Mining Association Rules in Temporal Document Collections

Kjetil Nørvåg*, Trond Øivind Eriksen, and Kjell-Inge Skogstad

Dept. of Computer and Information Science, NTNU
7491 Trondheim, Norway

Abstract. In this paper we describe how to mine association rules in temporal document collections. We describe how to perform the various steps in the temporal text mining process, including data cleaning, text refinement, temporal association rule mining and rule post-processing. We also describe the Temporal Text Mining Testbench, which is a user-friendly and versatile tool for performing temporal text mining, and some results from using this tool.

1 Introduction

Temporal document databases[1] can be used for efficient storage and retrieval of temporal documents. They also provide efficient support for queries involving both document contents and time [9]. However, in addition to the explicit information and knowledge that can be retrieved using these techniques, the documents also contain implicit knowledge inside particular documents, as well as inter-document and inter-version knowledge. In order to discover this knowledge, data mining techniques have to be applied. In this paper we describe how to mine association rules in temporal document collections An example of such a rule from a web newspaper is "if one version of the page contains the word *bomb*, a subsequent version will contain the word *Al Quaida*".

Finding temporal association rules in text documents can be useful in a number of contexts. Examples are 1) analysis of health records to find relationships between medicine, symptoms, and diseases, 2) investigations, and in general understanding impact of events in the real world.

The main contributions of this paper are 1) introducing temporal association rule mining in document databases, 2) identifying appropriate association rule models and algorithms for performing the association rule mining, 3) developing techniques for pre- and post-processing that increases the proportion of interesting rules, and 4) presenting some preliminary results from mining a web newspaper using the Temporal Text Mining (TTM) Testbench. It should be emphasized that mining text is quite different from mining structured data or retail transaction databases. The most important difference is the large number of itemsets/words (both number of distinct words and number of words in each "transaction").

* Email of contact author: Kjetil.Norvag@idi.ntnu.no

[1] A temporal document database stores both previous and deleted document versions in addition to current non-deleted versions, and to each version is also kept the associated timestamp.

The organization of the rest of this paper is as follows. In Section 2 we give an overview of related work. In Section 3 we perform a study of variants of temporal rule mining in order to find an appropriate starting point for mining association rules in temporal document collections. In Section 4 we describe in detail the process of mining temporal association rules in temporal document databases. In Section 5 we describe the TTM Testbench, a tool for performing the text mining process, and in Section 6 we present some results from applying the TTM Testbench on a temporal document collection. Finally, in Section 7, we conclude the paper and outline issues for further work.

2 Related work

Introduction to *data mining in general* can be found in many good text books, for example [2]. The largest amount of work in *text mining* have been in the areas of categorization, classification and clustering of documents, we refer to [1] for an overview of the area. Algorithms for mining association rules between words in text databases (if particular terms occur in a document, there is a high probability that certain other terms will occur in the same document) was presented by Holt and Chung in [3]. In their work, each document is viewed like a transaction, and each word being an item in the transaction.

Much research has been performed on aspects related to temporal data mining, and a very good survey of temporal knowledge discovery paradigms and methods is given by Roddick and Spiliopoulou [10]. As will be described in more detail in the rest of the paper, of particular interest in the context of our work is research in intertransaction association rules. The first algorithms for finding intertransaction rules described in the literature, E-Apriori and EH-Apriori[7], are based on the Apriori algorithm. These are extensions of the Apriori algorithm, where EH-Apriori also includes hashing. A further development of intertransaction rules is the FITI algorithm [12], which is specifically designed for efficient mining intertransaction rules.

A general problem in mining association rules is the selection of interesting association rules within the overall, and possibly huge set of extracted rules. Some work in this are exist, either based on statistical methods [11] or by considering the selection of association rules as a classification task [4].

Related to our work is trend analysis in text databases, were the aim is to discovery increasing/decreasing popularity of a set of terms [6, 8]. A variant of temporal association rule mining is taking into account the exhibition periods of items [5].

3 Strategy for Mining Association Rules in Temporal Document Databases

In temporal association rule mining one tries to adopt the traditional association rule mining to cover temporal relations. Different approaches have been conducted to try and utilize this extra temporal information. These approaches can be divided into five categories, *episode rules*, *trend dependencies*, *sequence rules*, *calendric rules* and *inter-transaction rules* [2]. We will in the rest of this section describe the various approaches and an analysis of their usefulness for mining rules in temporal document collections.

3.1 Strategies

Episode Rules. Episode rules are a generalization of association rules. Rules are here applied to sequences of events, each event occurring at a particular time. An episode is further a sequence of events in a partial order and an episode rule is defined as an implication on the form $\mathcal{A} \Rightarrow \mathcal{B}$, where $\mathcal{A}$ and $\mathcal{B}$ are episodes and $\mathcal{A}$ is a subepisode of $\mathcal{B}$. This technique can be used in for example predicting certain events by sequences of earlier events.

With regards to textual data this approach will produce rules where for example each word or concept is mapped to an event. A result of this can be the episode rule $\{Bush, Iraq\} \Rightarrow \{Bush, Iraq, UN\}$, predicting the word *UN* based on the words *Bush* and *Iraq*.

This technique requires that the data analyzed is represented as a sequence of events. This can be achieved with textual data, by mapping each event to a document. However, this only creates rules describing relations between whole documents. Finding rules describing relations between concepts is more difficult. If each document is represented by a single concept the task is trivial, however this severely limits the possible rules that can be found.

Sequence Rules. Sequence rules are much like episode rules and share many of their properties. Similar to episode rules, sequences are used to describe temporal relations, but they differ in the representation of the dataset. Instead of a single sequence, sequence rules use transactions like in the traditional association rules. In addition these transactions are grouped by a common concept so that rules can be found related to that concept.

An example of a sequence rule is $\{\mathcal{AB}, \mathcal{A}\} \Rightarrow \{\mathcal{C}, \mathcal{A}\}$, where $\mathcal{A}$ and $\mathcal{B}$ appears at the same time followed by $\mathcal{A}$, implies a sequence where $\mathcal{C}$ is followed by $\mathcal{A}$. The transactions in the example are grouped by a common concept. Using textual data, the same kind of rules can be found if one can map the documents to this representation. For example, can $Author : noervaag$ be used as grouping concept and each transaction represent an article produced by this author. Rules may then be found describing for example the evolution of writing patterns. This technique is not applicable if no such groupable concept is available.

Calendric Rules. With calendric association rules a predefined time unit and a calendar, given by a set of time intervals, are needed. This technique identifies rules within the predefined time intervals, which allows for seasonal changes to be analyzed. This also allows for the user to specify which time intervals he or she is most interested in. For example the user may specify the time unit to be day and time intervals by day number, for example $\{(1, 31)\,(334, 365)\}$. Here, the user is interested in rules describing patterns within January and December. The rules found is in the form $\mathcal{A} \Rightarrow \mathcal{B}$, like in traditional association rule mining, but restricted to the chosen calendar.

This approach is similar to finding traditional association rules with a limited dataset used, containing only transactions from the specified time interval. The user has with this approach more freedom in specifying the time intervals he or she is interested in. The technique does however not produce rules describing relations between items with different timestamp. That is, traditional rules are found, only limited by the calendar. For this reason, calendric rules do not cover the patterns this project aims to find, and will therefore not be studied further.

Trend Dependencies. Trend dependencies differ from the other approaches in that the attributes that are used have to be ordinal. That is, it has to be possible to create a totally ordered set of values of each attribute.

A trend dependency rule is a rule $\mathcal{A} \Rightarrow \mathcal{B}$, where $\mathcal{A}$ and $\mathcal{B}$ are patterns over a specific schema. These patterns are sets of items on the form (A, Θ), where A is a reference to a specific attribute and Θ is an element in $\{<, =, >, \geq, \leq, \neq\}$. Trend dependencies can for example be used to discover patterns like: *an employee's salary always increases over time*.

With regards to textual data, the requirement of ordinal values makes this approach less interesting. It may be possible to order some of the words or concepts that are used, but most likely this will not be the case with all of them.

Intertransaction Rules. Much of the literature focus on intratransaction association rules, which deal with relations within transactions. Only a few algorithms exist to mine intertransaction association rules, where relations across transactions (each having a timestamp) are analyzed. These algorithms usually utilize a time window to minimize computation.

Using an appropriate algorithm for finding intertransaction association rules, we can find rules on the form "*car* at time 0 and *hotel* at time 1 implies *leasing* at time 4". As can be seen, these algorithms produce rules with items from different transactions given by a timestamp. The benefits with this approach is that here you not only know that the itemset $\mathcal{B}$ follows the itemset $\mathcal{A}$, but you also get an indication on when this is supposed to happen. This can be viewed as a significant increase in potential for describing temporal patterns.

Conclusion. Based on the analysis of the different approaches, we consider the use of intertransaction rules as most appropriate for our task. An efficient algorithm for mining intertransaction association rules will be described in Section 4.3.

4 Text mining process

In this section we describe in more detail the process of mining temporal association rules in temporal document databases. The process can be divided into the following steps, each possibly consisting of a number of sub-steps as will be described shortly: 1) tokenization, 2) text filtering and refinement, 3) mining temporal association rules based on the terms resulting from the text-refinement step, and 4) extracting the *interesting* association rules.

4.1 Tokenization

In this step the terms are extracted from the text documents. Depending on the application domain, terms that are numeric values may or may not be kept for mining. This step is a mapping from documents to a list of list of terms.

4.2 Text filtering and refinement

Documents can be large and each contain a large number of distinct terms. In order to increase quality of rules as well as reduce the computational cost, in general only a subset of the terms in the documents are actually part of the rule mining process. In the text filtering and refinement step it is determined which of the terms that shall participate, and certain transformation may also be performed in order to increase the probability of useful rules. This step consists of a number of operations, each having as input a list of list of terms and producing a new list of list of terms. It should be noted that the operations can be executed in different order than how they are presented below, and that not all have to be used in one rule mining process.

Text filtering. The goal of text filtering is to remove terms that can be assumed to not contribute to the generation of meaningful rules. This step is vocabulary based, and the operations can be classified into two categories:

- Stop-word removal: Removing terms known to not be interesting or too frequently occurring, i.e., traditional stop-word removal. These terms are in general kept in a stop word list which contains terms that are considered stop words, but it might be that domain-related stop words are added as well.
- Finding interesting terms: Keeping only terms that are known to be good candidates for interesting rules. These terms are found in a particular vocabulary, which can either be general or domain-specific. In addition to domain-specific vocabularies, it is also possible to use more general vocabularies for this purpose, for example based on *nouns* or *proper nouns* (i.e., names of unique entities, in many languages like for example English these nouns start with a capital letter).

Text refinement. In order to reduce the number of terms as well as increasing quality of the contributing terms, various approaches to text refinement can be employed. We have studied the use of the following techniques:

- Stemming, which determines a stem form of a given inflected (or, sometimes, derived) word form generally a written word form. This means that a number of related words all will be transformed into a common term.
- Semantic analysis of the text can be used to combine expressions (two or more terms) into one. One simple example of such combination is *United* and *States* into *United States*.

Term selection. The goal of text refinement is to find those terms that are assumed to contribute most to giving meaningful rules. In order to increase quality of rules as well as reduce the computational cost, only a subset of the k terms in each document are actually part of the rule mining process. In the term selection step, the terms are selected based on their expected importance. In our work we have studied the use of two term-selection techniques: a) using the k first terms in a document, and b) using the k highest ranked terms based on the TF-IDF (term-frequency/inverse document frequency) weight of each term. The first technique is in particular useful for document collections where each document contains an abstract of the contents in the rest of the document. However, the latter is most appropriate for general document collections. It should be noted that there is a danger of filtering out terms that could contribute to interesting rules when only a subset of the terms are used, so the value of k will be a tradeoff between quality and processing speed.

4.3 Rule mining using the First Intra Then Inter (FITI) Algorithm

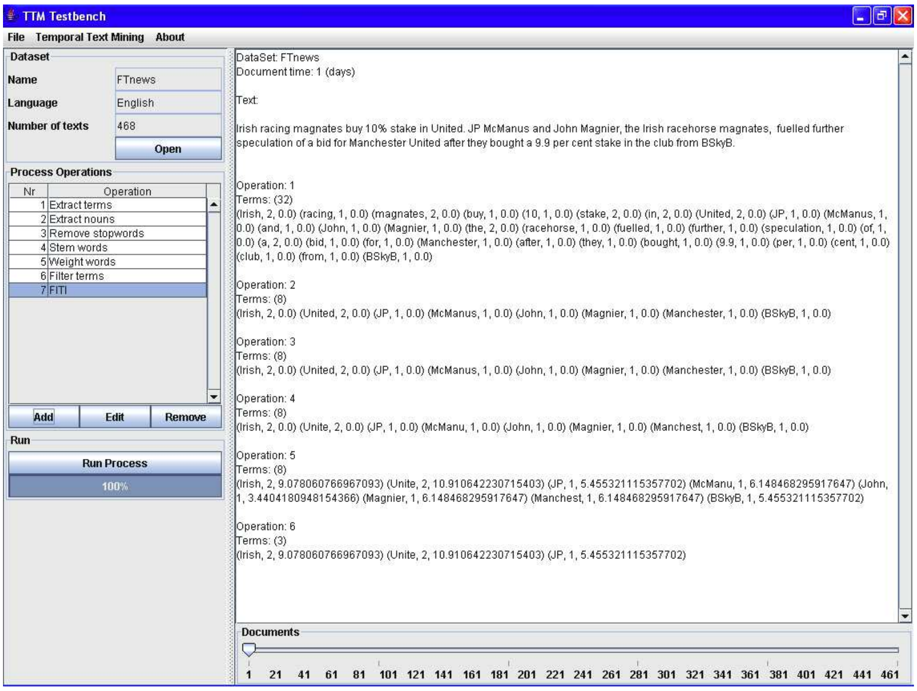
In order to find intertranaction association rules (see Section 3.1), we employ the FITI algorithm [12]. The FITI algorithm for mining intertransaction rules is based on the property that *a large intertransaction itemsets must be made up of large intratransaction itemsets*. That is, for an itemset to be large in intertransaction association rule mining it will also need to be large using traditional association rule techniques. This property can be utilized to reduce the complexity of creating intertransaction rules. If all large intratransaction itemsets are known in advance, the candidate generation task only needs to consider itemsets present in this set. The FITI algorithm is based on this way of thinking and involves three phases: 1) mining large intratransaction itemsets, 2) database transformation, and 3) mining large intertransaction itemsets. Due to space constraints we do not go into more details of the FITI algorithm here but refer the interested reader to [12].

4.4 Rule post-processing

In text mining in general, and temporal text mining in particular, a very large number of association rules will be found. A very important and challenging problem is to find those that are *interesting*.

Similar to traditional intertransaction association rules, parameters like *item set size* and measures like *support* and *confidence* are also important when creating intertransaction item sets and selecting final rules. Unlike traditional rule mining where often as large as possible item sets are created, in mining rules in text usually the rules based on relatively small item sets are sufficient. It should also be mentioned that because of the number of distinct terms the typical minimum support for association rules in text databases can be relatively low.

One particular aspect of rule mining in text is that often a high support means the rule is too obvious and thus less interesting. These rules are often a result of frequently occurring terms and can partly be removed by specifying the appropriate stop words. However, many will remain, and these can to a certain extent be removed by specifying a maximum support on the rules, i.e., the only resulting rules are those above a certain



Name: FTnews
Language: English
Number of texts: 468

Nr	Operation
1	Extract terms
2	Extract nouns
3	Remove stopwords
4	Stem words
5	Weight words
6	Filter terms
7	FITI

```
DataSet: FTnews
Document time: 1 (days)

Text:

Irish racing magnates buy 10% stake in United. JP McManus and John Magnier, the Irish racehorse magnates,  fuelled further
speculation of a bid for Manchester United after they bought a 9.9 per cent stake in the club from BSkyB.


Operation: 1
Terms: (32)
(Irish, 2, 0.0) (racing, 1, 0.0) (magnates, 2, 0.0) (buy, 1, 0.0) (10, 1, 0.0) (stake, 2, 0.0) (in, 2, 0.0) (United, 2, 0.0) (JP, 1, 0.0) (McManus, 1,
0.0) (and, 1, 0.0) (John, 1, 0.0) (Magnier, 1, 0.0) (the, 2, 0.0) (racehorse, 1, 0.0) (fuelled, 1, 0.0) (further, 1, 0.0) (speculation, 1, 0.0) (of, 1,
0.0) (a, 2, 0.0) (bid, 1, 0.0) (for, 1, 0.0) (Manchester, 1, 0.0) (after, 1, 0.0) (they, 1, 0.0) (bought, 1, 0.0) (9.9, 1, 0.0) (per, 1, 0.0) (cent, 1, 0.0)
(club, 1, 0.0) (from, 1, 0.0) (BSkyB, 1, 0.0)

Operation: 2
Terms: (8)
(Irish, 2, 0.0) (United, 2, 0.0) (JP, 1, 0.0) (McManus, 1, 0.0) (John, 1, 0.0) (Magnier, 1, 0.0) (Manchester, 1, 0.0) (BSkyB, 1, 0.0)

Operation: 3
Terms: (8)
(Irish, 2, 0.0) (United, 2, 0.0) (JP, 1, 0.0) (McManus, 1, 0.0) (John, 1, 0.0) (Magnier, 1, 0.0) (Manchester, 1, 0.0) (BSkyB, 1, 0.0)

Operation: 4
Terms: (8)
(Irish, 2, 0.0) (Unite, 2, 0.0) (JP, 1, 0.0) (McManu, 1, 0.0) (John, 1, 0.0) (Magnier, 1, 0.0) (Manchest, 1, 0.0) (BSkyB, 1, 0.0)

Operation: 5
Terms: (8)
(Irish, 2, 9.078060766967093) (Unite, 2, 10.910642230715403) (JP, 1, 5.455321115357702) (McManu, 1, 6.148468295917647) (John,
1, 3.4404180948154366) (Magnier, 1, 6.148468295917647) (Manchest, 1, 6.148468295917647) (BSkyB, 1, 5.455321115357702)

Operation: 6
Terms: (3)
(Irish, 2, 9.078060766967093) (Unite, 2, 10.910642230715403) (JP, 1, 5.455321115357702)
```

Fig. 1. TTM Testbench - After performing operations on the document collection.

minimum support and less than a certain maximum support. Another technique that can be used to remove unwanted rules is to specify *stop rules*, i.e., rules that are common and can be removed automatically.

5 The TTM Testbench

The Temporal Text Mining (TTM) Testbench is a user-friendly application developed in order to simplify experimenting with rule mining in temporal document collections. Text mining using the TTM Testbench is essentially a process consisting of three phases: 1) loading the document collection, 2) choosing which operations that should be performed, including pre- and post-processing as well as rule mining, and 3) let the system perform the selected operations and present the final result, i.e., the association rules generated based on the document collection, operations, and parameters.

Fig. 1 illustrates the TTM Testbench in use, after a document collection has been loaded. Studying the individual documents is also easy through the use of the scroll bar shown below the text window. To the left is the operation selection tool, where one or more operations are selected. It is also possible to view the result from the individual operations (for example extracted terms or TF-IDF values) for each document.

A number of operations are available in the TTM Testbench, each essentially working as part of a filtering/operator pipeline. The operators used in the experiments reported in this paper are:

Extract terms: Extract all terms in each document.

Extract nouns: Extract all proper nouns (i.e., nouns starting with a capital letter).

Remove stop words: Remove stop words. The stop words are stored in a stop word file where thee user himself add appropriate stop words.

Stem words: Perform stemming of the text. Both stemming of English and of Norwegian are supported.

Weight terms: Calculate a weight to each term, using TF-IDF.

Filter terms: Filter terms that should be included in the process when mining for temporal association rules. This operation has a number of options, the most important being the possibility of only keeping a certain number k of terms, which is either the k highest ranked based on TF-IDF value, or the k first words in the document.

FITI: The currently implemented algorithm for mining temporal association rules in the document collection that has been loaded into the system. For this operator a number of parameters are available, including minimum and maximum support, minimum and maximum confidence, the size of the time window, and maximum set size.

6 Experimental Evaluation

A number of experiments have been performed in order to validate the rule mining approach as well as studying the impact of various pre- and post-processing operations and rule mining parameters. We will in the following describe three of the experiments which have been performed using the TTM Testbench. The experiments are based on a relatively small collection, consisting of 38 days of the front page of the online version of Financial Times (468 news articles). The size of the dataset is relatively limited and it is therefore not expected that really interesting rules will be found using this dataset. However, it is believed that the dataset is large enough for identifying meaningful rules. Default parameters for the rule mining have been min/max support of 0.1/0.5, min/max confidence of 0.7/1.0, max time span 3 days, and max set size of 3.

In order to determine the quality of the mined rules, evaluation criteria have to be defined. In these experiments focus will be on the quality of the rules found. To determine if a rule that has been mined is meaningful or not, is difficult (if not impossible) to determine by using automatic methods. For this project, a manual approach to determine rule quality is therefore used. Focus will be on finding rules that include terms with some semantic meaning, or that can be said to be self-explanatory, for example *Bush*, *Iraq* and *UN*. Automatic methods for determining if a rule is meaningful or not, are outside the scope of this paper. This should however be considered as a field for further research.

All terms. In the first experiment the only filtering performed is stop word removal and stemming. Stop words are removed in order to minimize the amount of rules that will not have any meaning, and stemming is included to group similar words and increasing the support of these. As can be expected, without limiting the number of terms there will be too many resulting rules. This result argues that some technique should be used to extract the semantics of the text and represent this by fewer terms, if rule mining is to be successful.

Filtering. Given the observation that using all terms is not feasible, a filtering operation can minimize the solution space. The most simple approach is to only use the first k terms of each text. However, the resulting rules do not carry much meaning. Examples of such rules include $(John, 0) \Rightarrow (hit, 1)$ or $(back, 0) \Rightarrow (Iraq, 1)$. This also points towards a more semantically-based approach.

An extension of using only the k first terms of each text is to use the k most important terms from each document. This is an attempt to reduce the number of features describing the text, without loosing too much of the semantics. The TF-IDF-measure is used to determine the most important terms, and the k highest ranked terms are kept. Although the quality of the rules increased from the previous experiments, most of them were still not very meaningful.

Noun extraction. Nouns have a high semantic meaning, and in order to see how this affects the rules we used proper nouns only as basis for the rule mining. Also in this experiment, filtering is performed. The reason for this is that there might be too many nouns present, making the approach unfeasible. In the result reported here the eight highest-weighted proper nouns from each document are used. Results of this experiment are shown in Fig. 2.

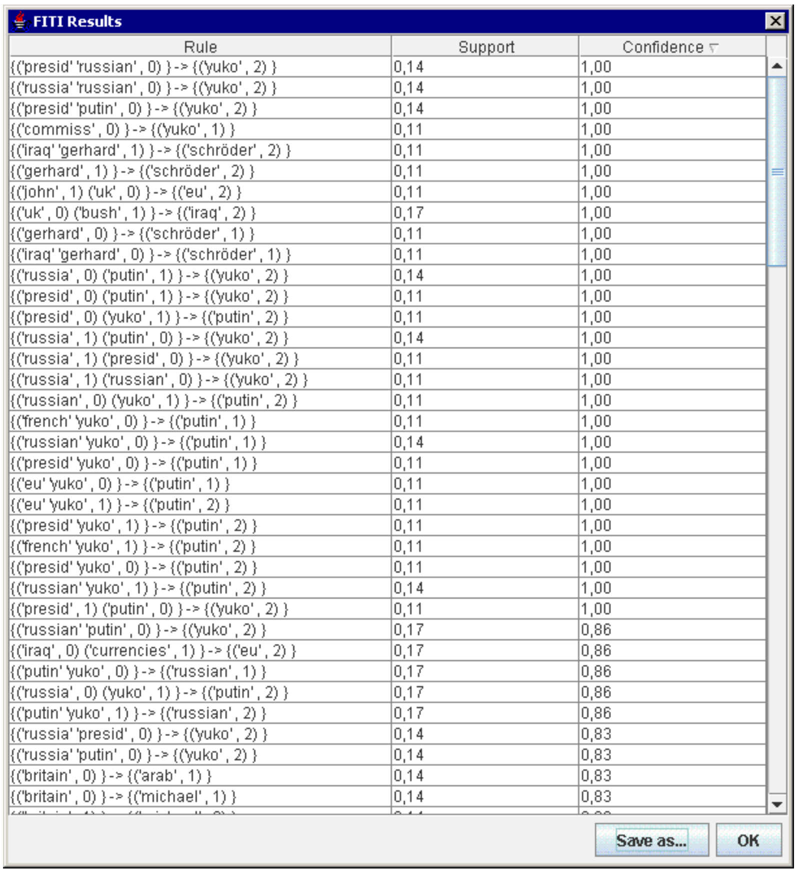
7 Conclusions and further work

In this paper we have described how to mine association rules in temporal document collections. We described how to perform the various steps in the temporal text mining process, including data cleaning, text refinement, temporal association rule mining and rule post-processing. We also described the TTM Testbench and some results from using this tool.

Future work includes the development of appropriate metrics for rule quality and develop new techniques for rule post-processing. Regarding the rule mining itself, it is obvious that it is a computationally very costly process, and more work should be performed on how to optimize this part of the process.

References

1. S. Chakrabarti. *Mining the Web - Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, 2003.
2. M. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall, 2003.
3. J. D. Holt and S. M. Chung. Efficient mining of association rules in text databases. In *Proceedings of CIKM'99*, 1999.
4. D. Janetzko, H. Cherfi, R. Kennke, A. Napoli, and Y. Toussaint. Knowledge-based selection of association rules for text mining. In *Proceedings of ECAI'2004*, 2004.
5. C.-H. Lee, C.-R. Lin, and M.-S. Chen. On mining general temporal association rules in a publication database. In *Proceedings of ICDM'2001*, 2001.



Rule	Support	Confidence ▿
{('presid' 'russian' , 0) } -> {('yuko' , 2) }	0,14	1,00
{('russia' 'russian' , 0) } -> {('yuko' , 2) }	0,14	1,00
{('presid' 'putin' , 0) } -> {('yuko' , 2) }	0,14	1,00
{('commiss' , 0) } -> {('yuko' , 1) }	0,11	1,00
{('iraq' 'gerhard' , 1) } -> {('schröder' , 2) }	0,11	1,00
{('gerhard' , 1) } -> {('schröder' , 2) }	0,11	1,00
{('john' , 1) ('uk' , 0) } -> {('eu' , 2) }	0,11	1,00
{('uk' , 0) ('bush' , 1) } -> {('iraq' , 2) }	0,17	1,00
{('gerhard' , 0) } -> {('schröder' , 1) }	0,11	1,00
{('iraq' 'gerhard' , 0) } -> {('schröder' , 1) }	0,11	1,00
{('russia' , 0) ('putin' , 1) } -> {('yuko' , 2) }	0,14	1,00
{('presid' , 0) ('putin' , 1) } -> {('yuko' , 2) }	0,11	1,00
{('presid' , 0) ('yuko' , 1) } -> {('putin' , 2) }	0,11	1,00
{('russia' , 1) ('putin' , 0) } -> {('yuko' , 2) }	0,14	1,00
{('russia' , 1) ('presid' , 0) } -> {('yuko' , 2) }	0,11	1,00
{('russia' , 1) ('russian' , 0) } -> {('yuko' , 2) }	0,11	1,00
{('russian' , 0) ('yuko' , 1) } -> {('putin' , 2) }	0,11	1,00
{('french' 'yuko' , 0) } -> {('putin' , 1) }	0,11	1,00
{('russian' 'yuko' , 0) } -> {('putin' , 1) }	0,14	1,00
{('presid' 'yuko' , 0) } -> {('putin' , 1) }	0,11	1,00
{('eu' 'yuko' , 0) } -> {('putin' , 1) }	0,11	1,00
{('eu' 'yuko' , 1) } -> {('putin' , 2) }	0,11	1,00
{('presid' 'yuko' , 1) } -> {('putin' , 2) }	0,11	1,00
{('french' 'yuko' , 1) } -> {('putin' , 2) }	0,11	1,00
{('presid' 'yuko' , 0) } -> {('putin' , 2) }	0,11	1,00
{('russian' 'yuko' , 1) } -> {('putin' , 2) }	0,14	1,00
{('presid' , 1) ('putin' , 0) } -> {('yuko' , 2) }	0,11	1,00
{('russian' 'putin' , 0) } -> {('yuko' , 2) }	0,17	0,86
{('iraq' , 0) ('currencies' , 1) } -> {('eu' , 2) }	0,17	0,86
{('putin' 'yuko' , 0) } -> {('russian' , 1) }	0,17	0,86
{('russia' , 0) ('yuko' , 1) } -> {('putin' , 2) }	0,17	0,86
{('putin' 'yuko' , 1) } -> {('russian' , 2) }	0,17	0,86
{('russia' 'presid' , 0) } -> {('yuko' , 2) }	0,14	0,83
{('russia' 'putin' , 0) } -> {('yuko' , 2) }	0,14	0,83
{('britain' , 0) } -> {('arab' , 1) }	0,14	0,83
{('britain' , 0) } -> {('michael' , 1) }	0,14	0,83

Fig. 2. Experimental results, nouns with stemming and the 8 first nouns kept.

6. B. Lent, R. Agrawal, and R. Srikant. Discovering trends in text databases. In *Proceedings of KDD'1997*, 1997.
7. H. Lu, L. Feng, and J. Han. Beyond intratransaction association analysis: mining multidimensional intertransaction association rules. *ACM Trans. Inf. Syst.*, 18(4):423–454, 2000.
8. Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of KDD'05*, 2005.
9. K. Nørvåg. Supporting temporal text-containment queries in temporal document databases. *Journal of Data & Knowledge Engineering*, 49(1):105–125, 2004.
10. J. F. Roddick and M. Spiliopoulou. Survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767, 2002.
11. P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of KDD'2002*, 2002.
12. A. K. H. Tung, H. Lu, J. Han, and L. Feng. Efficient mining of intertransaction association rules. *IEEE Transactions on Knowledge and Data Engineering*, 15(1):43–56, 2003.