DBSJ Journal, Vol.7, No.3
December 2008

Querying Topic Evolution in Time Series Document Clusters

Sophoin KHY[♡] **Yoshiharu ISHIKAWA**[♢]
Hiroyuki KITAGAWA[♠]

A document clustering method for time series documents produces a sequence of clustering results over time. Analyzing the contents and trends in a long sequence of clustering results is a hard and tedious task since there are too many number of clusters. In this paper, we propose a framework to find clusters of users' topics of interest and evolution patterns called *transition patterns* involving the topics. A cluster in a clustering result may continue to appear in or move to another cluster, branch into more than one cluster, merge with other clusters to form one cluster, or disappear in the adjacent clustering result. This research aims at providing users facilities to retrieve specific transition patterns in the clustering results. For this purpose, we propose a query language for time series document clustering results and an approach to query processing. The first experimental results on TDT2 corpus clustering results are presented.

1 Introduction

As information is more pervasive and overwhelming, organizing and extracting salient information becomes a challenging task. Document clustering is a method used to find and group documents such that documents in the same clusters are similar with each other and are dissimilar from other documents in different clusters. It has been used as a fundamental and pre-processing method in many areas, such as information retrieval [3], topic detection and tracking [2], text classification [9] and summarization of documents [8].

A document clustering method for time series documents such as the one in reference [5] produces a sequence of snapshots of clustering results over time (Fig. 1). Tracking topic evolution in a large stream of clusters is very important in the analysis of the characteristics of the data and the real world events. Analyzing the contents and trends in a long sequence of clustering snapshots, however, is a hard and tedious task since there are too many number of clusters; a user may need to access every cluster and read every document contained in each cluster. Some recent works have embraced tracing and analyzing changes in clusters over time [6, 7, 10]. Some further developped visualizing tools for browsing clusters and exploring trends in time series clustering results [1, 4]. In these works, without browsing, it is still difficult to obtain information of user interest.

Consider, for example, a sequence of clustering results as shown in Fig. 1. A user is interested in the US Democrat presidential nomination campaign. The topic can be represented, for example, by the keywords {clinton, obama, campaign}. The user wants to find occurrences of clusters, as shown in Fig. 1, highly related to the keywords. As another example, a user wants to know whether the topic {Clinton, campaignn} dissolves and changes to {Obama, campaign}? Or does {Clinton, campaign} merge with {Obama campaign} and form a broader topic {Democrat, campaign}? To the best of our knowledge, none of existing methods support such user requirements. By manually exploring the data, it is not easy for the user to judge whether those likely clusters correctly respond to his/her interest.

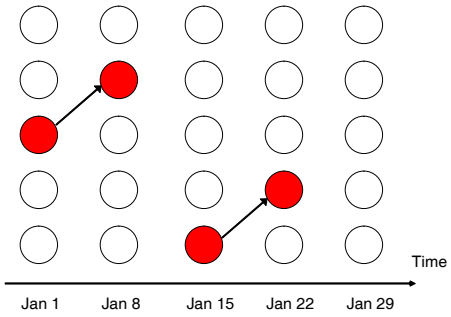

Figure 1: A Sequence of Clustering Results

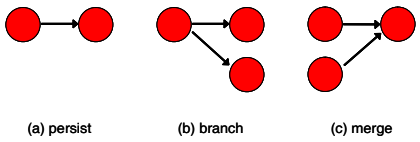

Figure 2: Transition Patterns

Further, in a sequence of clustering results of time series documents, a cluster in a clustering result may persist in another cluster, branch into more than one cluster, merge with other clusters to form one cluster, or disappear in the adjacent clustering result. These phenomena are called *cluster transitions* in reference [10] and we call the occurrence patterns of one or more such phenomena together, as shown in Fig. 2, *transition patterns*. In this paper, we propose a framework to find clusters related to users' topics of interest and transition patterns involving the topics. Specifically, we aim at providing users facilities to retrieve specific transition patterns in the clustering results. For this purpose, we propose a query language for time series document clustering results and an approach to query processing. Given a query of a transition pattern, it finds occurrences of cluster transitions that match the given pattern and are relevant to the query. The results are ranked and returned to the users.

♡ Student Member. Graduate School of System and Information Engineering, University of Tsukuba.
sophoin@kde.cs.tsukuba.ac.jp

♢ Regular Member. Information Technology Center, Nagoya University. ishikawa@itc.nagoya-u.ac.jp

♠ Regular Member. Graduate School of System and Information Engineering / Center for Computational Science, University of Tsukuba. kitagawa@cs.tsukuba.ac.jp

The remainder of this paper is organized as follows. Section 2 reviews related work. The preparations in this work are described in Section 3. Section 4 introduces the query language and explains the query processing scheme. The evaluation of the query language is shown in Section 5. Section 6 concludes the paper and discusses future work.

2 Related Work

There are several researches on identifying relationship between clusters. Mei et al. proposed an approach to discovering the evolutionary patterns of themes in a text stream [6]. In the approach, themes are generated using a probabilistic mixture model and theme evolutionary relations are discovered. Nallapati et al., based on the notion of events and topics in TDT [2], identified events that make up a topic and established dependencies among them [7]. Several dependency models have been proposed in the approach. Though the underlying problem in finding transitions between clusters is relevant to our work, the two approaches do not restrict to finding transitions between two consecutive time points, but finding all possible transitions between clusters from all time instances. These approaches aim at finding all relevant topics. Hence, their underlying objectives differ from our approach.

Spiliopoulou et al. proposed an approach called MONIC to model and track cluster transitions on clustering results at consecutive time points [10]. A cluster transition at a given time point is a change experienced by a cluster that has been discovered at an earlier time point. A transition may concern the content and form of the cluster, which is internal to it, or it may concern its relationship to the rest of the clustering, which is an external transition. MONIC detects both internal and external transitions of clusters and models their lifetime. The cluster transition graph (Section 3.2) in our approach is based on the idea of external transitions in MONIC.

T-Scroll [4] is an information visualization interface tool to visualize the overall trend of time series documents. It displays links between clusters, which represents related clusters from two consecutive time points, and allows users to explore more detailed information such as keyword lists and contents of documents in a cluster. In addition, Adomavicius et al. in reference [1] proposed a data analysis and visualization technique for representing trends in multiattribute temporal data called C-TREND, a system for temporal cluster construct. The idea in T-Scroll and C-TREND is partly related to our work. However, in our work, we proposed a query language for temporal clusters aiming at providing users facilities to search for specific patterns in the clustering results.

3 Preparations

3.1 Target Data

The target data in this work is a collection of accumulated document clustering results $D_1, \ldots, D_n$ at consecutive time points $T_1, \ldots, T_n$ generated by a clustering method on time series documents where a sliding window is adopted.

Let $S_1, \ldots, S_n$ be document sets in which S_i is the document set from which the document clustering result D_i is generated. S_i is assumed to be overlapped with S_{i+1}, i.e., $S_i \cap S_{i+1} \neq \emptyset$. All clusters in each $D_i (1 \leq i \leq n)$ are assumed non-overlapped, i.e., $\forall C_p, C_q \in D_i, C_p \cap C_q = \emptyset$, if $p \neq q$.

```
Query ::= Find_Clause From_Clause With_Clause
Find_Clause ::= 'find' Transition_Pattern
From_Clause ::= 'from' G
With_Clause ::= 'with' Bindings
Transition_Pattern ::= Snapshot_Spec
                       (-> Snapshot_Spec)*
Snapshot_Spec ::= Node | '{' Node_List '}'
Node_List ::= Node (, Node)*
G ::= Seq_Name('['TimeStamp, TimeStamp']')?
Bindings ::= Node '=' Keyword_List (, 'and'
             Node '=' Keyword_List)*
Keyword_List ::= '{'Keyword (, Keyword)*
                 ((, 'not('Keyword')')*)?'}'
```

Figure 3: The BNF of the Query Language

3.2 Constructing Cluster Transition Graph

A cluster transition, proposed in reference [10], at a given time point is a change experienced by a cluster that has been discovered at an earlier time point. It provides insights about the nature of cluster changes: is a cluster a newly emerged cluster or a disappeared one or does some of its member move to different clusters. In this work, we construct the cluster transition graph based on the idea of cluster transitions in reference [10].

A cluster transition graph is constructed by connecting pairs of adjacent clustering results. A pair of two adjacent clustering results D_i at time point T_i and D_j at the successive time point T_j is connected by detecting transitions between all clusters in D_i and D_j as follows.

For each cluster C_i in D_i and C_j in D_j, the degree to which C_i overlaps C_j is measured. The following function, the transition probability of the number of documents that were in C_i and moved to C_j, is used in this work.

$$\Pr(C_j|C_i) \stackrel{\text{def}}{=} \frac{|C_i \cap C_j|}{|C_i|}, \quad (1)$$

where $|C_i|$ and $|C_j|$ are the number of documents in C_i and C_j, respectively.

A list of clusters in D_j, which are matched clusters of clusters in D_i, and the corresponding transition probability values are obtained. Then, links between clusters and, hence, graphs can be constructed.

4 The Query Language

4.1 The Query Language Syntax

The query language proposed in this paper is a quite simple declarative language of small number of constructs, but can represent information needed for analyzing document clustering snapshots and detect transition patterns. Figure 3 shows the extended BNF notation of the query language.

4.2 Examples of Queries

The following examples show the query language syntax.

Query 1
```
find C -> C
from S08-7days[2008-Jan-01, 2008-Feb-28]
with C = {clinton, obama, campaign}
```

In the examples in this section, we assume `S08-7days` is the name of a sequence of document clusters and suppose it contains a sequence of document clusters from January 1, 2008 to present, and the clustering was performed based on seven-day incremental basis. The notation `S08-7days[2008-Jan-01, 2008-Feb-28]` specifies the restriction on the scope of the query target to the period between January 1, 2008 and February 28, 2008 within `S08-7days`.

In the above query example, the *transition pattern* `C -> C` in the `find` clause specifies the occurrences of the cluster transition to be found. The `with` clause `with C = {clinton, obama, campaign}` specifies the associate keywords for cluster `C`; it means we want to find occurrences of the cluster transition `C -> C` in which both clusters can be represented by the keywords {clinton, obama, campaign}.

The results of the query are given, for example, as follows.

```
1: C#Jan15-5 -> C#Jan22-4
2: C#Jan1-3  -> C#Jan8-2
...
```

It is a ranked list of cluster transitions corresponding to the query. The user may be able to specify the preferred number of entries in the list. The notation such as `C#Jan15-5` denotes the identity of a cluster. In this case, it represents the fifth cluster of January 15 clustering snapshot.

The next example shows a query to retrieve the repeated occurrences of a transition with length two.

Query 2
```
find C -> C -> C
from S08-7days[2008-Jan-01, 2008-Feb-28]
with C = {clinton, obama, campaign}
```

The following example shows a query using different keyword sets for two clusters.

Query 3
```
find C1 -> C2
from S08-7days[2008-Jan-01, 2008-Feb-28]
with C1 = {clinton, campaign, not(obama)},
 and C2 = {obama, campaign, not(clinton)}
```

This query retrieves cluster transitions, where the first cluster corresponds to {clinton, campaign} but does not include {obama}, and the second one corresponds to {obama, campaign} but does not include {clinton}.

The following query contains a transition pattern with a branch.

Query 4
```
find C1 -> {C2, C3}
from S08-7days[2008-Jan-01, 2008-Feb-28]
with C1 = {clinton, obama, campaign}, and
     C2 = {clinton, not(obama)}, and
     C3 = {obama, not(clinton)}
```

The query wants to find the occurrences of two transitions $C_1 \rightarrow C_2$ and $C_1 \rightarrow C_3$, where C_2 and C_3 branch from C_1 and are highly related to the given keywords.

The next one is the opposite; it contains a transition pattern of merge.

Query 5
```
find {C1, C2} -> C3
from S08-7days[2008-Jan-01, 2008-Feb-28]
with C1 = {clinton, not(obama)}, and
     C2 = {obama, not(clinton)}, and
     C3 = {clinton, obama, campaign}
```

4.3 Examples of Query Processing

In this section, the processing scheme of the query language regarding the query examples in Section 4.2 is given in the same order as follows.

4.3.1 Query 1: The Most Simple Case

We consider to answer Query 1. Let the set of keywords appearing in the `with` clause be $Q = \{t_1, \ldots, t_n\}$. In the example, Q is {clinton, obama, campaign}.

The `find` clause specifies that the two clusters, which have a transition, correspond to the same query Q, but are from two different clustering results at consecutive time points. To avoid confusion, the two clusters are denoted as C_1 and C_2, where C_1 temporally precedes C_2. We define the score for the cluster transition $C_1 \rightarrow C_2$ in terms of query Q by

$$\begin{aligned} & s(C_1(Q) \rightarrow C_2(Q)) \\ \stackrel{\text{def}}{=} \; & \Pr(C_1|Q) \cdot \Pr(C_2|Q) \cdot \Pr(C_1 \rightarrow C_2). \end{aligned} \tag{2}$$

Intuitively, the score gives a probability that the occurrence of a cluster transition is related to the given query Q. The score is calculated and used to rank the query results.

Next, we consider how to derive the probabilities $\Pr(C_1 \rightarrow C_2)$ and $\Pr(C|Q)$.

$\Pr(C_1 \rightarrow C_2)$ can be defined as

$$\Pr(C_1 \rightarrow C_2) \stackrel{\text{def}}{=} \Pr(C_2|C_1), \tag{3}$$

where $\Pr(C_2|C_1)$ is defined in Eq. 1 in Section 3.2.

$\Pr(C|Q)$ is a conditional probability that given the query Q, we obtain C as a relevant cluster to Q. It is defined as

$$\Pr(C|Q) \stackrel{\text{def}}{=} \prod_{i=1}^{n} \Pr(t_i \in C), \tag{4}$$

where $\Pr(t_i \in C)$ is the occurrence probability of a query term t_i of Q in cluster C. It is defined as follows.

$$\Pr(t_i \in C) \stackrel{\text{def}}{=} \frac{total\_freq(t_i)}{\sum_{j=1}^{l} total\_freq(t_j)}, \tag{5}$$

where $\{t_1, \ldots, t_l\}$ is the set of all terms appeared in the documents in cluster C and $total\_freq(t_i)$ is the total frequency of t_i in C and is defined as follows.

Suppose C contains documents $\{d_1, \ldots, d_{|C|}\}$ and let the frequency of term t_i in document d_k be $freq_k(t_i)$. The total frequency of t_i in C is defined by:

$$total\_freq(t_i) \stackrel{\text{def}}{=} \sum_{k=1}^{|C|} freq_k(t_i). \quad (6)$$

In summary, we get

$$\begin{aligned} & s(C_1(Q) \to C_2(Q)) \\ \propto & \left[\prod_{i=1}^{n} \Pr(t_i \in C_1)\right] \cdot \left[\prod_{i=1}^{n} \Pr(t_i \in C_2)\right] \cdot \frac{|C_1 \cap C_2|}{|C_1|} \\ = & \frac{|C_1 \cap C_2|}{|C_1|} \cdot \prod_{i=1}^{n} \left[\Pr(t_i \in C_1) \cdot \Pr(t_i \in C_2)\right]. \end{aligned} \quad (7)$$

4.3.2 Query 2: Long Sequence

Next, we consider Query 2. In this case, we assume the two transitions are nearly *independent* and approximate the probability.

$$\begin{aligned} & s(C_1(Q) \to C_2(Q) \to C_3(Q)) \\ \stackrel{\text{def}}{=} & s(C_1(Q) \to C_2(Q)) \times s(C_2(Q) \to C_3(Q)) \\ \stackrel{\text{def}}{=} & \Pr(C_1|Q) \cdot [\Pr(C_2|Q)]^2 \cdot \Pr(C_3|Q) \\ & \cdot \Pr(C_1 \to C_2) \cdot \Pr(C_2 \to C_3). \end{aligned} \quad (8)$$

The two probabilities can be calculated using the method of Query 1.

4.3.3 Query 3: Different Keyword Sets

We consider Query 3. In this case, we need to consider two keyword sets for C_1 and C_2. Let the corresponding keyword sets for C_1 be $Q_1 = \{t_1^1, \ldots, t_n^1\}$ and C_2 be $Q_2 = \{t_1^2, \ldots, t_m^2\}$. In the example, their contents are {clinton, campaign, not(obama)} and {obama, campaign, not(clinton)}, respectively.

In this query language, `not()` in the `with clause` is a predicate used to negate the effect of the query keyword in it. If a query contains the predicate is, for example, C = {clinton, campaign, not(obama)}, $\Pr(C|Q)$ is computed as follows.

$$\begin{aligned} \Pr(C|Q) &= \Pr(C = \{clinton \wedge campaign \wedge \neg obama\}) \\ &= \Pr(clinton \in C) \cdot \Pr(campaign \in C) \\ &\quad \cdot \Pr(obama \notin C) \\ &= \Pr(clinton \in C) \cdot \Pr(campaign \in C) \\ &\quad \cdot (1 - \Pr(obama \in C)). \end{aligned}$$

Finally, similar to Query1, we process this query as follows.

$$\begin{aligned} & s(C_1(Q_1) \to C_2(Q_2)) \\ \stackrel{\text{def}}{=} & \Pr(C_1|Q_1) \cdot \Pr(C_2|Q_2) \cdot \Pr(C_1 \to C_2). \end{aligned} \quad (9)$$

4.3.4 Query 4: Query with Branch

We consider Query 4. We assume the two cluster transitions are *independent*.

$$\begin{aligned} & s(C_1(Q_1) \to \{C_2(Q_2), C_3(Q_3)\}) \\ \stackrel{\text{def}}{=} & s(C_1(Q_1) \to C_2(Q_2)) \times s(C_1(Q_1) \to C_3(Q_3)). \end{aligned} \quad (10)$$

The results can be evaluated using the same approach as Query 1 and Query 3.

4.3.5 Query 5: Query with Merge

For Query 5, we use the same assumption as Query 4 that the two transitions are *independent*.

$$\begin{aligned} & s(\{C_1(Q_1), C_2(Q_2)\} \to C_3(Q_3)) \\ \stackrel{\text{def}}{=} & s(C_1(Q_1) \to C_3(Q_3)) \times s(C_2(Q_2) \to C_3(Q_3)). \end{aligned} \quad (11)$$

5 Evaluation

In this section, we evaluate the proposed query language and processing scheme described above.

For the implementation, $\Pr(t_i \in C)$ (Eq. 5) and $\Pr(C_j|C_i)$ (Eq. 1) are computed beforehand and stored persistently so that we can retrieve them when needed in the computation of query scores.

5.1 Experimental Setup

In this evaluation, the document clustering results of an extended-K-means-based incremental clustering method on TDT2 Corpus [11] in reference [5] is used as the data set. The TDT2 Corpus consists of chronologically ordered news stories obtained from various sources. The clustering results were generated by three-day incremental basis. We used 20 sets of clustering results dated from February 2 to March 31, 1998. Each clustering result contains 17 clusters and is associated with a timestamp, the date the clustering was performed.

Due to limitations in the data, some transition patterns and query keywords may not give meaningful results. Query keywords should be chosen such that the results are not empty. For this evaluation, we chose keywords which represent topics in the TDT2 Corpus. In addition, for query patterns of branch and merge, we use the same query keyword sets for all clusters C1, C2 and C3.

- `C -> C` pattern, {bomb, abortion, clinic}; {italy, cable, car, crash}; {pope, cuba}; {monica, clinton}; {tornado, florida}; {iraq, weapon, inspection}
- `C1 -> {C2, C3}` pattern, C1 = C2 = C3 = {monica, clinton}; {iraq, weapon, inspection}
- `{C1, C2} -> C3` pattern, C1 = C2 = C3 = {monica, clinton}; {iraq, weapon, inspection}

5.2 Results and Discussion

To evaluate the query results, topic labels for clusters of the clustering data are used as evaluation data. A topic label for a cluster was obtained by measuring precision of the cluster against the evaluation data of the TDT2 Corpus. If the precision score of the cluster for a topic is larger than a threshold, the cluster was labeled with the topic.

Table 1: Clusters and Labeled Topics

Cluster ID	Topic Name
Feb05-7, Feb08-8, Feb11-4, Feb14-4, Feb17-4, Feb17-4, Feb20-4, Feb23-4, Feb26-3, Mar01-3, Mar04-2, Mar07-2, Mar10-2, Mar13-4, Mar16-4, Mar19-3, Mar22-3, Mar25-3, Mar28-3, Mar31-3	Bombing AL Clinic
Feb08-3, Feb11-3, Feb14-3, Feb17-3, Feb17-3, Feb20-3, Feb23-3, Feb26-2, Mar01-2, Mar04-1, Mar07-1, Mar10-1, Mar13-3, Mar16-3, Mar19-2, Mar22-2, Mar25-2, Mar28-2, Mar31-2	Cable Car Crash
Feb02-6, Feb02-12, Feb05-11, Feb08-7, Feb08-12, Feb11-12, Feb14-12, Feb17-12, Feb20-12, Feb23-12, Feb26-12, Mar1-12, Mar4-12	Pope Visits Cuba
Feb02-3, Feb02-5, Feb05-3, Feb05-6, Feb08-6, Feb11-7, Feb14-7, Feb17-10, Feb20-10, Feb23-10, Feb26-9, Feb26-10, Mar01-10, Mar04-10, Mar07-9, Mar10-9, Mar13-11, Mar16-11, Mar19-9, Mar22-5, Mar22-9, Mar25-5, Mar25-10, Mar28-5, Mar28-9, Mar31-5, Mar31-10	Monica Lewinsky Case
Feb02-0, Feb05-0, Feb05-5, Feb08-0, Feb08-5, Feb11-0, Feb11-6, Feb14-0, Feb14-6, Feb17-0, Feb20-0, Feb23-0, Feb26-0, Mar01-0, Mar01-8, Mar01-14, Mar04-8, Mar04-14, Mar07-7, Mar07-14, Mar10-7, Mar10-14, Mar13-9, Mar16-10, Mar19-8, Mar19-14, Mar22-13, Mar25-8, Mar25-13, Mar28-7, Mar28-12, Mar31-8, Mar31-14	Current Conflict with Iraq

Table 1 shows some cluster ID's and the topic labels for the clusters of the clustering data when the threshold is set to 0.5. Ideally, clusters in Table 1 should appear in top ranked results of the queries for the topics.

Due to space constraint, we show only results of some queries. For the transition pattern `C -> C`, query {bomb, abortion, clinic} returns 36 instances as results; {italy, cable, car, crash} 39 instances; {pope, cuba} 21 instances; {monica, clinton} 53 instances; {tornado, florida} 24 instances; {iraq, weapon, inspection} 91 instances. Tables 2, 3, and 4 show top-10 results of queries {bomb, abortion, clinic}, {italy, cable, car, crash} and {pope, cuba}, respectively. For the transition pattern of branch `C1 -> {C2, C3}`, query {monica, clinton} returns 20 instances as results; {iraq, weapon, inspection} 30 instances. Table 5 shows top-10 results of query {iraq, weapon, inspection}. For the transition pattern of merge `{C1, C2} -> C3`, query {monica, clinton} returns 19 instances as results; {iraq, weapon, inspection} 52 instances. Table 6 shows top-10 results of query {monica, clinton}.

We compared the clusters in the query results with the evaluation data in Table 1. Low-ranked results generally have very small occurrence frequencies of the query keywords in the clusters and small transition probability values between the clusters. That is they are less similar to the topic of the query. The results in the middle ranks have either small occurrence frequencies of the query keywords and high transition probability values, or high occurrence frequencies of the query keywords and low transition probability values. The results in the top ranks in general have high occurrence frequencies of the query keywords and high transition probability values. Top-k results mostly contain transitions highly relevant to the topic of the query. Top k thus provide meaningful results.

Table 2: Top-10 Results of C → C Patterns of Query {bomb, abortion, clinic}

Query {bomb, abortion, clinic}	Score
Mar16-4 → Mar19-3	4.48E-10
Mar19-3 → Mar22-3	4.48E-10
Mar22-3 → Mar25-3	4.48E-10
Mar25-3 → Mar28-3	3.77E-10
Mar13-4 → Mar16-4	2.85E-10
Feb11-4 → Feb14-4	2.14E-10
Feb14-4 → Feb17-4	1.91E-10
Feb17-4 → Feb20-4	1.91E-10
Feb20-4 → Feb23-4	1.91E-10
Mar04-2 → Mar07-2	1.90E-10

Table 3: Top-10 Results of C → C Patterns of Query {italy, cable, car, crash}

Query {italy, cable, car, crash}	Score
Feb08-3 → Feb11-3	1.67E-16
Feb11-3 → Feb14-3	1.48E-16
Feb14-3 → Feb17-3	1.08E-16
Feb20-3 → Feb23-3	1.02E-16
Feb23-3 → Feb26-2	1.02E-16
Feb26-2 → Mar01-2	1.02E-16
Mar01-2 → Mar04-1	1.02E-16
Feb17-3 → Feb20-3	9.38E-17
Mar04-1 → Mar07-1	2.18E-17
Mar13-3 → Mar16-3	8.50E-18

6 Conclusions and Future Work

In this paper, we presented a novel declarative query language and its processing scheme to retrieve transition patterns in time series document clustering results. The query language is composed of simple and small constructs but can retrieve interesting and meaningful patterns. The experimental evaluation confirmed the effectivenss of the query processing scheme.

Depending on the applications and user requirements, other transition patterns may be interesting. The query language is not limited to only clustering data set. It can be applied to any sequences of a time series of grouped data set, which can be regarded as clusters in some sense, and where cluster transitions can be detected. Applications of the query language to other data set and further extension of the query language may present more interesting retrieval results and effectiveness of the language. In addition, a visualizing tool to highlight retrieval results in the sequence of periodical clustering results can facilitate users in analyzing and exploring the data.

[Acknowledgements]

This research is partly supported by the Grant-in-Aid for Scientific Research (19300027) from Japan Society for the Promotion of Science (JSPS) and the Grant-in-Aid for Scientific Research (19024006) from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

Table 4: Top-10 Results of C → C Patterns of Query {pope, cuba}

Query {pope, cuba}	Score
Feb02-12 → Feb05-11	1.73E-06
Feb05-11 → Feb08-12	1.70E-06
Feb11-12 → Feb14-12	1.69E-06
Feb08-12 → Feb11-12	1.60E-06
Feb14-12 → Feb17-12	9.59E-07
Feb17-12 → Feb20-12	4.54E-07
Mar01-12 → Mar04-12	4.40E-07
Feb26-12 → Mar01-12	4.03E-07
Feb20-12 → Feb23-12	2.10E-07
Feb23-12 → Feb26-12	1.36E-07

Table 5: Top-10 Results of Branch Patterns of Query {iraq, weapon, inspection}

Query {iraq, weapon, inspection}	Score
Mar25-13 → {Mar28-7, Mar28-12}	2.02E-21
Mar25-8 → {Mar28-7, Mar28-12}	9.30E-22
Feb05-0 → {Feb08-0, Feb08-5}	1.11E-22
Feb08-0 → {Feb11-0, Feb11-6}	1.22E-23
Mar04-8 → {Mar07-7, Mar07-14}	5.18E-24
Feb11-0 → {Feb14-6, Feb14-0}	3.26E-24
Feb11-6 → {Feb14-0, Feb14-6}	2.75E-24
Feb05-5 → {Feb08-0, Feb08-5}	1.90E-24
Mar07-7 → {Mar10-14, Mar10-7}	9.16E-25
Mar01-8 → {Mar04-8, Mar04-14}	7.16E-25

Table 6: Top-10 Results of Merge Patterns of Query {monica, clinton}

Query {monica, clinton}	Score
{Mar28-5, Mar28-9} → Mar31-5	1.28E-15
{Mar22-5, Mar22-9} → Mar25-5	2.56E-16
{Feb05-3, Feb05-6} → Feb08-6	7.35E-17
{Feb02-3, Feb02-5} → Feb05-6	6.49E-17
{Mar25-10, Mar25-5} → Mar28-5	6.42E-17
{Mar22-5, Mar22-9} → Mar25-10	3.73E-17
{Feb26-10, Feb26-9} → Mar01-10	1.05E-18
{Feb02-3, Feb02-5} → Feb05-3	1.60E-19
{Mar22-13, Mar22-9} → Mar25-10	1.61E-20
{Mar22-13, Mar22-5} → Mar25-10	1.20E-21

[References]

[1] Adomavicius, G., Bockstedt, J.: C-TREND: Temporal Cluster Graphs for Identifying and Visualizing Trends in Multiattribute Transactional Data. IEEE Trans. on Know. and Data Eng. **20**(6), 721–735 (June 2008)

[2] Allan, J. (ed.): Topic Detection and Tracking: Event-based Information Organization. Kluwer, Boston (2002)

[3] Cutting, D., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In: Proc. of 15th ACM SIGIR conference, pp. 318–329 (1992)

[4] Ishikawa, Y., Hasegawa, M.: T-Scroll: Visualizing Trends in a Time-series of Documents for Interactive User Exploration. In: Proc. of the 11th ECDL Conference, pp. 235–246 (2007)

[5] Khy, S., Ishikawa, Y., Kitagawa, H.: A Novelty-based Clustering Method for On-line Documents. World Wide Web Journal **11**(1), 1–37 (March 2008)

[6] Mei, Q., Zhai, C.: Discovering Evolutionary Theme Patterns from Text - An Exploration of Temporal Text Mining. In: Proc. of ACM KDD Conference, pp. 198–207 (2005)

[7] Nallapati, R., Feng, A., Peng, F., Allan, J.: Event Threading within News Topic. In: Proc. of CIKM Conference, pp. 446–453 (2004)

[8] Radev, D., Otterbacher, J., Winkel, A., Blair-Goldensohn, S.: NewsInEssence, Summarizing Online News Topics. Communications of the ACM, pp. 95–98 (2005)

[9] Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, **34**(1), pp. 1–47 (2002)

[10] Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y., Schult, R.: MONIC – Modeling and Monitoring Cluster Transitions. In: Proc. of KDD Conference, pp. 706–711 (2006)

[11] Linguistic Data Consortium (LDC), http://www.ldc.upenn.edu/

Sophoin KHY
Sophoin Khy is a Ph.D student of Graduate School of Systems and Information Engineering, University of Tsukuba. She received the M.Eng. degree from Master's Program in Sciences and Engineering, University of Tsukuba, in 2006. Her research interests include text and web mining and databases. She is a student member of ACM SIGMOD Japan and the Database Society of Japan.

Yoshiharu ISHIKAWA
Yoshiharu Ishikawa is a Professor at Information Technology Center, Nagoya University. His research interests include spatio-temporal and document databases, data and web mining, digital libraries, information retrieval and Web information systems. He is a member of the Database Society of Japan, IPSJ, IEICE, JSAI, ACM and IEEE Computer Society.

Hiroyuki KITAGAWA
Hiroyuki Kitagawa is a Professor at Graduate School of Systems and Information Engineering, University of Tsukuba. His research interests include integration of heterogeneous information sources, sensor databases and streams, data mining, WWW and databases, XML and semi-structured data, multimedia databases, and human interface. He is a member of ACM, IEEE Computer Society, the Database Society of Japan, IEICE, IPSJ, and JSSST. He is now an IEICE Fellow, an IPSJ Fellow, and a Trustee of the Database Society of Japan.