A Query Language and Its Processing for Time-Series Document Clusters

Sophoin Khy[†] Yoshiharu Ishikawa[‡] Hiroyuki Kitagawa[†,††]

[†] Graduate School of Systems and Information Engineering, University of Tsukuba
[‡] Information Technology Center, Nagoya University
[††] Center for Computation Sciences, University of Tsukuba
sophoin@kde.cs.tsukuba.ac.jp, ishikawa@itc.nagoya-u.ac.jp
kitagawa@cs.tsukuba.ac.jp

Abstract. Document clustering methods for time-series documents produce a sequence of snapshots of clustering results over time. Analyzing the contents (topics) and trends in a long sequence of clustering snapshots is hard and requires efforts since there are too many number of clusters; a user may need to access every cluster or read every document contained in each cluster. In this paper, we propose a framework to find clusters of user interest and change patterns called *transition patterns* involving the clusters. 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. 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.

Key words: cluster graph, cluster transition, clustering result, graph query, query language, query processing, transition pattern

1 Introduction

The Internet has become a common tool for information dissemination, where on-line information, such as news and blogs, is being profusely published everyday. Organizing or extracting salient information from such profuse amount of information is thus a challenging task. Document clustering is a method used to find and group similar documents into the same clusters, while dissimilar ones into 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].

Document clustering methods for time-series documents such as the one in reference [5] produce a sequence of snapshots of clustering results over time (Fig. 1). Analyzing the contents (topics) and trends in a long sequence of clustering snapshots is hard and requires efforts since there are too many number of clusters; a user may need to access every cluster or 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 specific 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, nomination}. The user wants to find occurrences of clusters, as shown in Fig. 1, highly related to the keywords. To the best of our knowledge, none of existing methods support such user requirements. By manually exploring the data in the system, it is not easy for the user to judge whether those likely clusters respond well to his/her interest. In addition, tracking topic evolutions in a large stream of clusters is very important in the analysis of the characteristics of the data or the real world events. For example, does the topic {Clinton nomination campaign} dissolve and change to {Obama nomination campaign}? Or does {Clinton nomination campaign} merge with {Obama nomination campaign} and form {Democrat US presidential campaign}?

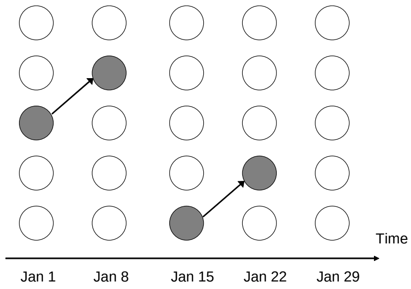

Fig. 1. A Sequence of Clustering Results

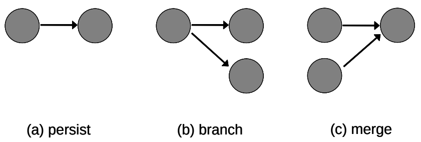

Fig. 2. Transition Patterns

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 is 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 clusters. 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 and transition pattern, it finds the occurrences of cluster transitions that match the given pattern and are relevant to the query. The results are ranked and returned to the users.

The remainder of this paper is organized as follows. Section 2 reviews related work. The preparations in this research 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. Although 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 find 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. 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 $p \neq q$.

3.2 Constructing Cluster Transition Graph

A cluster transition, proposed in MONIC [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 MONIC.

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

```
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’)’)*)?’}’
```

Fig. 3. BNF of Query Language

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.

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

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 cluster transition patterns. Figure 3 shows an 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, nomination}
```

We assume that `S08-7days` is the name of a sequence of document clusters. Suppose that it contains a sequence of document clusters from January 1, 2008 to present, and the clustering was performed on seven-day basis. The notation `S08-7days[2008-Jan-01, 2008-Feb-28]` represents that we restrict the scope of the 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 a transition pattern to be found. The `with` clause `with C = {clinton,` `obama, nomination`} represents the associate keywords for `C`; it means that we want to find the occurrences of a transition pattern in which both clusters can be represented by keywords {clinton, obama, nomination}.

The results of the query is 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. The user may be able to specify the preferred number of entries. The notation such as `C#Jan15-5` denotes the identity of a cluster. In this case, it represents the fifth cluster obtained at January 15.

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

Query 2

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

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

Query 3

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

This query retrieves cluster transitions in which the first cluster corresponds to {clinton, nomination} but does not include {obama}, and the second one corresponds to {obama, nomination} 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, nomination}, 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 query is the opposite; it is about the merge pattern.

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, nomination}
```

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.

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, nomination}.

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 $C_1 \to C_2$ in terms of query Q by

$$s(C_1(Q) \to C_2(Q)) \overset{\text{def}}{=} \Pr(C_1|Q) \cdot \Pr(C_2|Q) \cdot \Pr(C_1 \to C_2). \tag{2}$$

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 $\Pr(C_1 \to C_2)$ and $\Pr(C|Q)$.

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

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

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

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

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

In summary, we get

$$\begin{aligned} s(C_1(Q) \to C_2(Q)) &\propto \left[\prod_{i=1}^{n} \Pr(t_i \in C_1)\right] \cdot \left[\prod_{i=1}^{n} \Pr(t_i \in C_2)\right] \cdot \frac{|C_1 \cap C_2|}{|C_1|} \\ &= \frac{|C_1 \cap C_2|}{|C_1|} \cdot \prod_{i=1}^{n} \left[\Pr(t_i \in C_1) \cdot \Pr(t_i \in C_2)\right]. \end{aligned} \tag{5}$$

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

$$\begin{aligned} s(C_1(Q) \to C_2(Q) \to C_3(Q)) &\overset{\text{def}}{=} s(C_1(Q) \to C_2(Q)) \times s(C_2(Q) \to C_3(Q)) \\ &\overset{\text{def}}{=} \Pr(C_1|Q) \cdot [\Pr(C_2|Q)]^2 \cdot \Pr(C_3|Q) \\ &\quad \cdot \Pr(C_1 \to C_2) \cdot \Pr(C_2 \to C_3). \end{aligned} \tag{6}$$

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

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, nomination, not(obama)} and {obama, nomination, 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 the parenthesis. If a query contains the predicate is, for example, C = {clinton, nomination, not(obama)}, $\Pr(C|Q)$ is computed as follows.

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

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

$$s(C_1(Q_1) \to C_2(Q_2)) \stackrel{\text{def}}{=} \Pr(C_1|Q_1) \cdot \Pr(C_2|Q_2) \cdot \Pr(C_1 \to C_2). \tag{7}$$

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

$$\begin{aligned}&s(C_1(Q_1) \to \{C_2(Q_2), C_3(Q_3)\}) \\ &\stackrel{\text{def}}{=} s(C_1(Q_1) \to C_2(Q_2)) \times s(C_1(Q_1) \to C_3(Q_3))\end{aligned} \tag{8}$$

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

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

$$\begin{aligned}&s(\{C_1(Q_1), C_2(Q_2)\} \to C_3(Q_3)) \\ &\stackrel{\text{def}}{=} s(C_1(Q_1) \to C_3(Q_3)) \times s(C_2(Q_2) \to C_3(Q_3)).\end{aligned} \tag{9}$$

5 Implementation

In this section, we implemented the proposed query language described above.

5.1 Computing Term Frequency

For the implementation, the $\Pr(t_i \in C)$, where t_i is a term and C is a cluster, is defined as follows.

$$\Pr(t_i \in C) \stackrel{\text{def}}{=} \frac{total\_freq(t_i)}{\sum_{j=1}^{l} total\_freq(t_j)}, \tag{10}$$

where $\{t_1, \ldots, t_l\}$ is the set of all terms appeared in the documents in 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:

$$total\_freq(t_i) \stackrel{\text{def}}{=} \sum_{k=1}^{|C|} freq_k(t_i). \tag{11}$$

This $\Pr(t_i \in C)$ (Eq. 10) and the transition probability $\Pr(C_j|C_i)$ (Eq. 1) are computed beforehand and stored persistently so that we can retrieve them when needed in the computation of query scores.

Cluster ID	Topic Name
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
Feb02-6, 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
Feb2-0, Feb5-0, Feb5-5, Feb8-0, Feb8-5, Feb11-0, Feb11-6, Feb14-0, Feb14-6, Feb17-0, Feb20-0, Feb23-0, Feb26-0, Mar1-0, Mar1-8, Mar1-14, Mar4-8, Mar4-14, Mar7-7, Mar7-14, 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
Mar07-15, Mar10-15, Mar13-15, Mar16-15, Mar19-15, Mar22-15, Mar25-15, Mar28-15	Tornado in Florida

Table 1. Clusters and Labeled Topics

5.2 Experimental Evaluation

Experimental Setup In this evaluation, the time-series document clustering result of an extended-K-means-based incremental clustering method on TDT2 Corpus [11] in reference [5] is used as the data set. The clustering results were generated by three-day incremental basis. The TDT2 Corpus consists of chronologically ordered news stories obtained from various sources. We used 20 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 experiment, we chose keywords which represent topics in the TDT2 Corpus for transition patterns as given below. 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, C = {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}

Results and Discussion To evaluate the query results, topic labels for clusters in the clustering data are used to evaluate query results. A topic label of 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 is labeled with the topic. Table 1 shows some cluster ID's and topic labels for the clusters of the clustering data when the threshold is set to 0.51. Ideally, the clusters of a topic in Table 1 should appear in top-ranked results of the query for the topic.

Due to space constraint, we show only results of some queries. For the transition pattern `C -> C`, query {pope, cuba} returns 21 instances as results; {monica, clinton} 53 instances; {tornado, florida} 24 instances; {iraq, weapon,

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 2. Top-10 Results of C → C Pattern of Query {pope, cuba}

Query {tornado, florida}	Score
Mar19-15 → Mar22-15	8.06E-07
Mar25-15 → Mar28-15	6.72E-07
Mar22-15 → Mar25-15	5.52E-07
Mar16-15 → Mar19-15	1.34E-07
Mar13-15 → Mar16-15	5.89E-08
Mar10-15 → Mar13-15	4.90E-08
Mar07-15 → Mar10-15	4.88E-08
Mar04-16 → Mar07-15	8.12E-10
Mar01-16 → Mar04-16	7.68E-11
Feb26-16 → Mar01-16	2.38E-11

Table 3. Top-10 Results of C → C Pattern of Query {tornado, florida}

Query {monica, clinton}	Score
Mar28-9 → {Mar31-5, Mar31-10}	3.52E-16
Mar19-9 → {Mar22-9, Mar22-5}	2.87E-16
Mar22-5 → {Mar25-10, Mar25-5}	1.35E-16
Mar22-9 → {Mar25-5, Mar25-10}	7.04E-17
Mar25-10 → {Mar28-5, Mar28-9}	1.70E-17
Feb02-3 → {Feb05-6, Feb05-3}	6.28E-18
Feb02-5 → {Feb05-3, Feb05-6}	1.66E-18
Feb23-10 → {Feb26-9, Feb26-10}	5.78E-19
Mar19-9 → {Mar22-5, Mar22-13}	5.71E-20
Mar19-9 → {Mar22-13, Mar22-9}	4.24E-20

Table 4. Top-10 Results of Branch Pattern of Query {monica, clinton}

Query {iraq, weapon, inspection}	Score
{Mar25-13, Mar25-8} → Mar28-7	1.10E-20
{Mar25-13, Mar25-8} → Mar28-12	1.71E-22
{Feb14-0, Feb14-6} → Feb17-0	3.89E-23
{Feb11-0, Feb11-6} → Feb14-0	1.98E-23
{Feb05-0, Feb05-5} → Feb08-0	1.68E-23
{Feb05-0, Feb05-5} → Feb08-5	1.26E-23
{Feb08-0, Feb08-5} → Feb11-0	6.13E-24
{Mar10-14, Mar10-7} → Mar13-9	4.51E-24
{Mar07-14, Mar07-7} → Mar10-7	1.54E-24
{Feb08-0, Feb08-5} → Feb11-6	1.38E-24

Table 5. Top-10 Results of Merge Pattern of Query {iraq, weapon, inspection}

inspection} 91 instances. Tables 2 and 3 show top-10 results of queries {pope, cuba} and {tornado, florida}, respectively. For the transition pattern of branch `C1 -> {C2, C3}`, query {monica, clinton} returns 20 instances as results; {iraq, weapon, inspection} 30 instances. Table 4 shows top-10 results of query {monica, clinton}. For the transition pattern of merge `{C1, C2} -> C3`, query {monica, clinton} returns 19 instances as results; {iraq, weapon, inspection} 52 instances. Table 5 shows top-10 results of the query {iraq, weapon, inspection}.

Low-ranked results of a query 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. Observing top-10 results of the above queries by comparing the clusters in the transitions of the top-10 results with the topic-labeled clusters in Table 1 reveals that most of the clusters in the top-10 results appear in Table 1. This shows that they are highly related to the queries since the query keywords are topic-representative keywords. In general, top-ranked results of a query mostly contain transitions highly relevant to the topic of the query.

6 Conclusions and Future Work

In this paper, we presented a 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 is powerful in retrieving 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 performance and effectiveness of the language. In addition, a visualizing tool to highlight retrieval results in the sequence of periodical clustering results facilitates 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. In addition, this work is supported by the grants from Hoso Bunka Foundation.

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 Online 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/