
Overview of our algorithm:

1. We start comparison of nodes at leaves of the tree, and proceed upward in the breadth-first manner.
2. First we compare each leaf with its parent node. If multiple siblings are merged with their parent node, we also compare those siblings.
3. If B and C are merged, then we compare $\{B, C\}$ and A. If they are not similar, we also compare B and A, and if they are similar, we create a new cut $\{A, B\}$.



Cut Detection (5/6)

An example of our algorithm



Cut Detection (6/6)

Evaluation of the Results

- Our pragmatic, ad-hoc algorithm produces intuitively better results than that of the simple algorithm for newsgroups or mailing lists data.
- Both simple algorithm and our algorithm can produce satisfactory results for WWW data only when there is a clearly separated document consisting of multiple pages.

We need to design a proper measure of the correctness of the result for quantitative analysis.

Related Work (1/2)

1. Document clustering
   — partitioning a set of documents into subsets of similar documents.
   - We partition a graph into **connected** subgraphs corresponding to **logical data units**.
2. Retrieval of hypertext data using link information [Croft 89, Weiss 96]
   — use of information of neighboring nodes.
   - They do not detect how far a logical data unit expands.
3. Subtopic structuring of documents [Hearst 93, Nomoto 94]
   — detect subtopic structure in sequential documents.
   - We apply the same concept to hypertext data.

Related Work (2/2)

4. Aggregating hypertext data [Botafogo 91]
   — detecting substructure in hypertext data in order to produce an overview map of the whole structure.
   - They use information on link structure while we use information on contents similarity.
5. Structural Query
   — e.g.:

   select a $\longrightarrow^*$ b
   where include(a, `'WWW'`) $\wedge$ include(b, `'query'`)

   - They do not detect how far a logical data unit expands.

Conclusion

- We propose the concept of cuts for querying unit in hypertext data.
- We developed a method to detect precise cuts in WWW, Netnews, or E-mail data.

Future work

- To design a proper measure of the correctness of graph partitioning allowing overlaps.
- Quantitative evaluation of our approach by using that measure.
- Comparison with other retrieval model, such as probabilistic clustering of WWW pages.