A Method for Relating Multiple Newspaper Articles by Using Graphs, and Its Application to Webcasting

Naohiko Uramoto and **Koichi Takeda**
IBM Research, Tokyo Research Laboratory
1623-14 Shimo-tsuruma, Yamato-shi, Kanagawa-ken 242 Japan
{uramoto,takeda}@trl.ibm.co.jp

Abstract

This paper describes methods for relating (threading) multiple newspaper articles, and for visualizing various characteristics of them by using a directed graph. A set of articles is represented by a set of word vectors, and the similarity between the vectors is then calculated. The graph is constructed from the similarity matrix. By applying some constraints on the chronological ordering of articles, an efficient threading algorithm that runs in $O(n)$ time (where n is the number of articles) is obtained. The constructed graph is visualized with words that represent the topics of the threads, and words that represent new information in each article. The threading technique is suitable for Webcasting (push) applications. A threading server determines relationships among articles from various news sources, and creates files containing their threading information. This information is represented in eXtended Markup Language (XML), and can be visualized on most Web browsers. The XML-based representation and a current prototype are described in this paper.

1 Introduction

The vast quantity of information available today makes it difficult to search for and understand the information that we want. If there are many related documents about a topic, it is important to capture their relationships so that we can obtain a clearer overview. However, most information resources, including newspaper articles do not have explicit relationships. For example, although documents on the Web are connected by hyperlinks, relationships cannot be specified.

Webcasting ("push") applications such as Pointcast [1] constitute a promising solution to the problem of information overloading, but the articles they provide do not have links, or else must be manually linked at a high cost in terms of time and effort.

This paper describes methods for relating newspaper articles automatically, and its application for a Webcasting application. A set of article on a particular topic is ordered chronologically, and the results are represented as a directed graph. There are various ways of relating documents and visualizing their structure. For example, USENET articles can be accessed by means of newsreader software. In the system, a label (title) is attached to each posted message, specifying whether it deals with a new topic or is a reply to a previous message. A chain of articles on a topic is called a *thread*. In this case, the relationships between the articles are explicitly defined. This post/reply-based approach makes it possible for a reader to group all the messages on a particular topic. However, it is difficult to capture the story of the thread from its thread structure, since appropriate titles are not added to the messages.

This paper aims to provide ways of relating multiple news articles and representing their structure in a way that is easy to understand and computationally inexpensive. A set of relationships is defined here as a directed graph. A node indicates an article, and an arc from node X to Y indicates that the article X is followed by Y (or that X is adjacent to Y). An article contains both known and unknown (new) information. Known information consists of words shared by the beginning and ending points of an arc. When node X is adjacent to Y, the words are represented by $(X \cap Y)$. The known information is called *genus words* in this paper. Even if an article follows another one, it generally contains some new information. This information can be represented by subtraction $(Y - X)$ (Damashek, 1995), and is called ***differentia words***, by analogy with definition sentences in dictionaries, which contain genus words and differentia. In this paper, genus and differentiae words are used to calculate the similarities between two articles, and to visualize topics in a set of articles.

Since articles are ordered chronologically, there are some time constraints on the connectivity of nodes. A graph is created by constructing an adjacency matrix for nodes, which in turn is created from a similarity matrix for nodes.

Some potential features of articles in a set can be determined by analyzing some formal aspects of the

[1] http://www.pointcast.com

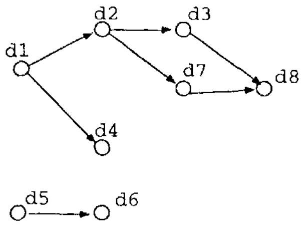


Figure 1: Example of a Directed Graph G

corresponding graph. For example, the paths in the graph show the stories of the nodes they contain. Multiple paths for a node (article) show that there are multiple stories associated with it. Furthermore, if the node has a long path, it is in the "main stream" of the topic represented by the graph. An efficient algorithm for finding such paths is described, later in the paper.

Application of the threading method to documents on the Web would be very useful because, although such documents are connected by hyperlinks, their relationships cannot be specified. In this paper, generated threads by this method are represented in eXtended Markup Language (XML) (XML, 1997), which is the proposed standard for exchange of information on the Web. XML-based threads can be used by webcasting or push services, since various tools for parsing and visualizing threads are available.

In Section 2, a directed graph structure for articles is defined, and the procedure for constructing a directed graph is described in Section 3. In Section 4, some features of the created graph are discussed. Section 5 introduces a webcasting application by using the threading technique, and Section 6 concludes the paper.

2 Definition of a Graph Structure

A set of articles is represented as an ordered set V: $V = \{d_1, d_2, \ldots, d_n\}$.

The suffix sequence $1, 2, \ldots, n$ represents the passage of time. Article d_i is older than d_{i+1}. The order is obtained from the publication dates of the articles. Different time points arbitrarily are assigned to articles published on the same day.

Related articles are represented as a directed graph (V, A). V is a set of nodes. A is a set of ordered pairs (i, j), where i and j are members of V. Figure 1 shows an example of a directed graph. In this case, the graph is represented as follows:

$$V = \{d_1, d_2, d_3, d_4, d_5, d_6, d_6, d_7\}, \quad A = \{(d_1, d_2), (d_2, d_3), (d_1, d_4), (d_5, d_6), (d_2, d_7), (d_3, d_8), (d_7, d_8)\}$$

The nodes are ordered chronologically. The following constraint is introduced into the graph:

$$M = \begin{array}{c|cccccccc} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 \\ d_1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ d_2 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ d_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ d_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ d_5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ d_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ d_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ d_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Figure 3: Adjacency Matrix M_G of G

Constraint 1
For $(d_i, d_j) \in A, \ i < j$

The constraint simply shows that an old article cannot follow a new one.

3 Creating a Graph Structure for Articles

This section describes how to construct a directed graph structure from a set of articles. Any directed graph can be represented by a matrix. Figure 3 shows the adjacency matrix M_G of the graph G in Figure 1.

For example, a value of "1" for the (1, 2) element in M indicates that d_1 is adjacent to d_2. Since an article cannot follow itself, the value of (i, i) elements is "0". From the time constraint defined in Section 3, M_G is an upper triangle matrix.

The following is a procedure for constructing a directed graph for related articles:

1. Calculate the similarity and difference between articles.
2. Construct a similarity matrix.
3. Convert the matrix into an adjacency matrix.

In the next section, each step is illustrated by using the set of articles V in Figure 2 on the subject of nuclear testing taken from the Nikkei Shinbun.[2]

3.1 Calculating the similarities and differences between articles

The function $sim(d_i, d_j)$ calculates the word-based similarity between two articles. It is defined on the basis of Salton's Vector Space Model (Salton, 1968). Words are extracted from an article by using a morphological analyzer. Next, nouns and verbs are extracted as keywords.

$$sim(d_i, d_j) = \frac{\sum_{kw} w_{kw}^{d_i} w_{kw}^{d_j}}{\sqrt{\sum_{kw} (w_{kw}^{d_i})^2} \sqrt{\sum_{kw} (w_{kw}^{d_j})^2}}$$

.

[2] The articles were originally written in Japanese.

d_1: The prime minister of France says that it is necessary to restart nuclear testing.
d_2: The Defense Minister suggests restarting nuclear testing.
d_3: At a summit conference, the Prime Minister will adopt a policy of requesting the French Government to halt nuclear testing.
d_4: China's latest nuclear test will hold up negotiations on a treaty to abolish such testing.
d_5: The Minister of Foreign Affairs, Mr. Youhei Kohno, takes a critical attitude toward China, and asks France to understand Japan's position.
d_6: The prime minister of New Zealand asks the French Government not to restart nuclear testing.
d_7: President of France states that nuclear testing will restart in September, and that France will conduct eight tests between now and next May.
d_8: France states that it will restart nuclear testing. This will hamper nuclear disarmament.
d_9: France states that it will restart nuclear testing. Australia halts defense cooperation with France.
d_{10}: France states that it will restart nuclear testing. The U.S. expresses regret at the decision.

Figure 2: V: Articles about nuclear testing

Here, $w^{d_i}_{kw}$ is the weight given to the keyword kw in article d_i. Modification of the TF·IDF value (Robertson *et al.*, 1976) is used for the weighting. $g^{d_i}_{kw}$ is the weight assigned to the keyword kw, which is a differentia word for d_i.

$$w^{d_i}_{kw} = \frac{C_{d_i}(kw)}{C_{d_i}} \cdot log\frac{k}{N_k(kw)} \cdot g^{d_i}_{kw},$$

$$g^{d_i}_{kw} = \begin{cases} 1.5 & kw \in differentia(d_i) \\ 1 & otherwise. \end{cases}$$

Other parameters are defined as follows:

k: constant value
$C_{d_i}(kw)$: frequency of word kw in $d_(i)$
C_{d_i}: number of words in $d_(i)$
$N_k(kw)$: number of articles that contain the word kw in k articles $d_{i-k}, \ldots, d_i$

The function *differentia*(d_i) returns a set of keywords that appear in d_j but do not appear in the last k articles.

$differentia(d_i) = \{kw \mid C_{d_i}(kw) > 0$, and for all d_l,
$where\ i - k < l < i, C_{d_l}(kw) = 0\}$

3.2 Constructing a similarity matrix

A similarity matrix for a set of articles is constructed by using the *sim* function. In a conventional hierarchical clustering algorithm, a similarity for any combination of two articles is required in order to construct a hierarchical tree of the set of articles. This causes $\frac{n(n-1)}{2}$ calculations of the similarity function, for n articles, with a consequent complexity of $O(n^2)$. This is very expensive when n is large. In our algorithm for constructing a similarity matrix, shown in Figure 4, the complexity of constructing a graph structure for an article set by using a constraint is $O(n)$. The following constraint, which includes Constraint 1, is used for in threading algorithm.

```
procedure MakeDistanceMatrix
for i= 2 to n begin
    if i - k < 1 then s ← 1 else s ← i - k
    for j = s to i - 1 begin
        $a(i, j) \leftarrow sim(d_i, d_j)$
        j ← j + 1
    end
    i ← i + 1
end
```

Figure 4: Procedure for Constructing Similarity Matrix

Constraint 2
For $(d_i, d_j) \in A,\ j - (k + 1) < i < j$

This constraint means that an article can only follow the last k articles. As the result, the number of times the similarity matrix needs to be calculated is reduced by kn, giving a complexity of $O(n)$.

By using the algorithm, each similarity between nodes is calculated, and the similarity matrix in Figure 5 shows a similarity matrix S of V. In this case, keywords are extracted from title sentences, and k is set to five.

3.3 Conversion into an adjacency matrix

From the similarity matrix, an adjacency matrix is constructed. An element $s(i, j)$ in the similarity matrix corresponds to the element $ss(i, j)$ in the adjacency matrix SS. There are various strategies for the conversion. In this paper, $ss(i, j)$ is set to 1 when s(i, j) > 0.18, and any node can follow at most k/2 nodes, in this case two nodes. Figure 6 shows a result of the conversion. Finally, a directed graph for V is created (Figure 7). Figure 8 shows a graph that visualizes the content of the articles in our example.

$$S = \begin{array}{c|cccccccccc|} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 & d_9 & d_{10} \\ d_1 & 0 & .309 & .239 & .072 & .131 & .319 & 0 & 0 & 0 & 0 \\ d_2 & 0 & 0 & .159 & .072 & .131 & .319 & .197 & 0 & 0 & 0 \\ d_3 & 0 & 0 & 0 & .056 & .103 & .498 & .103 & .124 & 0 & 0 \\ d_4 & 0 & 0 & 0 & 0 & .186 & .056 & .046 & .056 & .046 & 0 \\ d_5 & 0 & 0 & 0 & 0 & 0 & .102 & .085 & .102 & .128 & .096 \\ d_6 & 0 & 0 & 0 & 0 & 0 & 0 & .154 & .176 & .206 & .209 \\ d_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .308 & .320 & .323 \\ d_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .257 & .279 \\ d_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .287 \\ d_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Figure 5: Similarity Matrix S

$$\begin{array}{c|cccccccccc|} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 & d_9 & d_{10} \\ d_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ d_2 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ d_3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ d_4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ d_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ d_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ d_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ d_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ d_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ d_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Figure 6: Adjacency Matrix SS Converted from S

There are two threads in the graph. One concerns for France's restarting of nuclear testing. The other concerns China's latest nuclear test. The "France" thread contains two sub-threads. One concern requests by other countries for France to reconsider its stated intension of restarting nuclear testing, and the other concerns responses by other countries to the France government's official statement on testing. Some articles are followed by multiple articles. For example, d_7 is the first official statement on France's restarting of nuclear testing, and many related articles on this topic follow.

Each rectangle in Figure 8 represents an article. Words in a rectangle are differentia words for the articles. These words show new information in the article, and make it easy to understand the content of the articles. If a word in an article appears in the differentia words for its parent article, the word may represent a "turning point" in the story of the articles. For example, the word "state" is the differentia word for d_7, and is in its adjacent articles $d_8, d_9, and d_{10}$. This means that d_7 is a starting point of the new topic "state." Such words are called *topic words*, and are represented in Figure 8 by bold type.

Several features of the graph visualize the charac-

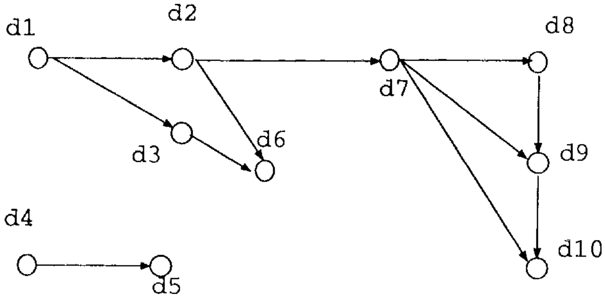


Figure 7: Directed Graph G_1 for V

teristics and relationships of the articles: these features will be discussed in the next section.

It is difficult to evaluate the result of threading. We are implementing it in a webcasting (push) application so that it can be evaluated by the many people who use ordinary web browsers. The attempt is described in Section 5.

4 Features of a Graph

This section describes how the features of a constructed graph represent the characteristics of articles.

4.1 In-degree and Out-degree

The *in-degree* is the number of arcs leading to a node, while the out-degree is the number of arcs leading from it. The in-degree of d_i can be calculated by adding up the elements in the i-th column of an adjacency matrix. The out-degree of d_i can be calculated by adding up the elements in the i-th row of the matrix (Figure 9). In Botafogo et al. (Botafogo *et al.*, 1992), a node that has a high out-degree is called an *index node*, while a node that has a high in-degree is called a *reference node* in their analysis of hypertext. In the set of articles V shown in Figure 9, d_7 is an index node. In this paper, an index node denotes the beginning of a new topic. When the topic is important, many articles follow, and consequently the out-

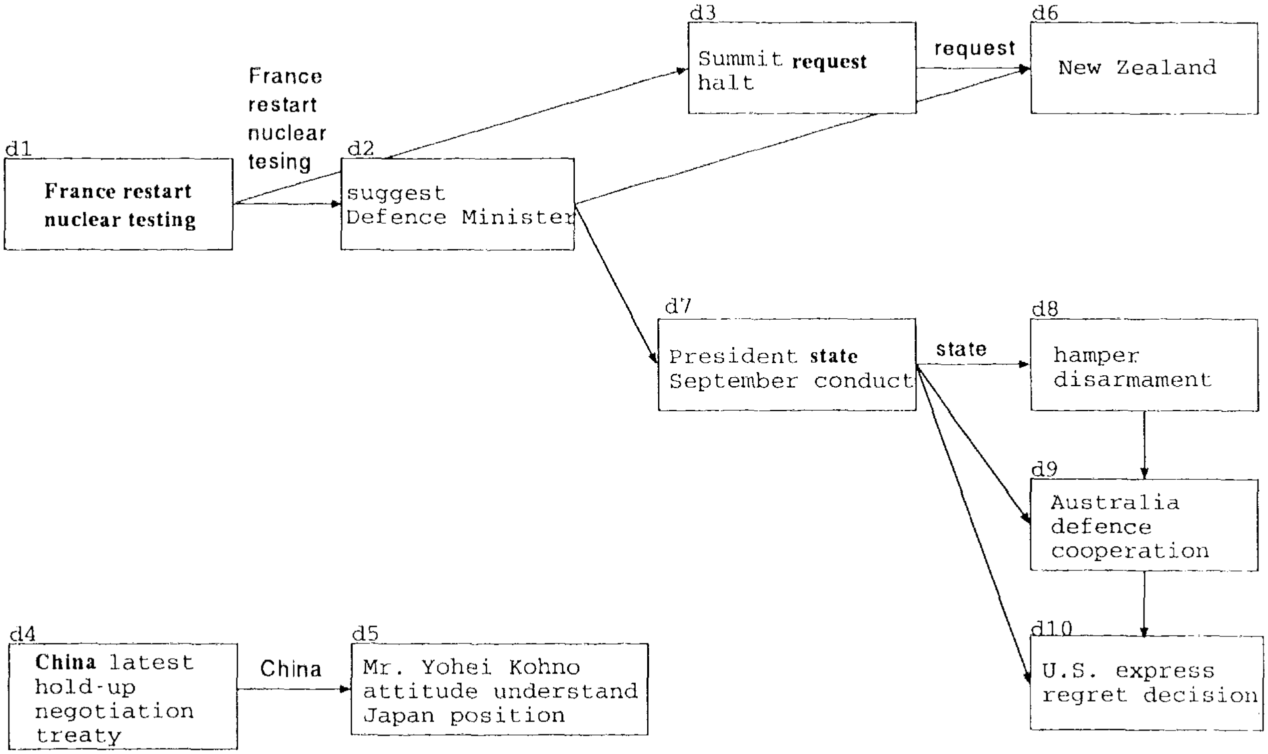

d1
France restart nuclear testing
France restart nuclear tesing
d2
suggest Defence Minister
d3
Summit request halt
request
d6
New Zealand
d7
President state September conduct
state
d8
hamper disarmament
d9
Australia defence cooperation
d10
U.S. express regret decision
d4
China latest hold-up negotiation treaty
China
d5
Mr. Yohei Kohno attitude understand Japan position

Figure 8: Visualized Content for G_1

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
in	0	1	1	0	1	2	1	1	2	2
out	2	2	1	1	0	0	3	1	1	0

Figure 9: In-degree/Out-degree of the Graph G_1

degree for the node increases. The contribution of reference nodes is not clear in V (d_6, d_8, and d_9 have max in-degrees). Nodes that have high in-degree have two characteristics. The first is that when the articles contain multiple topics, they have many inbound arcs, each representing a different topic. The second is that when the articles are closely related for a particular topic, the in-degrees of related nodes increase, since these articles are connected to each other.

4.2 Path

A path from one node to another node shows the "story flow" of articles. Multiple paths between two nodes show different stories about the nodes. For example, there are three paths between d_1, which is a first node, and d_{10}. The shortest path $(d_1, d_2, , d_7, d_{10})$ gives a simple outline of the articles. The longest path $(d_1, d_2, d_7, d_8, d_9, d_{10})$ contains all related information on the topic. By extracting long paths from the graph and combining them, various stories can be created.

The length of a path shows how the nodes on it belong to the "main stream" of the story. For example, the maximum length of a path through d_6, is three, while that of a path through d_7 is five. This means that a path that contains d_7 is on a main stream of the thread and is likely to be continued.

The longest paths for nodes can be calculated by using the algorithm shown in Figure 11. Its complexity is $O(n)$, since the maximum number of arcs is at most nk for n nodes, from Constraint 2, defined in Section 3.2.

4.3 Cycle

A cycle[3] shows the existence of a topic. In V, $\{d_7, d_8, d_9, d_{10}\}$ is a cycle for the topic "statement." By recognizing cycles, we can extract topics from the whole graph. Furthermore, we can abstract articles by reducing cycles to single nodes.

5 XML-based Representation for Threads

It is important that the threading information be exchangeable when we apply our method to Web documents. Extended Markup Language (XML) is a proposed standard (XML, 1997) specified by the World Wide Web Consortium (W3C). In XML, tags and

[3] Formally, it is called a semi-cycle, since the graph is directed.

attributes can be defined, whereas in HTML they are fixed. XML documents can be used to exchange information that has various data structure. For example, Channel Definition Format (CDF)(CDF, 1997) is a standard to offer frequently updated collections of information (channels) on Web. A CDF document can contains a collection of articles that have tree structure. In this paper, graph structures of created threads are represented in XML. Figure 10 shows a part of the thread in Figure 8.

The <thread> tag shows the beginning of the thread. It contains a set of deceptions for articles, each marked <article>. Each instance of the <article> tag has a reference to its source document, an identifying id, genus and differentia words, and other information on the article. The tag <follows> is used to denote arcs from the article to related articles.

The XML documents can be separate from the source articles. They can be provided as part of a "push" service for Internet users, offering a solution to the problem of information overloading. In such a service, *gatherer* collects articles from Web sites and *threader* makes threads for them. The results are stored in XML, and then pushed to subscribers who can capture the flow of topics by following the threads. In another scenario, when a user gets an article, and wants to see its origin or the next related article, he or she gets the thread containing the article by consulting the threading server. The advantage of using XML is that it will be supported by various tools, including Web browsers. Now we are prototyping the threading service system by using a XML processor developed at our laboratory. Figure 12 shows a Java applet for viewing threads, which can run on major Web browsers. A XML document is parsed and visualized as tree-like structure.

6 Related Work

There have been several studies how to relate articles (McKeown *et al.*, 1995; Yamamoto *et al.*, 1995; Mani *et al.*, 1997). McKeown et al. reported a method for summarizing news articles (McKeown *et al.*, 1995). In their approach, templates, which have slots and their values (for example, incident-location="New York"), are extracted from the articles. Summary sentences are constructed by combining the templates. Although this approach can capture topics contained in the articles, the relationships between articles are not visualized.

Clustering techniques make it possible to visualize the contents of a set of documents. Hearst et al. proposed the scatter/gather approach for facilitating information retrieval (Hearst *et al.*, 1995). Maarek et al. related documents by using an hierarchical clustering algorithm that interacts with the user. Although these clustering algorithms impose a heavy computation cost, our threading algorithm is efficient, because it uses a chronological constraint.

```
procedure GetMaxtPath($A$)
// Get max path MaxPath[i] for $d_i$. $A$ is a set of arcs.

for i = 1 to n begin MaxPath[i] ← NULL end
for j = 1 to n begin
  for i = j - k to j - 1 begin
    if $(d_i, d_j) \in A$ then
      if Length(MaxPath[j]) < Length(MaxPath[i]) + 1
        then MaxPath[j] ← Connect(MaxPath[i],($d_i$,$d_j$))
    i ← i + 1
  end
  j ← j + 1
end

procedure Length($path$)
returns the number of arcs in $path$.

procedure Connect($path$, $arc$)
if path = $(d_0, \ldots, d_i)$ and arc = $(d_i, d_j)$, then
return $(d_0, \ldots, d_i, d_j)$.
```

Figure 11: Procedure for Finding the Longest Path

7 Conclusion

We have described methods for threading multiple articles and for visualizing various characteristics of them by using directed graphs. An efficient threading algorithm whose complexity is $O(n)$ (where n is the number of articles) was introduced with some constraints on the chronological ordering of articles.

Some further work can be done to improve our method. There are some strategies for constructing an adjacency matrix from a distance matrix. Different strategies give different graphs. We are now evaluating our method by testing it with various strategies.

The development of a technique for visualizing directed graphs is another task for the future. Although directed graphs show more useful information than tree structures, they are difficult to display in a readily understandable way. Software tools for handling graphs are also required.

Formal features of graphs can express the underlying characteristics of articles. More efficient and useful algorithms are needed to overcome the problem of information overload.

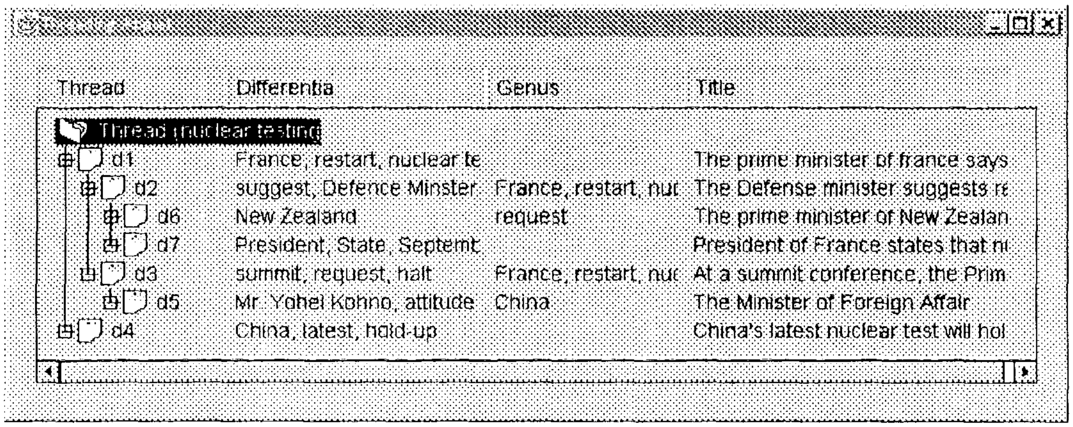
References

R. Botafogo, E. Rivlin, and B Shnederman. 1992. *Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics. ACM Transaction on Information Science*, pages 143–179, Vol. 10, No. 2.

C. Ellerman. 1997.

```
<thread id="thread1">
 <article id="d1" HREF="foo.bar.com/article/d1.html">
   <title>The prime minister of France says that it is necessary to
   restart nuclear testing.</title>
   <genus></genus>
   <diff>France, restart, nuclear testing</diff>
   <follows HREF="#d2"/>
   <follows HREF="#d3"/>
 </article>
 <article id="d2" HREF="foo.bar.com/article/d2.html">
   <title>The Defense Minister of France suggests restarting nuclear testing.</title>
   <genus>nuclear testing, restart, France</genus>
   <diff>suggest, Defense minister</diff>
   <follows HREF="#d6"/>
   <follows HREF="#d7"/>
 </article>
 ...
</thread>
```

Figure 10: XML-Based Presentation of the Thread



Thread	Differentia	Genus	Title
Thread (nuclear testing			
d1	France, restart, nuclear te		The prime minister of france says
d2	suggest, Defence Minster	France, restart, nuc	The Defense minister suggests re
d6	New Zealand	request	The prime minister of New Zealan
d7	President, State, Septemt		President of France states that n
d3	summit, request, halt	France, restart, nuc	At a summit conference, the Prim
d5	Mr. Yohei Kohno, attitude	China	The Minister of Foreign Affair
d4	China, latest, hold-up		China's latest nuclear test will hol

Figure 12: Thread Viewer Applet

Channel Definition Format (CDF). http://www.microsoft.com/standards/cdf.htm.

M. Damashek. 1995. *Gauging Similarity with n-Grams: Language Independent Categorization of Text.* Proc. of *Science*, pages 843–848, Vol. 267.

M. A. Hearst, D. R. Karger, and J. O. Pederson. 1995. *Scatter/Gather as a Tool for Navigation of Retrieval Results.* Proc. of *AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval.*

N. Jardine, and R. Sibson. 1968. *The Construction of Hierarchic and Non-Hierarchic Classifications. Computer*, pages 177–184.

I. Mani and E. Bloedorn. 1997. *Multi-document Summarization by Graph Search and Matching* Proc. of *AAAI'97*, pages. 622–628.

Y. Maarek and A. Wecker. 1994. *The Librarian Assistant: Automatically Assembling Books into Dynamic Bookshelves.* Proc. of *RIAO.*

K. McKeown and D. Radev. 1995. *Generating Summaries of Multiple News Articles.* Proc. of *SIGIR*, pages 74–82.

S. E. Robertson and K. S. Jones. 1976. *Relevance Weighting of Search Terms. JASIS*, pages 129–146, Vol. 27.

G. Salton. 1968. *Automatic Information Organization and Retrieval.* New York, NY: McGraw-Hill.

T. Bray, J. Paoli, and C. M. Sperberg-McQeen. 1997 *Extensible Markup Language (XML).* Proposed Recommendation. World Wide Web Consortium. http://www.w3.org/TR/PR-xml/

K. Yamamoto, S. Masuyama, and S. Naito. 1995. *An Empirical Study on Summarizing Multiple Texts of Japanese Newspaper Articles.* Proc. of *NLPRS'95*, pages 461–466.