

A Web Search Method Based on the Temporal Relation of Query Keywords

Tomoyo Kage and Kazutoshi Sumiya

University of Hyogo
1-1-12 Shinzaike-honcho, Himeji, 670-0092, Hyogo, Japan
nd05w005@stshse.u-hyogo.ac.jp
sumiya@shse.u-hyogo.ac.jp

Abstract. As use of the Web has become more popular, searching for particular content has been refined by allowing users to enter multiple keywords as queries. However, simple combinations of multiple query keywords may not generate satisfactory search results. We therefore propose a search method which automatically combines query keywords to generate queries by extracting the relations among query keywords. This method consists of two Web search processes: one to determine the temporal relations between query keywords, and one to generate queries based on the obtained temporal relations. We discuss these two processes along with experimental results and implementation issues regarding a prototype system.

Keywords: Information Retrieval, Temporal Relation, Web Archive, Query Generation.

1 Introduction

Current Web search engines based on the relations between keywords and Web pages only consider the presence or absence of keywords. For that reason, if users input some keywords as query keywords to a Web search engine, it outputs only Web pages that include all the query keywords. For example, suppose a user inputs $\{\textit{cherry blossoms}, \textit{autumn leaves}, \textit{trip}\}$ to a Web search engine. Current search engines will output Web pages that include all the query keywords; i.e., pages containing information about sightseeing. However, in some cases the user is likely to want information from travel agency pages which provide tour information or from personal sites where trip topics are discussed.

In this study, the automatic generation of meaningful queries using OR bindings makes it possible to get useful Web pages. We determine the relation between query keywords on Web pages as a temporal relation to generate the queries. This temporal relation is extracted from a time series of Web pages. The time series of Web pages we use are collected from Web archives¹.

¹ <http://www.archive.org/>

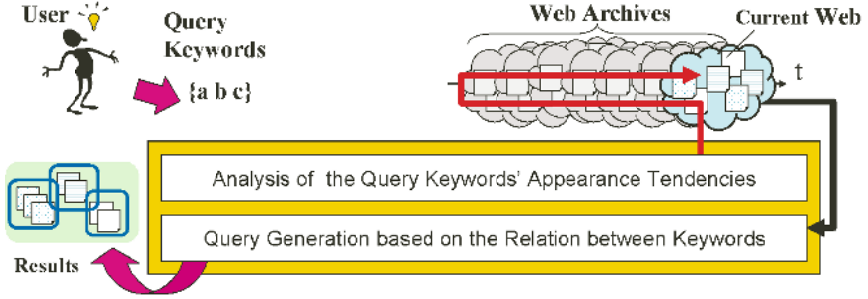


Fig. 1. Concept of Our Approach

In this paper, we propose using the temporal relations between query keywords to generate query keyword combinations (Fig. 1). We analyze the appearance tendencies of query keywords, generate queries by extracting the temporal relation, and then get Web pages which cannot be obtained by current engines.

Where the query keywords are $\{cherry\ blossoms, autumn\ leaves, trip\}$, the characteristic of $\{cherry\ blossoms, autumn\ leaves\}$ alternately appearing in a time series of Web pages is extracted, and queries returning $\{R(cherry\ blossoms \wedge trip) \vee R(autumn\ leaves \wedge trip)\}$ are generated. That is, the system will find that there is a fairly small chance of these keywords appearing at the same time. Users can get travel agency pages including tour information through the query. Furthermore, this method is much more efficient than a method that generates all combinations joined by AND binding or OR binding. This is because only meaningful queries are generated by using the relation between query keywords.

Related work on query generation includes the multimedia retrieval of Web pages by query relaxation [8] and interval glue operations for video data retrieval [9]; these methods differ from ours, however, because their retrieval target is different. Regarding studies using time series pages, approaches specialized for the purpose of summarization [1][2][3][7] have been proposed. Furthermore, [5][6] aim at search engine ranking. They deal with temporal Web pages with the same target as our method, but the goals and methods differ from ours. Research on the temporal relation of keywords includes [10] and [4]. Temporal characteristics are extracted from the query logs in the search engine; however, we extract these from Web logs, and so their approach differs from ours. A search engine which applies clustering to search results is also available [11]. This search engine classifies subtopics for query keywords by analyzing Web page summaries, while we use the temporal features of query keywords.

The remainder of this paper is organized as follows. Section 2 describes the process to determine the temporal relation that we propose, and Section 3 describes the query generation. Section 4 looks at an experiment to test the proposed method and the design of the prototype system. In Section 5, we summarize this work.

2 Temporal Relation of Query Keywords

2.1 Definition

We define the temporal relations of query keywords that users input into a Web search engine. There may be several kinds of temporal relation between the keywords. We make connections between the relations. Three relations — co-occurring, ordered, and repeated — plus the independent relation, are all basic temporal relations. The strict co-occurring relation is a special case of co-occurring in that it is a strictly simultaneous co-occurrence, while the cyclic relation is a special case of repeated in that the time intervals are exactly the same. The temporal relations are defined as follows.

Co-occurring. The co-occurring relation is where keywords a and b both appear at the same time on time series pages. General co-occurrence means that multiple keywords appear on a Web page. However, we do not pinpoint a page because the scale is time. Consequently, if keywords a and b appear at more or less the same time, we say that they have a co-occurring relation. For example, a keyword pair $\{SARS, corona\}$ has a co-occurring relation. In this case, a tight relation exists between the keywords, and a strict co-occurring relation is classified as a strong relation of this type. This is described in more detail below.

Ordered. The ordered relation is where keywords a and b are order dependent with respect to each other; i.e., one of them appears before the other in time series pages. That is, with two keywords, either " a appears before b " or " b appears before a ". A concrete example of this relation is that of the keyword pair $\{SBC, AT\&T\}$. In this way, a "causal" semantic relation is included and the relation expresses a topic change. We classify ordered as repeated, though, when there is a strong relation, as described in more detail below.

Repeated. The repeated relation is where keywords a and b appear in a circular pattern in chronological order; i.e., a , and b alternately appear on time series pages. For instance, the keywords pair $\{disaster, volunteer\}$ or $\{remodeling, re-opening\}$ has this relation. A repeated relation includes the ordered relation because a keyword pair of the repeated relation has the ordered relation and appears two or more times. A cyclic relation is classified as a strong repeated relation. This is described in more detail below.

Independent. The independent relation is where keywords a and b appear independently on time series pages. That is, they appear in random order with respect to each other; for example, such a keyword pair might be $\{pumpkin pie, recipes\}$. We define the independent relation as a residual which includes relations other than the three relation types described above.

Strict Co-occurring. The strict co-occurring relation is where keywords a and b both appear on the same page from among the time series pages, and it is strongly related to the co-occurring relation described above. For example, $\{Halloween, costume\}$ has such a relation.

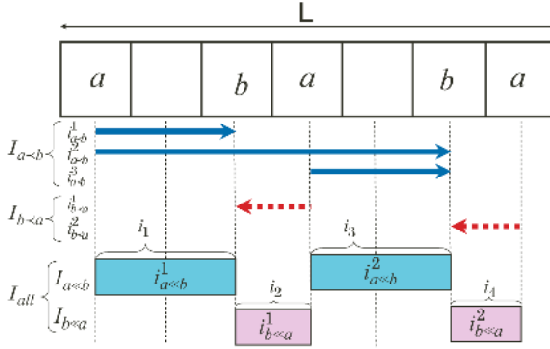


Fig. 2. Extraction of Intervals

Cyclic. The cyclic relation is where keywords a and b appear temporally and periodically, and it is strongly related to the repeated relation described above. This relation differs from the repeated one in that the keywords appear regularly. In particular, the time length of a to b is constant, as it is for b to a . That is, the time lengths of intervals a to b and b to a are approximate. For example, in the case of $\{cherry\ blossom, autumn\ leaves\}$, the time interval of spring to autumn is the same every year, as it is for autumn to spring. The same would hold for events (e.g., $\{Halloween, Christmas\}$) which usually appear periodically.

2.2 Determination

In this study, we determine the temporal relation between the keyword pairs on each URL. We generate keyword pairs and operating intervals around a keyword pair. We then identify each pair as having either a co-occurring, ordered, or repeated relation. The determination is done in the order of co-occurring, ordered, and cyclic. The method of determination for a keyword pair a, b is described below.

First, we extract intervals $I_{a \prec b}$ and $I_{b \prec a}$ to determine the temporal relation between query keywords. $I_{a \prec b}$ is a set of $i_{a \prec b}$, and $i_{a \prec b}$ is an interval from "a page that includes a " to "a page that includes b ". Correspondingly, $I_{b \prec a}$ is a set of $i_{b \prec a}$, and $i_{b \prec a}$ is an interval from "a page that includes b " to "a page that includes a ". We then extract intervals $I_{a \ll b}$ and $I_{b \ll a}$. $I_{a \ll b}$ is a set of $i_{a \ll b}$, and $i_{a \ll b}$ is an interval satisfying a condition that a appears earlier than b . As before, $I_{b \ll a}$ is a set of $i_{b \ll a}$, and $i_{b \ll a}$ satisfies a condition that b appears earlier than a . The extraction process of $I_{a \ll b}$ is as follows (Fig. 2).

- **STEP 1** Extract $I_{a \prec b}$
- **STEP 2** Remove $i_{a \prec b}$ which can be regarded as having $i_{b \prec a}$

STEP 2 is necessary because $I_{a \ll b}$ is not extracted from the parts of intervals where $i_{b \prec a}$ are of $i_{a \prec b}$.

In addition, we determine the temporal relations through extracted intervals. The definition of Fig. 2 and the algorithm for determining the relations are explained in the following. We define the intervals of time-series pages as follows. L is the time length of time-series pages. $I_{a \prec b}$, $I_{b \prec a}$, $i_{a \prec b}$, $i_{b \prec a}$, $I_{a \ll b}$, $I_{b \ll a}$, $i_{a \ll b}$, and $i_{b \ll a}$ are as having been defined. The set of $I_{a \ll b}$ and $I_{b \ll a}$ is I_{all} , and $I_{all} = \{i_1, i_2, \dots, i_n\}$.

```

//Determination of Co-occurring Relation
if (  $\frac{cnt(less\_time(I_{a \prec b})) + cnt(less\_time(I_{b \prec a}))}{cnt(I_{a \prec b}) + cnt(I_{b \prec a})} \geq \alpha$  )
  then //Determination of Strict Co-occurring Relation
    if (  $\frac{cnt(time\_0(i_{a \prec b}))}{cnt(I_{a \prec b}) + cnt(I_{b \prec a})} \geq \alpha$  )
      then
        strict co-occurring
      else
        co-occurring
    fi
  else goto Determination of Ordered Relation
fi

```

Fig. 3. Determination of Co-occurring Relation

Co-occurring Relation. If the total of co-occurring keywords among all keywords that appear is large, we determine that the relation is co-occurring. We set a threshold value which can be regarded as the same time, and if the total within which both keywords appear is more than the threshold value, the relation is co-occurring. The algorithm is shown below (Fig. 3).

We define the function as follows. $time(i)$ returns the time length of i . $cnt(I)$ returns the total of intervals included in I . $less_time(I)$ returns intervals (included in I) whose time length is less than a threshold value. $time_0(i)$ returns the total of i whose time length is zero. α is a threshold value.

We use $i_{a \prec b}$ and $i_{b \prec a}$, which can be regarded as keyword-appearing intervals, to determine co-occurring. The appearing intervals of keyword pair a and b are calculated by $time(i_{a \prec b})$, and the total of co-occurring keywords is a summation of $i_{a \prec b}$ and $i_{b \prec a}$ where the value is less than the threshold. We extract the ratio of the total of co-occurring to the total of appearing, and a relation is regarded as co-occurring if the value is larger than threshold α . When a relation is found to be co-occurring, we determine whether it is strict co-occurring². If the ratio of total of $time(i_{a \prec b}) = 0$ to the total of appearing is larger than threshold α , the relation is strict co-occurring.

Ordered Relation. If the relation is not determined as co-occurring, ordered relation determination is then done. If the order of a keyword pair appearing on time series pages is unique, the relation is determined as ordered. The method to determine an ordered relation is shown in Fig. 4 and proceeds as follows.

² We consider this to mean that both keywords appear on the same pages.

```

//Determination of Ordered Relation
if
  (  $\frac{\sum_{i=1}^n(\text{time}(i_i))}{L} \geq \beta$  )
  then
    ordered  $a < b$ 
  else
    if (  $\frac{\sum_{i=1}^n(\text{time}(i_{b \ll a}))}{\sum_{i=1}^n(\text{time}(i_{a \ll b}))} \geq \gamma$  )
      then
        ordered  $b < a$ 
      else
        goto Determination of Repeated Relation
    fi
  fi
fi
else
  independent
fi

```

Fig. 4. Determination of Ordered Relation

First, we calculate the ratio I_{all} to L (time length of all time-series pages) to determine whether a temporal relation is based on the keyword pair over the whole interval on the time series pages. The keyword pair is considered to depend on the order if the intervals keeping the order occupy the whole interval. If the value is less than β (threshold), the relation is independent, and the process is finished. Otherwise, the process is continued.

We next calculate the ratio of the summation of $I_{a \ll b}$ to L and the ratio of the summation of $I_{b \ll a}$. If the bias of two values is larger than γ (threshold), the relation is regarded as ordered.

Repeated Relation. When the relation is not determined as ordered, determination of whether it is a repeated relation occurs. If a keyword pair appears alternately, the keywords pair is repeated. The method of determining a repeated relation is shown in Fig. 5 and described below.

$reverse(i_j, i_{j+1})$ returns "true" or "false". If $((i_j == i_{a \ll b}) \wedge (i_{j+1} == i_{b \ll a}))$ or $((i_j == i_{b \ll a}) \wedge (i_{j+1} == i_{a \ll b}))$, $reverse(i_j, i_{j+1})$ returns "true" or "false". Otherwise, it returns "false". $variance(x)$ returns the variance value of x . β, θ , and δ are threshold values.

We determine whether the intervals keeping order $i_{a \ll b}$ and $i_{b \ll a}$ appear alternately. If the words of a keyword pair appear alternately, the intervals are added together. If the ratio of the summation to the time length of I_{all} exceeds θ , the relation is repeated; if not, the relation is independent.

When a relation is found to be repeated, we then determine whether it is cyclic. In the cyclic case, the period is constant so we calculate the variance of i_i (regarded as the time cycle). If the variance values of $i_{a \ll b}$ and $i_{b \ll a}$ are both less than δ , the relation is cyclic³.

³ The variance is a measure of the spread of the data.

```

//Determination of Repeated Relation
foreach (ij)
  if (reverse(ij, ij+1) == true)
    s = Add(time(ij));
    j ++;
  fi
end
if (  $\frac{s}{\sum_{i=1}^n (time(i_i))} \geq \theta$  )
  then //Determination of Cyclic Relation
    if ((variance(ia<<b)) < δ) ∧ (variance(ib<<a)) < δ)
      then
        cyclic
      else
        repeated
    fi
  else independent
fi

```

Fig. 5. Determination of Repeated Relation

3 Query Generation Based on Temporal Relation

3.1 Basic Query Generation

The Web retrieval method we propose generates queries based on the temporal relations between query keywords. We extract semantic relations (i.e., topic dependence and modification relation) from the temporal relations. And then we relate semantic relations to logical expression and generate queries. The relation is shown in Table 1. In this section, we describe the query generation by AND binding and OR binding. The basic queries for a keyword pair a, b based on the temporal relations are shown in Table 2.

The query $Q_{a||b}$ means the generated queries are $(a \vee b)$ and $(a \wedge b)$ because the keywords are considered to have topic dependence or a modification relation. $Q_{a||b}$ means at least one word of the keyword pair must be found; because the keywords always co-occur, the search is not always required for both keywords. The keywords have strong topic dependence, but they do not have a modification relation.

$Q_{a<b}$ means the keyword pair is joined by AND and the second keyword b is joined by OR. The former is based on a view that the ordered relation includes semantic relations like a causal relationship. Alternatively, through the latter a user can use this query to retrieve Web pages which currently include only b but also included a in the past.

$Q_{a\odot b}$ means the generated queries are $(a \wedge b)$, (a) , and (b) . The first one is because the keywords are considered to have topic dependence along with the ordered relation. Another query is needed because the components of keyword pairs do not usually appear together on the same page. $Q_{a\ominus b}$ means the generated queries are (a) and (b) as in the case of two queries for a repeated relation.

Table 1. Relation of Logical Expression and Semantic Relation

Logical Expression	Semantic Relation
$R(a \wedge b)$	modification, commonness
$R(a \vee b)$	topic dependence
$R(a) \cup R(b)$	topic independence

Table 2. Queries Generated from Query Keywords

Temporal Relation	Result for Generated Query
$Q_{a \parallel b}$ (co-occurring)	$R(a \vee b) \cup R(a \wedge b)$
$Q_{a \parallel b}$ (strictly co-occurring)	$R(a \vee b)$
$Q_{a < b}$ (ordered)	$R(a \wedge b) \cup R(b)$
$Q_{a \odot b}$ (repeated)	$R(a \wedge b) \cup (R(a) \cup R(b))$
$Q_{a \otimes b}$ (cyclic)	$R(a) \cup R(b)$
$Q_{\{a,b\}}$ (independent)	$R(a \wedge b)$

However, it often happens that the keywords have an independent topic, and $(a \wedge b)$ is not generated.

$Q_{\{a,b\}}$ means the keyword pair is joined by AND. The keywords do not make any sense without co-occurrence because they share no connection. That is, the keywords have a modifier relation with each other.

3.2 Query Combination

We generate queries by combining the basic queries. When the total number of query keywords is more than three, queries are generated taking into consideration the relationships between the temporal relations.

Example 1. In the case of keywords a , b , and c , for the case discussed in the preceding section, $\{a, b\}$ is co-occurring, and $\{b, c\}$ and $\{c, a\}$ are independent. From the relationship between the temporal relations, we extract that c is independent and a pair of a and b is co-occurring. The result of a query combination of $R(Q_{a \parallel b})$, $R(Q_{\{b,c\}})$, and $R(Q_{\{c,a\}})$ is $R(c \wedge (a \vee b)) \cup R(c \wedge a \wedge b)$. For example, if a is *SARS*, b is *corona*, and c is *virus*, the result for the generated query is as follows:

$$R(virus \wedge (SARS \vee corona)) \cup R(virus \wedge SARS \wedge corona)$$

Example 2. If $\{a, b\}$ is repeated, $\{b, c\}$ and $\{c, a\}$ are independent. The result of a query combination of $R(Q_{a \odot b})$, $R(Q_{\{b,c\}})$, and $R(Q_{\{c,a\}})$ is $R(a \wedge b \wedge c) \cup (R(c \wedge a) \cup R(c \wedge b))$. For example, if a is *cherry blossoms*, b is *autumn leaves*, and c is *castle*, the result for the generated queries is as follows:

$$R(cherry\ blossoms \wedge autumn\ leaves \wedge castle) \cup \\ R(castle \wedge cherry\ blossoms) \cup R(castle \wedge autumn\ leaves)$$

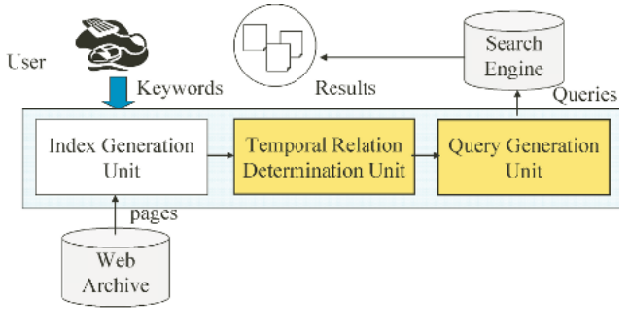


Fig. 6. System Architecture

4 Prototype System and Evaluation

4.1 Prototype System

Fig. 6 shows an overview of our prototype system. The system is organized into three units as follows:

1. Index Generation Unit

This unit collects a time series of Web pages and generates an index. The pages are collected by following links from the Internet Archive (http://web.archive.org/web/*/a targeted URL). The collection area is two links deep within the same site to take into consideration changing URLs.

The index generation unit indexes the words and records the times at which they were collected. The time dates are extracted from numerical values in the Internet Archive ([http://web.archive.org/web/14figures/targeted URL](http://web.archive.org/web/14figures/targetedURL)), and then are added to the indexes.

2. Temporal Relation Determination Unit

This unit extracts the time intervals by referring to the generated index and determines the temporal relation for each keyword pair. The targeted sites are the top N URLs that include all query keywords.

3. Query Generation Unit

This unit generates queries based on the temporal relation and performs retrieval using them. The targeted relations are the top M relations of the relations determined for N URLs.

4.2 Evaluation

We evaluated the extraction of the temporal relations and Web searching based on our method (Table 2). In this experiment, we manually picked out Web sites because at present we cannot generate a complete index of all Web pages stored in Web archives. We applied temporal relations that accounted for more than 30 percent of the determined relations to the query generation. In this regard, we assumed that $R(a \vee b)$ was equal to $R(a) \cup R(b)$ because the former

Table 3. Precision and Recall [Experiment 1]

	Query Result	Precision	Recall	F score
original	$R(SARS \wedge corona \wedge virus)$	0.95	0.35	0.51
proposed	$R(SARS \wedge corona \wedge virus) \cup R(virus \wedge SARS) \cup R(virus \wedge corona)$	0.90	0.84	0.87
other	$R(virus \wedge SARS) \cup R(virus \wedge corona)$	0.87	0.60	0.71
other	$R(virus \wedge corona) \cup R(SARS \wedge corona)$	0.92	0.63	0.75
other	$R(SARS \wedge corona) \cup R(virus \wedge SARS)$	0.90	0.63	0.74
other	$R(SARS \wedge corona)$	0.95	0.35	0.51
other	$R(virus \wedge corona)$	0.90	0.33	0.48
other	$R(virus \wedge SARS)$	0.85	0.31	0.54

was heavily dependent on the search engine. We evaluated the search results by comparing the precision, recall, and F score ($2pr/(p+r)$) of results when using generated queries to the results when using other combinations of query keywords ("original" is the current query). In addition, the total number of answer pages used to compute recall was the summation of answers obtained from queries generated for all combinations of keywords. We used Google as the search engine and examined the top 20 URLs.

Experiment 1. We first used $\{SARS, corona, virus\}$ as the query keywords and the URLs of websites concerning infectious diseases that included the keywords as the target Web pages. The keyword pair $\{SARS, corona\}$ had a co-occurring relation⁴. In contrast, the keywords $\{SARS, virus\}$ and $\{corona, virus\}$ were independent because these words are widely found on medical and public health sites.

As shown in Table 3, the F score of the proposed method was higher than that of any other query. We also found that two queries of the proposed method output specific Web pages. When one query, $(virus \wedge SARS)$, of this method was used, the answer set included news announcing general contents. These were pages on which gSARS virush was written rather than gcorona virush. When another query, $(virus \wedge corona)$, of this method was used, the answer set included information about the corona virus of pets (dogs and cats). At the same time, ten out of twenty Web pages included both $(SARS \wedge corona \wedge virus)$ and $(SARS \wedge corona)$. Hence, proposed query — i.e., $(SARS \wedge corona \wedge virus)$ — to return correct Web pages and $(virus \wedge SARS)$ and $(virus \wedge corona)$ to return specific Web pages were appropriate queries.

Experiment 2. Next, we used $\{cherry\ blossoms, autumn\ leaves, castle\}$ as query keywords and the URLs of pages including these keywords as the target Web pages. The keyword pair $\{cherry\ blossoms, autumn\ leaves\}$ had a repeated relation. We had expected that the relation would be a cyclic relation, but it depended on Web page conditions in the Web archive. In contrast, the keyword pairs $\{cherry\ blossoms, castle\}$ and $\{autumn\ leaves, castle\}$ were independent.

⁴ Note that this experiment was run when SARS first appeared and that SARS was later found to be caused by a corona virus.

Table 4. Precision and Recall [Experiment 2]

	Query Result	Precision	Recall	F score
original	$R(\text{cherry blossoms} \wedge \text{autumn leaves} \wedge \text{castle})$	0.75	0.31	0.44
proposed	$R(\text{cherry blossoms} \wedge \text{autumn leaves} \wedge \text{castle}) \cup$ $R(\text{castle} \wedge \text{cherry blossoms}) \cup R(\text{castle} \wedge \text{autumn leaves})$	0.78	0.90	0.83
other	$R(\text{castle} \wedge \text{cherry blossoms}) \cup R(\text{castle} \wedge \text{autumn leaves})$	0.80	0.67	0.73
other	$R(\text{castle} \wedge \text{autumn leaves}) \cup R(\text{cherry blossoms} \wedge \text{autumn leaves})$	0.57	0.56	0.56
other	$R(\text{cherry blossoms} \wedge \text{autumn leaves}) \cup R(\text{castle} \wedge \text{cherry blossoms})$	0.69	0.43	0.53
other	$R(\text{castle} \wedge \text{cherry blossoms})$	0.90	0.38	0.53
other	$R(\text{castle} \wedge \text{autumn leaves})$	0.70	0.29	0.41
other	$R(\text{cherry blossoms} \wedge \text{autumn leaves})$	0.50	0.21	0.29

As shown in Table 4, the F score of the proposed method was higher than that of any other query. We also found that two queries ($\text{castle} \wedge \text{cherry blossoms}$) and ($\text{castle} \wedge \text{autumn leaves}$) of the proposed method output specific Web pages.

When part of proposed query ($\text{castle} \wedge \text{cherry blossoms}$) of this method was used, the answer set included information about castles with a view of cherry blossoms. When another part of proposed query ($\text{castle} \wedge \text{autumn leaves}$) of this method was used, the answer set included information about castles with a view of autumn leaves. The query ($\text{cherry blossoms} \wedge \text{autumn leaves}$), which used unrelated query keywords, returned specific Web pages that were not appropriate for the query; e.g., pages including information on the ancient Japanese story.

Hence, the proposed query ($\text{cherry blossoms} \wedge \text{autumn leaves} \wedge \text{castle}$) returned correct Web pages and ($\text{castle} \wedge \text{cherry blossoms}$) and ($\text{castle} \wedge \text{autumn leaves}$) returned specific Web pages that were appropriate for the query.

5 Conclusion and Future Work

The Web search method we propose improves the effective recall rate by using query generation based on the temporal relations between keywords. We have found that search results from this method differ depending on the combination of query keywords. Our future work will be aimed at finding ways to generate a greater variety of queries, improve the algorithm, support more complex temporal relations, and solve the identification problem that arises when time series pages are the same.

Acknowledgments. This research was supported in part by a Grant-in-Aid for Scientific Research (B)(2) 16300028 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. Jatowt, A., and Ishizuka, M.: Summarization of Dynamic Content in Web Collections. In Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy (2004) 245–254

2. Jatowt, A., and Ishizuka, M.: Temporal Web Page Summarization. In Proceedings of the 5th International Conference on Web Information Systems Engineering, Brisbane, Australia (2004) 303–312
3. Jatowt, A., and Ishizuka, M.: Web page summarization using dynamic content. In Proceedings of the 13th International World Wide Web Conference (poster) (2004) 344–345
4. Chien, S., and Immorlica, N.: Semantic Similarity Between Search Engine Queries Using Temporal Correlation. In Proceedings of the 14th International Conference on World Wide Web (WWW2005) (2005) 2–11
5. Jatowt, A., Kawai, Y., and Tanaka, K.: Temporal Ranking of Search Engine Results. In Proceedings of the Fifth International Conference on Web Information Systems Engineering (WISE2005) (2005) 43–52
6. Jatowt, A., Kawai, Y., and Tanaka, K.: Using Web Archive for Improving Search Engine Results. In Proceedings of the Eighth Asia Pacific Web Conference (AP-Web2006) (2006) 893–898
7. Jatowt, A., Khoo, KB, and Ishizuka, M.: Change Summarization in Web Collections. In Proceedings of the 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Ottawa, Canada (2004) 653?662
8. Kuwabara, A., and Tanaka, K. RelaxImage: A Cross-Media Meta-Search Engine for Searching Images from Web Based on Query Relaxation. In Proceedings of the 21st International Conference on Data Engineering (ICDE2005) (2005)
9. Pradhan, S., Tajima, K., and Tanaka, K.: A Query Model to Synthesize Answer Intervals from Indexed Video Units. IEEE Transactions on Knowledge and Data Engineering, Vol.13, No.5 (2001) 824–838
10. Vlachos, M., Meek, C., Vagena, Z., and Gunopulos, D.: Identifying Similarities, Periodicities and Bursts for Online Search Queries. In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD2004) (2004) 131–142
11. Clusty.com: <http://clusty.com/>