International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:6, 2008

Discovery of Time Series Event Patterns based on Time Constraints from Textual Data

Shigeaki Sakurai, Ken Ueno, and Ryohei Orihara

***Abstract*—This paper proposes a method that discovers time series event patterns from textual data with time information. The patterns are composed of sequences of events and each event is extracted from the textual data, where an event is characteristic content included in the textual data such as a company name, an action, and an impression of a customer. The method introduces 7 types of time constraints based on the analysis of the textual data. The method also evaluates these constraints when the frequency of a time series event pattern is calculated. We can flexibly define the time constraints for interesting combinations of events and can discover valid time series event patterns which satisfy these conditions. The paper applies the method to daily business reports collected by a sales force automation system and verifies its effectiveness through numerical experiments.

***Keywords*—Text mining, Sequential mining, Time constraints, Daily business reports.

I. Introduction

Textual data with time information, such as daily or weekly business reports, evaluation information of products, and web log data, are stored on many local area networks or web sites. There is a need to analyze textual data with time information, because useful knowledge which supports decision making is buried in the textual data. The textual data has three features. The first one is text, the second one is time, and the third one is sequences of texts, where the sequences are decided by time information. In order to analyze the data, we may be able to use text mining techniques [2] [4] [5] [10] and sequential mining techniques [1] [8] [14]. However, these methods cannot sufficiently analyze the data, because these methods deal with only one of these features.

On the other hand, a paper [7] proposed a method that extracts frequent phrases from textual data, compares them with the changes in frequency of the phrases over multiple intervals, and discovers trends. Another paper [6] proposed a method that divides a sequence of numerical data into sub-sequences; the sub-sequences are called trends. This method relates each trend to past items of textual data. Yet another paper [13] proposed a method that classifies items of textual data into two classes based on time. This method evaluates relations between features and the time by using χ^2-test, where the features are composed of phrases with nouns and/or named entities. The textual data is characterized by features combined with the time. Furthermore a paper [11] proposed a method that discovers interesting sequential event patterns from sequential textual data, where the patterns are composed of events. The method extracts frequent patterns and extracts interesting patterns from the frequent patterns by a comparison with predefined combinations of events. The extracted patterns can predict future events. Even if these methods can analyze the textual data with time information to some extent, they deal with only parts of the features of textual data. It is necessary to develop a method that deals with all the features spontaneously in order to analyze the data in detail.

Thus, this paper proposes a new method that discovers time series event patterns from textual data with the time information. Here, the patterns are composed of events, which are extracted from the textual data. The method introduces 7 types of time constraints. The constraints allow an analyst to flexibly define time constraints for combinations of events. The method also evaluates the time constraints in the discovery process of the patterns. The analysts can discover valid patterns without discovering invalid patterns which have too long time intervals. With this method, we can easily discover interesting patterns and can use the patterns for decision making. This paper verifies the effectiveness of the method by applying it to daily business reports collected by a Sales Force Automation (SFA) system.

II. Mining Method

A. Textual Data with Time Information

This paper deals with textual data with time information. An item of the data is composed of 3 kinds of information: time, attributes, and text. Here, the time is a time stamp in the item, but some items may share the same time. The attributes give additional information to the item. Each attribute has finite attribute values. The text is composed of sentences written in a natural language. In the case of a daily business report as shown in Figure 1, “date” is the time when the report was written. “name”, “company”, and “product name” are the attributes, where “name” is the name of the salesperson who wrote the report, “company” is the name of the company that the salesperson visits, and “product name” is the name of the task that the salesperson performs for the company. “memo” is the text, where sentences related to “date”, “name”, “company”, and “product name” are described.

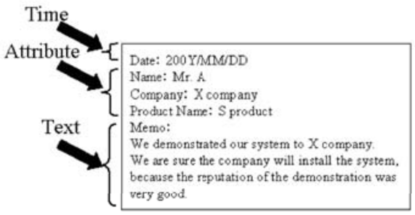
B. Time Series Event Pattern

The mining method discovers interesting patterns from textual data with time information. The patterns are composed

Shigeaki Sakurai is with System Engineering Laboratory, Corporate Research & Development Center, Toshiba Corporation, email: shigeaki.sakurai@toshiba.co.jp

Ken Ueno is with System Engineering Laboratory, Corporate Research & Development Center, Toshiba Corporation, email: ken.ueno@toshiba.co.jp

Ryohei Orihara is with System Engineering Laboratory, Corporate Research & Development Center, Toshiba Corporation, email: ryohei.orihara@toshiba.co.jp



Date: 200Y/MM/DD
Name: Mr. A
Company: X company
Product Name: S product
Memo:
We demonstrated our system to X company.
We are sure the company will install the system, because the reputation of the demonstration was very good.

Fig. 1. Textual data with time information

of interesting events. In the case of a daily business report, “inquiry regarding system specifications”, “demonstration of a system”, “strong interest in a system”, and “acceptance of an order” are interesting events. The method discovers a pattern as shown in Figure 2. This figure shows that left events occur earlier than right events. The pattern is composed of 3 elements, each element is composed of events with the same time, and the number of elements is called the size.

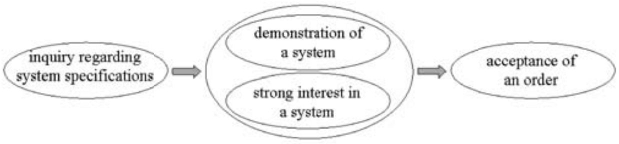

inquiry regarding system specifications
demonstration of a system
strong interest in a system
acceptance of an order

Fig. 2. Time series event pattern

C. Mining Flow

The mining method is composed of 4 processes: “Morphological Analysis”, “Extraction of Events”, “Generation of Time Series Event Data”, and “Discovery of Time Series Event Patterns” as shown in Figure 3. The first process decomposes each item of textual data into a set of words where parts of speech are assigned by using morphological analysis. The second process extracts a set of events corresponding to an item by referring to a key concept dictionary [4]. Here, the dictionary is created by human experts for a target task. The part of the dictionary is shown in the right upper side of Figure 3. The dictionary is composed of 3 layers: a concept class, a key concept, and an expression. The expression describes important words or important phrases in the items using regular expressions. The key concept is a set of expressions with the same meaning. The concept class is a set of relevant key concepts. Each key concept is regarded as an event. In Figure 3, a concept class “Business Action” includes a key concept “acceptance of an order” and the key concept includes 3 expressions “receive an order”, “a system is sold”, and “accept an order”. The third process creates groups by collecting items with the same attribute values. The process also arranges events extracted from items in each group in order of their time. The process can generate a sequence of time series event data from a group. The fourth process discovers interesting time series event patterns from the data by using our discovery method. The process is described in subsection II-E in detail.

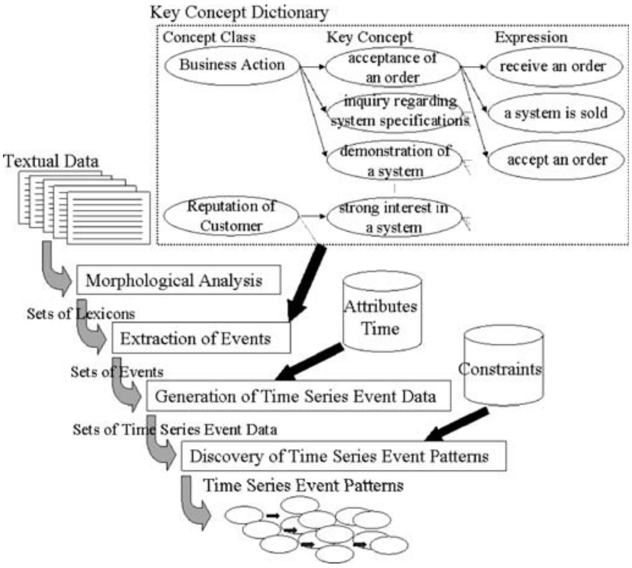


Fig. 3. Mining flow of time series event patterns

D. Time Constraint

Discovery method of sequential patterns, such as AprioriAll [1], PrefixSpan [8], and SPADE [14], efficiently discovers frequent sequential patterns. However, the discovered patterns are not always interesting to analysts, because the patterns are common and do not give new knowledge to the analysts. It is necessary to give some constraints and squeeze the frequent patterns in order to discover interesting patterns. It is in this context that the discovery methods introducing constraints have been studied. One paper [3] introduced constraints based on regular expressions. The constraints evaluate whether rows of events included in frequent sequential patterns are accepted or not by the regular expressions. Another paper [11] introduced constraints based on sub-patterns. The constraints evaluate whether frequent sequential patterns include one of given sub-patterns or not. A third paper [9] proposed a method that introduces constraints to projection-based discovery methods of sequential patterns. The paper also proposed an algorithm, called prefix-growth, that efficiently discovers sequential patterns which satisfy the constraints. These methods deal with the continuity of events, but these methods do not deal with the time stamp of events. These methods may extract patterns which include events with a long time interval. However, one event cannot influence another event owing to the long time interval. The events do not always have a relationship. Therefore, these methods may extract invalid patterns. It is necessary to pay attention to the time interval of events.

On the other hand, another paper [12] introduced a minimum gap (min-gap) and a maximum gap (max-gap) in order to evaluate time interval between events. The gaps deal with the time interval of neighboring events. However, the gaps cannot define flexible time constraints for combinations of events. It is necessary to introduce time constraints for various types of combinations of events, even if the events are not neighboring. Thus, the paper defines 7 types of new time constraints in order to flexibly introduce time constraints for combinations of events. The constraints are introduced based on an analysis of the time series event data for an SFA system and a medical examination. The time constraints are defined below.

(1) Time constraint between the first event and the last event: This constraint defines the minimum and the maximum time interval from the first event to the last event. We can evaluate the whole time interval of time series event patterns by using this constraint. The constraint is formally described as $MinTime \leq time(ev_L) - time(ev_F) \leq MaxTime$. Here, $MinTime$ is the minimum time interval of events, $MaxTime$ is the maximum time interval of events, ev_F is an event included in the first element of a sequential pattern, ev_L is an event included in its last event, and $time(ev)$ is a time stamp corresponding to an event ev. Figure 4 graphically shows the constraint. In this figure, each circle is an event.

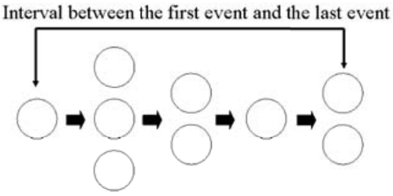


Fig. 4. Time constraint between the first event and the last event

(2) Time constraint between the first event and a specified event: This constraint defines the minimum and the maximum time interval from the first event to a specified event. We can evaluate the relative time from the first event if the specified event occurs in time series event patterns. The constraint is formally described as $MinTime \leq time(ev_S) - time(ev_F) \leq MaxTime$. Here, ev_S is a specified event. Figure 5 graphically shows the constraint. In this figure, a dotted circle is a specified event.

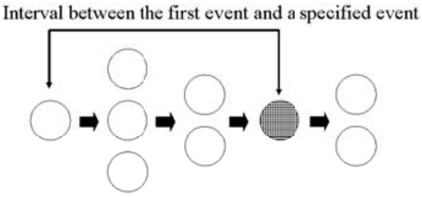


Fig. 5. Time constraint between the first event and a specified event

(3) Time constraint between a specified event and the last event: This constraint defines the minimum and the maximum time interval from a specified event to the last event. We can evaluate the relative time to the last event if the specified event occurs in time series event patterns. The constraint is formally described as $MinTime \leq time(ev_L) - time(ev_S) \leq MaxTime$. Figure 6 graphically shows the constraint.

(4) Time constraint between neighboring events: This constraint defines the minimum and the maximum time interval from an event and its prior neighboring event. The constraint is similar to the gaps proposed in the paper [12]. We can evaluate the relative time of all neighboring events in time series event patterns. The constraint is formally described as $MinTime \leq time(ev_i) - time(ev_{i+1}) \leq MaxTime$. Here, ev_i is an event included in i-th element of a sequential pattern. Figure 7 graphically shows the constraint.

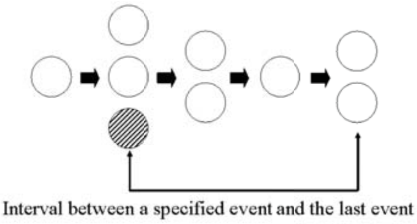


Fig. 6. Time constraint between a specified event and the last event

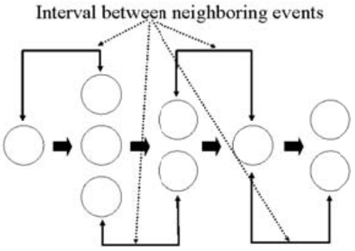


Fig. 7. Time constraint between neighboring events

(5) Time constraint between specified events: This constraint defines the minimum and the maximum time interval from a specified event to another specified event. The discovery method evaluates only the combination corresponding to the time order when there are multiple combinations of the specified events. For example, the 4-size pattern A → A → B → B is given, where both A and B are specified events. The method evaluates the relationship for both the first A and the first B, and the second A and the second B. However, the method does not evaluate relationships for both the first A and the second B, and the second A and the first B. The reason is that it is necessary to extract a pattern that includes cyclic relationships. These relationships sometimes occur in medical examination data. We can reflect cyclic relationships and evaluate the relative time for 2 types of specified events in time series event patterns. The constraint is formally described as $MinTime \leq time(ev_{S_1,i}) - time(ev_{S_2,i}) \leq MaxTime$. Here, $ev_{S_1,i}$ is i-th event of the specific event ev_{S_1}, $ev_{S_2,i}$ is i-th event of the specific event ev_{S_2}, and $time(ev_{S_1,i}) - time(ev_{S_2,i}) \geq 0$. Figure 8 graphically shows the constraint. In this figure, two types of dotted circles are two different specified events.

(6) Time constraint between a specified event and its prior event: This constraint defines the minimum and the maximum time interval from an event that occurs just before a specified event to the specified event. We can evaluate the relative time if an event occurs just before the specified event in time series event patterns. The constraint is formally described as $MinTime \leq time(ev_S) - time(bef(ev_S)) \leq MaxTime$.

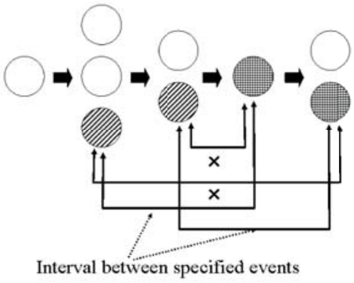


Fig. 8. Time constraint between specified events

Here, $bef(ev)$ is a prior event of an event ev. Figure 9 graphically shows the constraint.

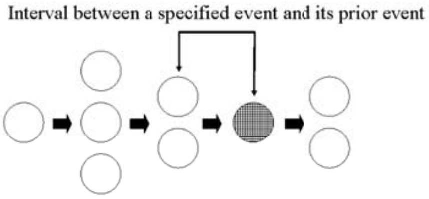


Fig. 9. Time constraint between a specified event and its prior event

(7) Time constraint between a specified event and its subsequent event: This constraint defines the minimum and the maximum time interval from when a specified event occurs to an event that occurs just after the specified event. We can evaluate the relative time if an event occurs just after the specified event in time series event patterns. The constraint is formally described as $MinTime \leq time(next(ev_S)) - time(ev_S) \leq MaxTime$. Here, $next(ev)$ is a subsequent event of an event ev. Figure 10 graphically shows the constraint.

These constraints are checked for all combinations of events of each candidate pattern in the discovery method of the patterns. That is, when a sequence of time series event data includes a candidate pattern, events corresponding to given time constraints are extracted from the pattern included in the sequences. The discovery method evaluates whether the time interval of the events satisfies the constraints or not. If all constraints are satisfied, the discovery method adds 1 to the pattern counter. All sequences are evaluated for the pattern. Thus, the method can discover frequent patterns whose time interval between the events is in the range of the user-defined minimum and maximum values.

We can flexibly define time constraints for various types of combinations of events by means of these time constraints.

E. Discovery of Patterns

The discovery method of time series event patterns is similar to AprioriAll algorithm. Firstly, the discovery method extracts event sets whose support values are equal to or larger than the minimum support defined by a user. The support value is calculated by dividing the number of sequences of time series event data into the number of sequences that include a specified pattern. The event sets are regarded as 1-size frequent time series event patterns. Next, the method generates a 2-size candidate pattern from two 1-size frequent time series event patterns. The method also calculates their support values with evaluating given time constraints. The candidate pattern is regarded as a 2-size frequent time series event pattern when a support value for the candidate pattern is equal to or larger than the minimum support. A set composed of 2-size frequent time series event patterns (L_2) is generated. The method outputs only the patterns that satisfy other constraint conditions from L_2. The output patterns are regarded as 2-size interesting time series event patterns. Here, one condition evaluates whether the patterns are included in other patterns of L_2 and the other conditions evaluate whether the patterns include predefined sub-patterns. The method repeatedly performs the process from the generation of the candidate patterns to the output of the interesting time series event patterns, until the method outputs all interesting time series event patterns. Figure 11 shows that a k-size candidate pattern is generated from two $(k-1)$-size frequent sequential patterns, where it is necessary that $(k-1)$ elements from the top correspond to each other. In the discovery method, we note that the method can generate k-size interesting patterns from two $(k-1)$-size patterns, where the $(k-1)$-size patterns may not be interesting but may be frequent. Therefore, it is necessary for the method to keep all $(k-1)$-size frequent patterns.

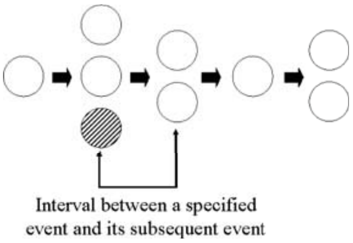


Fig. 10. Time constraint between a specified event and its subsequent event

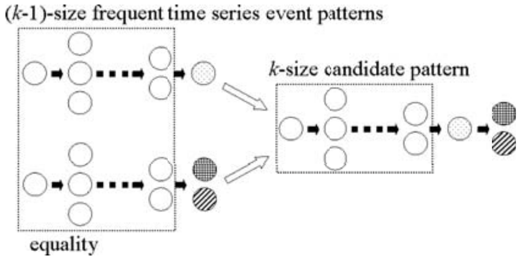


Fig. 11. Generation of a candidate pattern

Table I shows the pseudo coded algorithm of the discovery method. In this table, a deletion process is added before the main discovery process. The process deletes time series event data which does not include interesting sub-patterns given by analysts. Also, SeqDB is a set of time series event data, MinSup is the minimum support, TconsList is a set of time constraints, and CondList is a set of interesting sub-patterns. MinSup, CondList, and TconsList are defined by the analysts based on their interests. Moreover, checkFreqSeqTime() is a function that checks whether a time series event pattern is frequent under the time constraints TconsList, subseq() is a function that picks up a sub-pattern with the number of elements of an input value from the top of a time series event pattern, $\bowtie_{seq}$ is an operator that generates a k-size pattern from two (k-1)-size patterns, checkInclusion() is a function that checks whether a time series event pattern is included in other patterns with the same size, and checkInclusionCond() is a function that checks whether a time series event pattern includes sub-patterns defined by CondList. We note that a sub-pattern of a time series event pattern is not always satisfied with time constraints, even if the pattern is satisfied with the constraints. This is because the first event, the last event, and neighboring events in the sub-pattern are not always equal to the events in the pattern. In addition, the relationships between a specified event and other events are not always equal to the relationships in the pattern. Therefore, we have to recalculate the support value for candidate patterns by referring to SeqDB.

TABLE I
DISCOVERY METHOD OF TIME SERIES EVENT PATTERNS

```
MinSup = User-defined value;
//Delete of unnecessary time series event data
For each sequence es $\in$ SeqDB
  If !checkInclusionCond(es,CondList,1)
  Then delete es from SeqDB;
//Discovery of frequent elements
$L_1 = \phi$;
For each element el $\in$ es,es $\in$ SeqDB
  If checkFreqSeqTime(el,SeqDB,MinSup,TconsList,1)
  Then add el to $L_1$;
//Discovery of time series event patterns
$L_k = \phi$;
For($k$=2; $L_{k-1}$!=$\phi$; $k$++)
  //Generation of candidate patterns
  For each sequence es1 $\in L_{k-1}$
    For each sequence es2 $\in L_{k-1}$
      If subseq(es1,$k$-2)==subseq(es2,$k$-2)
      Then cs = es1 $\bowtie_{seq}$ es2;
           add cs to $L_k$;
  //Discovery of frequent patterns
  For each sequence cs $\in L_k$
    If checkFreqSeqTime(cs,SeqDB,MinSup,TconsList,$k$)
    Else delete cs from $L_k$;
  //Output of interesting patterns
  For each sequence cs $\in L_k$
    If !checkInclusion(cs,$L_k$)
      If checkInclusionCond(cs,CondList)
      Then output cs;
```

The discovery method can discover patterns composed of events with the valid time interval. Therefore, the discovery method can discover interesting patterns more easily than the previous discovery methods. In addition, it is not necessary for the discovery method to define time constraints for all combinations of events. We can efficiently use background knowledge of events and can define time constraints that are appropriate for the analysis task. Therefore, the discovery method can reduce omissions of discovery of interesting patterns.

III. NUMERICAL EXPERIMENTS

A. Experimental Data

We used 27,731 daily business reports in our experiments. The reports were collected by an SFA system that was introduced in 5 departments (A Dept. ∼ E Dept.) of our employer. We generated 6,434 sequences of time series event data from the reports. Table II shows the number of sequences, elements, events, and reports included in the data sets. Here, we used a key concept dictionary composed of 3 concept classes, 61 key concepts, and 835 expressions. The dictionary was created by a human expert for the task of analyzing daily business reports. Therefore, we used 61 kinds of events in this experiment. These events express negative or positive reputation for products, failure of an order, acceptance of an order, etc.

TABLE II
DATA FEATURES

Department	Sequence	Element	Event	Report
A Dept.	416	818	1,101	849
B Dept.	1,302	3,612	4,601	4,116
C Dept.	334	951	2,072	1,018
D Dept.	3,984	17,056	44,778	19,737
E Dept.	398	1,812	4,581	2,011

B. Experimental Method

In these experiments, we used 2.0%, 1.0%, 0.75%, and 0.5% as the minimum supports and used 23 sub-patterns as interesting sub-patterns. Here, the sub-patterns show that the event “acceptance of an order” occurs after events corresponding to the concept class “negative reputation” occur. The sub-patterns are important for the task of analyzing daily business reports, because analysts are interested in the reason for acceptance of an order even though the negative reputation of the product is sometimes pointed out. In addition to these constraints, we used the 6 time constraints shown in Table III. Each time constraint is described by the unit of day. The time constraints evaluate the whole time interval of patterns and the relative time for the combination of “negative reputation” and “acceptance of an order”. For example, t4 shows that the time interval from the start of business actions for a product to its end is within 180 days and we accept an order within 30 days from the customer after negative reputation is pointed out for the product. By using these time constraints, we can expect that the mining method will delete patterns that are not regarded as valid patterns over a long time period. Also, we can expect that the mining method will extract different trends based on the different time interval for the combination of “negative reputation” and “acceptance of an order”.

C. Experimental Results

Figure 12 ∼ 14 have 3 graphs that show the experimental results. Figure 12 shows the change in the number of patterns when the time constraints are relaxed. Figure 13 shows the change in the number of related texts when the time constraints are relaxed. On the other hand, Figure 14 shows the change

TABLE III
TIME CONSTRAINTS

ID	Time constraint
t1	First-Last∈[0,180]
t2	t1∩“negative reputation”.*-“acceptance of an order”∈[0,90]
t3	t1∩“negative reputation”.*-“acceptance of an order”∈[0,60]
t4	t1∩“negative reputation”.*-“acceptance of an order”∈[0,30]
t5	t1∩“negative reputation”.*-“acceptance of an order”∈[31,60]
t6	t1∩“negative reputation”.*-“acceptance of an order”∈[61,90]

in the number of patterns when “negative reputation” has a different time interval from “acceptance of an order”. In each graph, the x-axis shows the combination of the time constraints and the size of the patterns, and the y-axis shows the number of patterns or texts. “No” shows that there is no time constraint. The experimental results of the case “No” corresponds to the results of the previous method [11].

D. Discussion

(1) Validity of time constraints: The previous method [12] deals with time constraints of neighboring events by using 2 types of gaps. This method is unable to deal with the time constraints of events that are not continuous in a pattern. In the case of this analysis task, the event “acceptance of an order” does not always occur just after the events corresponding to “negative reputation”. Typically, some events are inserted between “acceptance of an order” and the events corresponding to “negative reputation”. In addition, the method evaluates all neighboring events with the same time constraint, even if different event combinations are evaluated. However, this task does not require evaluation of the time constraint for events other than “acceptance of an order” and “negative reputation”. The evaluation may lead to omission of interesting patterns. Therefore, if we use this method, we have to define more relaxed time constraints to avoid omissions. The method discovers many patterns including invalid patterns. A user requires more time to discover interesting patterns. On the other hand, the proposed method can flexibly introduce time constraints into combinations of events. The method can introduce time constraints to the only combination of “acceptance of an order” and “negative reputation”. It is possible to more strictly define the time constraints. The method will discover fewer patterns without omitting interesting patterns.

Figure 12 shows that the number of patterns decreases when stricter time constraints are introduced. The number of patterns in the case of t4 is about 6% when there is no time constraint. The proposed method succeeds in drastically decreasing the number of patterns. The number of related texts also decreases as shown in Figure 13. The number of patterns in the case of t4 is about 56% when there is no time constraint. The rate of decrease in the related texts is not as large as the rate of decrease in the patterns. This is because the method discovers patterns that include many sub-patterns and the discovered patterns are related to many texts.

On the other hand, 3 valid interesting patterns have been discovered for Dept A. with size 2 in the previous analysis [11]. The summary of the patterns is shown in the following.

- When there was a risk of an order going to another company due to inappropriate actions in regard to the customer, both a product and related documents were checked more strictly to clear our name. These actions prevented the loss of the order.
- When a customer commented that a product was expensive, measures were discussed among related departments and a discount of the product was decided. These actions led to the receiving of an order.
- When the term of guarantee for a product was about to expire, we talked with the customer about a new range of guarantee and clarified the range. These actions led to a new contract.

These patterns are also discovered in the case of t1 $\sim$ t4. These time constraints do not omit the valid interesting patterns. That is, the proposed method can make the task easier for the user without omitting valid interesting patterns.

We checked the difference in the trend by giving different time ranges. Each time constraint gives the maximum number for size 4 as shown in Figure 14. The trend is almost similar, even if the total number of patterns decreases as the ranges increase. In addition, we were unable to discover interesting patterns for t5 and t6. This is because “negative reputation” is only slightly related to “acceptance of an order” when the time interval is more than 30 days, and the relationship between “negative reputation” and “acceptance of an order” was discovered by the wrong extraction of events. Therefore, analysis based on different time ranges is not always important in this analysis task. However, there are tasks in which this analysis is required. For example, an event “Taking a drug” in an analysis task of a medical examination can relate to different events in different time ranges.

(2) Completeness of time constraints: This paper introduced the 7 types of time constraints based on the analysis of the data in an SFA system and a medical examination. For the SFA system data, each sequence of time series event data arrives at the event “acceptance of an order” or the event “failure of an order”. It is necessary to introduce a time constraint that evaluates from the top to theses events or from these events to the last in a candidate pattern. Also, it is important for salespersons to perform countermeasures for the events related to “negative reputation”. Timely and appropriate countermeasures lead to the event “acceptance of an order”. It is necessary to introduce a time constraint that evaluates the combination of a specified event and its neighboring events. Moreover, multiple sequences are sometimes grouped as a sequence. This is because some salespersons describe some products for the same company under the same product name. Thus grouping based on the company name and the product name generates an incorrect group of reports. It is necessary to introduce a time constraint that evaluates from the first event to the last event in order to deal with a group that includes multiple sequences. On the other hand, in the case of the medical examination data, combinations of events, such as the event “Taking a drug” and the event “Low blood pressure”, cyclically occur in sequences. A time constraint has to keep a cyclic relationship of the events. It is necessary to introduce a time constraint that pays attention to the cyclic correspondence

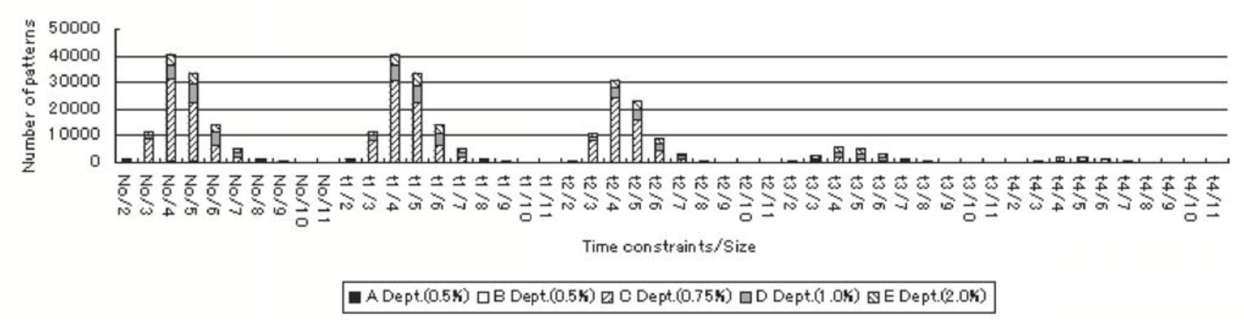


Fig. 12. Number of patterns for time constraints t1∼t4 and no time constraint

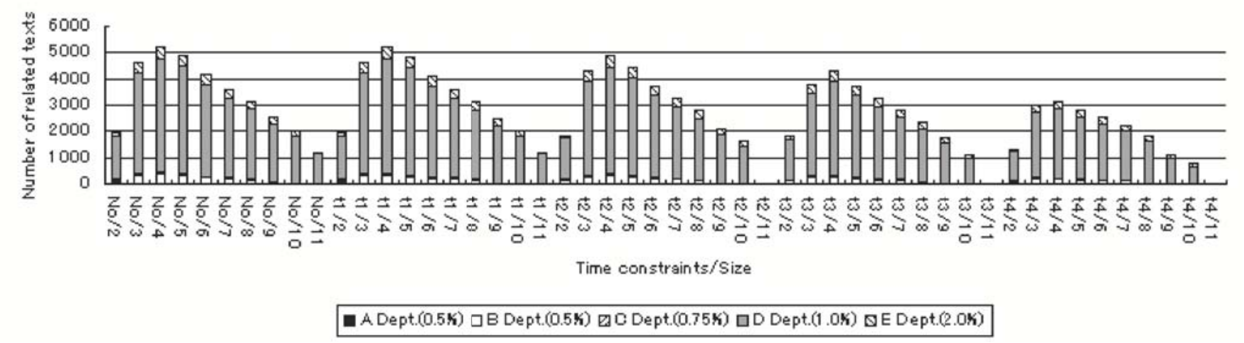


Fig. 13. Number of related texts for time constraints t1∼t4 and no time constraint

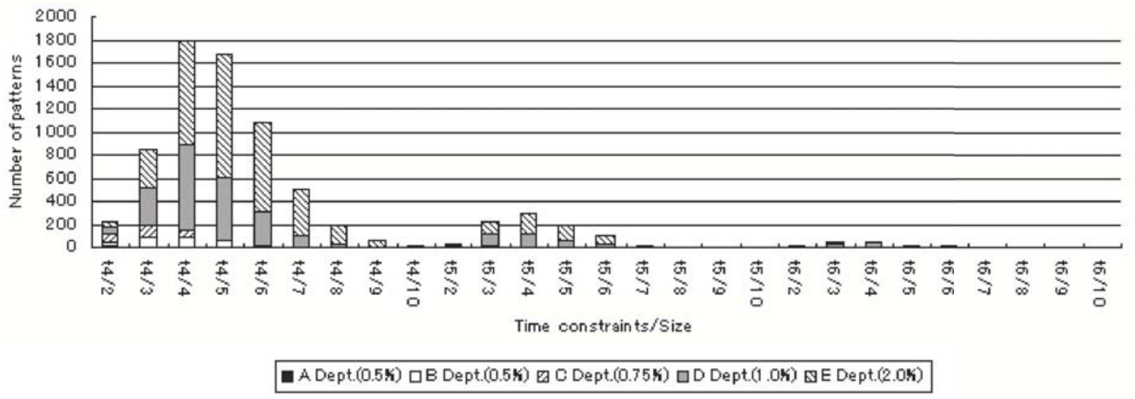


Fig. 14. Number of patterns for time constraints t4∼t6

of specified events and evaluates their combination.

Based on these discussions, we can describe sufficient time constraints for similar data sets by using the 7 types of time constraints. However, we are unable to claim that the proposed time constraints are complete for all data sets. In the future, it will be necessary to verify whether the constraints are complete through application to many tasks.

(3) A task of analyzing daily business reports: It is necessary for the proposed mining method to define events according to interests of analysts and to create a key concept dictionary in order to extract the events. It is not easy to create a dictionary. However, we can use a GUI-based support system to create one. We can interactively create the dictionary by using the system to refer to daily business reports. In addition, we can apply the dictionary to similar tasks of analyzing daily business reports and we have a large number of reports. On the other hand, it is also necessary for the method to define time constraints. In this experiment, we introduced a time constraint between the first event and the last event, because salespeople mostly finish actions for a product within 180 days. We also introduced time constraints for combinations of specified events, “negative reputation” and “acceptance of an

order", because the former can have an influence on the latter within the limited number of days. This number of days is not necessarily appropriate for other types of business reports or for different products. However, it is comparatively easy to determine the appropriate number of days through consultation with the analysts or by examining some reports. In addition, we can introduce time constraints of various combinations of events based on knowledge of the analysts. Therefore, the mining method is sufficiently practical for the task of analyzing daily business reports.

These discussions indicate our new mining method incorporating the 7 types of time constraints is efficient and helps decision making.

IV. SUMMARY AND FUTURE WORK

This paper has proposed 7 types of time constraints and a new mining method of time series event patterns. The paper has also verified the effectiveness of the method by applying it to the task of analyzing daily business reports. The method is able to discover interesting patterns more easily, allowing us to flexibly define time constraints.

In the future, we aim to introduce different indexes instead of support values. This is because it is difficult for constraints based on background knowledge to discover unexpected but interesting patterns. On the other hand, we will consider methods that seamlessly deal with discrete events and numerical events. This is because there are many events described as numerical values in medical examination data.

REFERENCES

[1] R. Agrawal and R. Srikant, "Mining Sequential Patterns," *in Proc. of the 11th Int. Conf. Data Engineering*, 1995, Taipei, Taiwan, pp. 3-14.

[2] R. Feldman, I. Dagan, and H. Hirsh, "Mining Text using Keyword Distributions," *J. of Intelligent Information Systems*, vol. 10, no.3, pp. 281-300, May, 1998.

[3] M. N. Garofalakis, R. Rastogi, and K. Shim, "SPIRIT: Sequential Pattern Mining with Regular Expression Constraints," *in Proc. of the Very Large Data Bases Conf. 1999*, 1999, Edinburgh, Scotland, UK, pp. 223-234.

[4] Y. Ichimura, Y. Nakayama, M. Miyoshi, T. Akahane, T. Sekiguchi, and Y. Fujiwara, "Text Mining System for Analysis of a Salesperson's Daily Reports," *in Proc. of the Pacific Association for Computational Linguistics 2001*, 2001, Kitakyushuu, Japan, pp. 127-135.

[5] K. Lagus, T. Honkela, S. Kaski, and T. Kohonen, "Websom for Textual Data Mining," *J. of Artificial Intelligence Review*, vol. 13, no. 5/6, pp. 335-364, Dec., 1999.

[6] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan, "Mining of Concurrent Text and Time-Series," *in Proc. of the KDD-2000 Workshop on Text Mining*, 2000, Boston, Massachusetts, USA, pp. 37-44.

[7] B. Lent, R. Agrawal, and R. Srikant, "Discovering Trends in Text Databases," *in Proc. of the 3rd Int. Conf. on Knowledge Discovery and Data Mining*, 1997, Newportbeach, California, USA, pp. 227-230.

[8] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth," *in Proc. of 2001 Int. Conf. Data Engineering*, 2001, Heidelberg, Germany, pp. 215-224.

[9] J. Pei, J. Han, and W. Wang, "Mining Sequential Patterns with Constraints in Large Databases," *in Proc. of the 11th ACM Int. Conf. on Information and Knowledge Management*, 2002, McLean, Virginia, USA, pp. 4-9.

[10] S. Sakurai, Y. Ichimura, and A. Suyama, "Acquisition of a Knowledge Dictionary from Training Examples including Multiple Values," *Proc. of the 13th Int. Symposium on Methodologies for Intelligent Systems*, 2002, Lyon, France, pp. 103-113.

[11] S. Sakurai and K. Ueno, "Analysis of Daily Business Reports Based on Sequential Text Mining Method," *in Proc. of the 2004 IEEE Int. Conf. on Systems, Man and Cybernetics*, 2004, Hague, Netherlands, pp. 3279-3284.

[12] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements," *Proc. of the 5th Int. Conf. Extending Database Technology*, 1996, Avignon, France, pp. 3-17.

[13] R. Swan and D. Jensen, "TimeMines: Constructing Timelines with Statistical Models of Word Usage," *Proc. of the KDD-2000 Workshop on Text Mining*, 2000, Boston, Massachusetts, USA, pp. 73-80.

[14] M. J. Zaki, "SPADE: An Efficient Algorithm for Mining Frequent Sequences," *Machine Learning*, vol. 42, no. 1/2, pp. 31-60, Jan., 2001.