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)
WWW, Netnews, E-mail

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

<table>
<thead>
<tr>
<th></th>
<th>node</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td>WWW</td>
<td>page</td>
<td>hyper link</td>
</tr>
<tr>
<td>Netnews</td>
<td>article</td>
<td>reference</td>
</tr>
<tr>
<td>E-mail</td>
<td>mail</td>
<td>reference</td>
</tr>
</tbody>
</table>
Querying Tools for WWW, Netnews, E-mail

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.

Data unit in querying is a node
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

\[ \text{a document/discussion} \neq \text{a node} \]

- 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.
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 \texttt{hypertext} and \texttt{query}
- retrieve all cuts including both the keywords \texttt{hypertext} and \texttt{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 $d_i$:

$$V(d_i) = (f^1_{d_i} \cdot F_1, \ldots, f^m_{d_i} \cdot F_m)$$

where

$$f^j_{d_i} = \text{the frequency of the word } j \text{ in } d_i$$
$$F_j = \log(\text{the ratio of the documents including } j)$$

- the similarity of two documents $d_i$ and $d_k$:

$$\angle(d_i, d_k) = \frac{V(d_i) \cdot V(d_k)}{|V(d_i)| \cdot |V(d_k)|}$$
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
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.
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, ... discuss the misery of terminal patients of cancer.
- *C*₁, *C*₂, ... 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.
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 $\{A, B\}$. 

\begin{figure}[h]
\centering
\begin{tikzpicture}
  \node (A) at (0,0) [circle, draw] {A};
  \node (B) at (0,-1) [circle, draw] {B};
  \node (C) at (0,-2) [circle, draw] {C};
  \draw (A) -- (B);
  \draw (B) -- (C);
\end{tikzpicture}
\end{figure}
An example of our algorithm
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.
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.
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 →* b
where include(a, 'WWW') ∧ 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.