Semantic-Based Temporal Text-Rule Mining

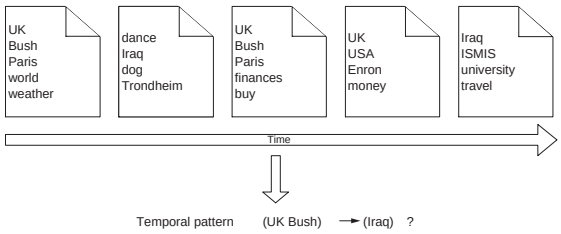
Kjetil Nørvåg* and Ole Kristian Fivelstad

Dept. of Computer Science, Norwegian University of Science and Technology
Trondheim, Norway

Abstract. In many contexts today, documents are available in a number of versions. In addition to *explicit knowledge* that can be queried/searched in documents, these documents also contain *implicit knowledge* that can be found by text mining. In this paper we will study association rule mining of temporal document collections, and extend previous work within the area by 1) performing mining based on *semantics* as well as 2) studying the impact of appropriate techniques for ranking of rules.

1 Introduction

In many contexts today, documents are available in a number of versions. Examples include web newspapers and health records, where a number of timestamped document versions are available. In addition to *explicit knowledge* that can be queried/searched in documents, these documents also contain *implicit knowledge.* One category is inter-document knowledge that can be found by conventional text-mining techniques. However, with many versions available there is also the possibility of finding *inter-version knowledge*. An example of an application is given in the figure below, where a number of document versions are available, and where the aim is to find and/or verify temporal patterns:



In the example above, one possible temporal rule is the terms[1] *UK* and *Bush* appearing in one version means a high probability of *Iraq* to appear in one of the following versions.

How to mine association rules in temporal document collection has been previously described in [16]. In the previous work, the rule mining was performed on *words* extracted from the documents, and ranking of rules (in order to find the most interesting

* E-mail of contact author: Kjetil.Norvag@idi.ntnu.no

[1] A term can be a single word as well as multiword phrase.

ones) was based on traditional measures like support and confidence. However, based on the results it was evident that using simple words did not give satisfactory results, and that more appropriate measures were needed for rule ranking.

In this paper we extend the previous work by performing the temporal mining based on *semantics* as well as studying the impact of other techniques for ranking of rules. Thus the *main contributions* of this paper are 1) presenting the appropriate pre-processing for use of semantics in temporal rule mining, 2) studying the impact of additional techniques for ranking of rules, and 3) presenting some preliminary results from mining a web newspaper.

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 outline the assumed data model, rule mining process, and provide an introduction to our Temporal Text Mining (TTM) Testbench tool. In Section 4 we describe how to perform semantic-based pre-processing. In Section 5 we describe techniques that can increase quality of rule selection by considering semantic similarity. In Section 6 we describe experiments and results. 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 [4]. The largest amount of work in *text mining* have been in the areas of categorization, classification and clustering of documents, we refer to [3] for an overview of these 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 [6]. In their work, each document is viewed like a transaction, and each word being an item in the transaction. In [5] a more thorough overview of previous research in rule mining of text collections is given, with particular emphasis on the case when additional background information is available.

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 [17]. 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[13], 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 [20], 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 [18] or by considering the selection of association rules as a classification task [8].

Related to our work is trend analysis in text databases, were the aim is to discover increasing/decreasing popularity of a set of terms [11, 15]. A variant of temporal association rule mining is taking into account the exhibition periods of items [10].

3 Preliminaries

In this section we outline the underlying data model for our work, the rule mining process, and a description of the TTM Testbench tool.

3.1 Data Model

We will now outline the data model for temporal documents we use as context for our research. Note that *document* D_i is here used as a generic term, specific types of documents include web pages as well as document formats like MS Word and Adobe PDF. For these document types pre-processing will be employed in order to filter out the actual text from formatting information etc.

The document collection C_i on which we perform the rule mining are assumed to be (or can be converted to) an ordered list of documents $C = [D_1...D_n]$. A document in this context can be the one and only version of a document, or it can be a particular version of a document. Each document is timestamped with the time of creation, and is essentially a tuple containing a timestamp T and an ordered list of words, i.e., $D = (T, [w_1, ..., w_k])$. A word w_i is an element in the vocabulary set V, i.e., $w_i \in V$. There can be more than one occurrence of a particular word in a document version, i.e., it is possible that $w_i = w_j$.

3.2 Rule Mining Process

Mining association rules from a text collections can be described as a 3-step process consisting of 1) pre-processing, 2) the actual mining and 3) post-processing. In the pre-processing phase the documents are converted from external documents into some common representation, words are extracted (tokenization), and then various operations might be performed on the text aiming at increasing the quality of the results or reducing the running time of the mining process. Then the actual mining is performed, resulting in a number of association rules. The number of rules can be very high, and in the post-processing phase the system tries to determine which rules are most interesting, based on some measure. We will now describe the steps as performed in the previous word-centric approach. In Section 4 we will describe how to improve by using semantics.

Pre-Processing In word-centric pre-processing the text of the documents is filtered and refined. In general the processing time increases with both number of words and size of vocabulary, so the aim of the pre-processing is to reduce both without significantly reducing the quality of the results.

The goal of text filtering is to remove words that can be assumed to not contribute to the generation of meaningful rules. One simple technique is stop-word removal, in which words occurring in a separate user-maintained stop-word list are removed from the text. In addition, words that are very frequently occurring can be removed.

In order to reduce the vocabulary size as well as increasing quality of the contributing terms, stemming can be performed. By employing stemming, a number of related words will be transformed into a common form (similar to the stem of the words).

Finally, term selection can be performed in order to reduce the number of terms. In this process, a subset of the k terms most important words in each document are selected. One such technique we have employed is using the k highest ranked terms based on the TF-IDF (term-frequency/inverse document frequency) weight of each. 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.

Rule Mining Techniques for temporal rule mining can be classified into a number of categories [4]. As described in [16] the most appropriate in the context of temporal rule mining is *intertransaction association rules.* 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. In order to find intertransaction association rules, we employ a variant of the FITI algorithm [20].

Rule Post-Processing From the potentially high number of rules created during the rule mining, a very important and challenging problem is to find those that are *interesting*. Traditionally, measures like *support* and *confidence* have been used. Unfortunately, these measures have been shown to be less useful in text mining. One particular aspect of rule mining in text is that a high support often 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 minimum support and less than a certain maximum support. In section 5 we will describe two approaches more suitable in our context.

3.3 The Temporal Text Mining Testbench

In order to help discovering inter-version knowledge as well as developing new techniques for this purpose, we have developed the *Temporal Text Mining (TTM) Testbench* tool. The TTM Testbench is a user-friendly application that provides powerful operators for rule mining in temporal document collections, as well as providing extensibility for other text mining techniques.

The TTM Testbench consists of two applications: one for converting a document collection into the TTM format (essentially XML files containing the text and additional metadata), and one for performing the actual mining (which in general will be performed a number of times for each document collection, with different operations and parameters). A number of operations are available in the TTM Testbench, each essentially working as part of a filtering/operator pipeline. Examples of operators include **ExtractTerms**, **RemoveStopWords**, **FilterTerms**, **ExtractConcepts** and **FITI**. Text mining on a collection is performed by choosing which operations should be performed, and let the system perform the selected operations and present the final result. The result of a rule mining process is a number of rules, for example:

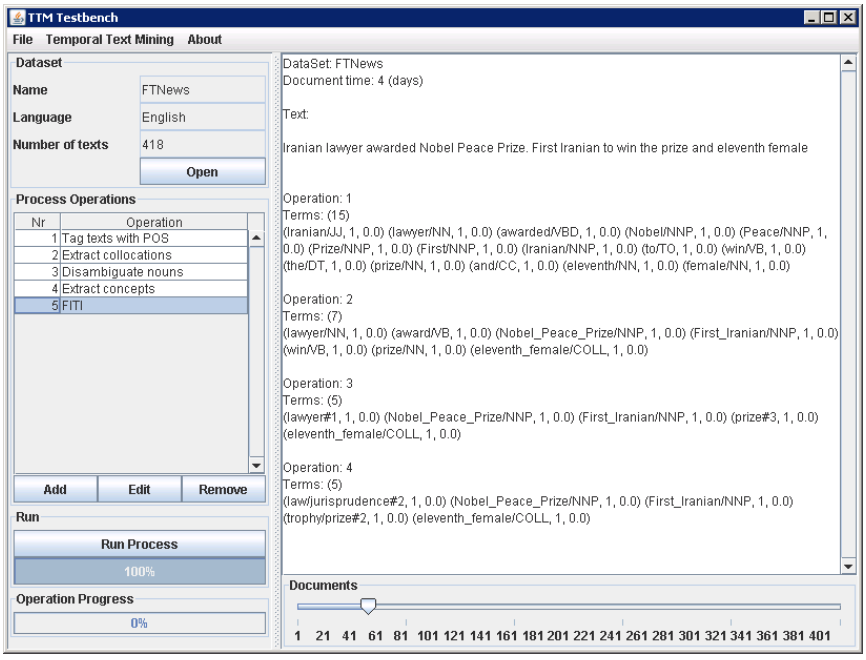


Fig. 1. Screenshot of TTM Testbench after performing operations on a document collection

Rule	Sup	Conf	Sim
(('attack', 0) ('profits', 1))→('bush', 2)	0.11	1.0	0.3
...	...	...	...

The above rule says that if the word *attack* appears in a document version one day, and the word *profits* the day after, there is a high probability that the word *bush* will appear the third day (this is an actual example from mining a collection of Financial Times web pages). The last three columns give the *support*, *confidence*, and *semantic similarity* (to be described in more detail in Section 5.1) for the rule. Fig. 1 shows the TTM Testbench and the results after each text refinement operation, using the semantic operators which will be described in more detail below.

4 Semantic-Based Pre-Processing

Performing the mining based on words extracted and refined as described in Section 3 did not achieve the desired quality. Factors contributing to the problem include those described above, i.e., feature dimensionality (i.e., vocabulary size) and feature sparsity, but also semantic aspects like synonyms (words having same or almost same meaning) and homonyms (words with same spelling but different meaning).

Considering semantics in the pre-processing phase could reduce the problems with synonyms and homonyms. In addition, by employing *concepts* instead of words in the rule mining process the dimensionality can be reduced, in addition to giving rules not

found when not considering semantics. This is typically words that each have a low frequency but when represented as a common concepts could be important. An example is the concept *vehicle* used instead of the words *bike car* and *lorry*. This considerably reduces dimensionality, in addition to giving rules containing these words higher support, and in that way increasing the probability that they will be found by the user, or detected automatically by the system.

We will in the following describe how semantic-based pre-processing and how it is integrated into the TTM Testbench. Note that we only consider semantics in the pre- and post-processing, while the mining is performed on semantic concepts in the same way as mining previously was performed on words.

The aim of the semantic-based pre-processing is twofold: find collocations (sequence of words or terms that occur together, for example *oil price*) and extract concepts (from single words or collocations). As will be described, this is performed in a multistep process involving: 1) part-of-speech tagging, 2) collocation extraction, 3) word-sense disambiguation (WSD), and 4) concept extraction.

For WSD and concept extraction we employ WordNet,[2] which essentially provides us with words and semantic relationships (for example hypernyms) between the words, and *synset*, which are words considered semantically equivalent (synonyms). For each word sense there is also a short description (gloss).

4.1 Part-Of-Speech Tagging

Some word classes are more important than other in the mining process. In order to keep the number of participating terms as low as possible, it might be useful to filter out terms from only one or a few word classes from the text, for example nouns and adjectives. This can be performed by *part-of-speech tagging*. TTM Testbench uses the *Stanford Log-linear Part-Of-Speech Tagger*[3] to tag the document collection. This tagger uses a Maximum Entropy model, similar to stochastic tagging [19].

After the texts in the document collection are tagged, the operation extracts words tagged with one of a set of user-specified part-of-speech tags. Available tags include nouns, proper nouns and proper noun groups, verbs, adjectives, numbers and adverbs.

4.2 Collocation Extraction

A collocation is an expression consisting of two or more words that corresponds to some conventional way of saying things [14], for example *weapon of mass destruction* or *car bomb*. Collocations are common in natural languages, and a word can not be classified only on the basis of its meaning, sometimes co-occurrence with other words may alter the meaning dramatically.

The task of finding collocations is essentially to determine sequences of words or terms which co-occur more often than would be expected by chance. Hypothesis testing can be used to assess whether this is the case. In our work, the chi-square ($\mathcal{X}^2$) test has been used. When a noun occur together with another noun in the text they are

[2] http://wordnet.princeton.edu/

[3] http://nlp.stanford.edu/software/tagger.shtml

collocation candidates, and the chi-square test is used to determine of they should be considered as a collocation.

4.3 Word-Sense Disambiguation

Word sense disambiguation (WSD) is the process of examining word tokens in a text and specify exactly which sense of each word is being used. As an example, consider the word *bank*, and two of its distinct senses: 1) a financial institution and 2) sloping land. When this word occur in a text, it is usually obvious for a human which of the senses of *bank* that is used, but creating robust algorithms for computers to automatically perform this task or more difficult.

In the TTM Testbench we employ the Lesk and adapted Lesk algorithms for WSD [1, 12]. Using these algorithms, the process of WSD consists of two steps: 1) find all possible senses for all the relevant words in a text, and 2) assign each word its correct sense. The first step is straightforward and accomplished by retrieving the possible senses from WordNet. The second step is accomplished by matching the context of the word in the document with the description of the senses in WordNet (glosses). Because the dictionary glosses tend to be fairly short, and may thus provide an insufficient vocabulary for fine-grained distinctions in relatedness, *extended gloss overlaps* is used to overcome the problem of too short glosses [1]. To create the extended gloss in the adapted Lesk algorithm, the algorithm uses the glosses of related words in WordNet (for example hypernyms, hyponyms, meronyms and holonyms for nouns, and hypernyms and troponyms for verbs).

4.4 Concept Extraction

Aiming at improving quality as well as reducing number of items in the mining process, terms are transformed into concept-level document features. This is done by utilizing the hierarchical structure of WordNet. Note that the concept extraction operation is dependent on WSD, since a word may have different senses, and these are linked to different synsets. The operation has three methods for finding concepts in a document. These are described in the following.

First, WordNet contains a relation called *category*. This relation links a synset to a higher-level category, where the category is represented by another synset. An example of this is that *basic training* is linked to the category *military*. By exploring this relation for each disambiguated word, it is possible to extract a set of categories which are descriptive of the contents of a document. Note however that only a limited set of the synsets in WordNet are linked to a category.

The second method of finding concepts in a document is based on finding common parent synsets of the words in the document. This is performed for each combination of disambiguated nouns in the texts. If the distance between the two words is below or equal to a user-specified threshold, the common parent synset is extracted as a concept. As an example of this, consider a part of the WordNet hierarchy, where *yen* and *euro* has *monetary unit* as a common ancestor, but while *euro* is direct child of *monetary unit* , *yen* is child of *Japanese monetary unit* which is child of *monetary unit*. Depending on the distance threshold, this may be extracted as a concept.

Finally, if no concepts was found using the two methods presented above, the user can specify that the parent node(s) of a word is to be extracted in addition to the word. This is found using the hypernym-relation. Recall the figure above, if only *euro* is present in document, *monetary unit* can be extracted. This method may however result in very high feature dimensionality, and increase the complexity in the rule mining process.

In addition, this concept extraction operation tries to resolve the problem of synonyms in the text. This is done by replacing disambiguated words with the two first words in the synset it belongs to. The reason for using two words instead of only one, is that this may lead to more meaningful terms. For example, if the word *auto* is present in a document, and it belongs to the synset {car, auto, automobile, machine, motorcar}, then *auto* is replaced with the term *car/auto*. All words in the document collection which belong to this synset will therefore be represented by this term.

5 Post-Processing

In general, the number of rules from rule mining of text will be very high. In order to reduce this to an amount that can be useful for a user, in the post-processing phase the most interesting rules are selected based on ranking the rules on some interestingness measure(s). Although the traditional support and confidence measures can be employed, these will often have less value in our context. For example, when mining temporal text databases, many interesting rules are rare, i.e., have a low support. We have studied the use of two other techniques that could have potential in our context. The measures are based on 1) semantic distance and 2) clustering.

5.1 Semantic Similarity

Words present in an association rule and that are close together (semantically related) in a knowledge hierarchy like WordNet, are more likely to be known by the user already. Therefore, rules where the words are less semantically related, can be considered more interesting [2].

The semantic similarity can then be used to rank the association rules. The higher the score, the more semantically similar the words in the antecedent and the consequent of the rule are. The rules with the lowest scores can therefore be considered interesting.

In order to calculate semantic distance we use the *JCn Measure* [9]. This measure is based on information content, defined in as the negative log likelihood of encountering an instance of the concept, i.e.:

$$IC(c) = -log(\frac{freq(c)}{N})$$

where $freq(c)$ is the frequency of the concept, and N is the number of concepts in the corpus. The similarity measure of two concepts, c_1 and c_2, is then defined by the following formula, where c is the most specific concept in common between c_1 and c_2 (for example, the most specific concept in common between *desktop computer* and *portable computer* could be *personal computer*):

$$sim(c_1, c_2) = IC(c_1) + IC(c_2) - 2 * IC(c)$$

This measure is calculated after the association rules have been mined. The score of an association rule is calculated as the average semantic similarity between the words in the antecedent and the consequent of the rule. However, note that it is only possible to calculate semantic similarity with disambiguated nouns or collocations which are present in WordNet. This is because the similarity is calculated between synsets, and the sense is needed to know which synset a word is present in.

5.2 Clustering

Many association rules can be said to display commonsense, for example *hammer⇒nail*. *Hammer⇒shampoo* on the other hand, is more interesting because hammer has little relation with shampoo, they can be said to be dissimilar. With this in mind, dissimilarity between the items can be used to judge the interestingness of a pattern. Based on the approach for structured data presented by Zhao et al. [21] we have experimented with clustering to measure the dissimilarity between items in an association rule.

In the first step of clustering-based rule selection, the document collection is clustered so that documents are grouped together according to their contents. Then, given an association rule A⇒B where A is in cluster C_A and B is in C_B, interestingness is defined as the distance between the two clusters C_A and C_B:

$$Interestingness(A \Rightarrow B) = Dist(C_A, C_B)$$

If the antecedent or consequent consists of more than one item, interestingness is defined as the minimal distance between clusters of antecedent and consequent. Finally, only rules which have terms from different clusters in the antecedent and the consequent are presented to the user.

6 Experiments And Results

This section presents some of the results from applying semantics in the rule mining. The experiments have been performed using the TTM Testbench, extracting collocations and concepts as described above, resulting in association rules that span across texts with different timestamps.

Filtering and weighting (cf. Section 3.2) have not be employed, since the IDF part of TF-IDF dampens the weight of terms which appear in many documents. This may not be always be desired, since association rules containing frequent terms in some cases can be interesting.

A number of document collections based on web newspapers have been used in the experiments. Each document collection have been created by downloading the web page once a day. Due to space constraints we will in this paper only report from the use of a collection based on Financial Times pages. Mining the other collections gave similar results. Due to limitations on the FITI implementation where the memory usage

increases with document collection size, we have in the reported results used a relatively small collection consisting of only 107 documents.

The parameters for the operations are as follows:

- Collocation extraction: Only verbs and adjectives are extracted in addition to collocations and single-word nouns.
- Word sense disambiguation: The adapted Lesk algorithm is used with context size of 6 words, and verbs and adjectives are not kept after the disambiguation process.
- Concept extraction: We set the maximum distance in the WordNet hierarchy to 5 (this includes the words themselves), parent nodes of words with no concepts are not added, and original terms are not kept when a concept is found.

As a result, the following terms will be extracted from each document and used in the rule mining process:

- Collocations.
- Proper nouns and proper noun groups.
- Common parents between terms in the same document.
- Categories.
- Disambiguated nouns with no common parent or category.
- Nouns which have not been disambiguated.

The parameter values for the FITI algorithm:

Parameter	Value
Minimum support	0.1
Maximum support	0.5
Minimum confidence	0.5
Maximum confidence	1.0
Maxspan (time/days)	3
Max set size (terms in rule)	3

Unfortunately, experiments showed that determining interestingness based on clustering did not work particularly well. As as result, we used only the semantic similarity measure for rating rules (note that only rules containing at least one disambiguated word on each side of the rule will get a score).

6.1 Evaluation Criteria

Automatically deciding if a rule is interesting or not, is difficult, if not impossible. The main focus in this project will be to see if the association rules and their items are meaningful, and to study whether there is any difference between rules with a high semantic similarity and rules with low semantic similarity; the idea is that rules with low semantic similarity are more interesting than those with high similarity.

6.2 Results From Mining The Financial Times Collection

The result of this experiment was 56 rules (the complete set of rules is given in Table 1). In Tables2-4, a subset of 15 rules are presented: The 5 first with no semantic similarity (Table 2, keep in mind that it is not possible to calculate the semantic similarity of rules not containing any disambiguated terms, concepts or categories, these will therefore get a semantic similarity of zero and thus appear first in the result set), the 5 with lowest semantic similarity (Table 3), and the 5 with highest semantic similarity (Table 4). The rule numbers in the rules presented for the individual experiments refer to their number in the full result set.

The terms present in the rules will sometimes include the symbol #, this is used to indicate the sense number of the term in WordNet. This can be used by the user in a lookup in WordNet (for example, the user can determine whether the term *market/marketplace* in rule 22 means a physical location in a city or the world of commercial activity). Another symbol which may appear, is */nnp*. This means that the term is a proper noun. One aspect that becomes clear when inspecting the rules is that it is easier to understand the meaning of the items when they are represented by two synonyms. As an example, see rule 54. Here the item *depository_financial_institution/bank* is present. Because a synonym is present, the rule is more meaningful than if for example only *bank* was present.

As the above show, many of the terms included in the rules are meaningful, and the user can therefore make sense of the discovered rules. Whether semantic similarity is able to distinguish between interesting and uninteresting rules or not, is difficult to decide. The reason for this is that it is not entirely clear what an interesting association rule would look like when mining for association rules in web newspapers.

When looking at the rules from this experiment, it becomes apparent that rule number 33 (Table 5) may also be considered interesting. Consider for example that there is an article discussing an event in China at time 0, then the next day a related article appears where the US President is mentioned. Finally, at time 2 an article containing military news which is related to the two previous articles appear. It is however difficult to know whether these cases are related, or just coincidental. But it gives an indication that it may in fact be possible to detect interesting temporal relationships between news items from different versions of the front page of a web newspaper.

6.3 Summary

The experiments have shown that the main problem of mining textual association rules from web newspapers is that it is difficult, if not impossible, to clearly see which rules are interesting. However, the rules found using the new document feature extraction operations can be said to make sense. Contributing to this is also that synonyms are added to words if available, and thus *head/chief* is easier to understand than only the word *head*.

When it comes to using semantic similarity for rating association rules, it is still an open question whether this can lead to good results. The reason for this is that identifying interesting rules is difficult, and it is therefore not possible to say if rules with low semantic similarity are more interesting than rules with high similarity.

Rule#	Rule	Sup	Conf	SemSim
1	{('europe/nnp',0)} → {('market/marketplace#1',1)}	0.16	0.52	0.0000
2	{('china/nnp',0)} → {('military/armed_forces#1',1)}	0.21	0.54	0.0000
3	{('russia/nnp',0)} → {('military/armed_forces#1',1)}	0.12	0.54	0.0000
4	{('iraq/nnp',0)} → {('military/armed_forces#1',1)}	0.10	0.52	0.0000
5	{('uk/nnp',0)} → {('military/armed_forces#1',1)}	0.19	0.53	0.0000
6	{('year#3',0)} → {('china/nnp',1)}	0.10	0.50	0.0000
7	{('europe/nnp',0)} → {('china/nnp',1)}	0.16	0.52	0.0000
8	{('year#3',0)} → {('europe/nnp',1)}	0.11	0.55	0.0000
9	{('commercial_enterprise/business_enterprise#2',0)} → {('eu/nnp',1)}	0.13	0.52	0.0000
10	{('uk/nnp',0)} → {('eu/nnp',1)}	0.18	0.50	0.0000
11	{('depository_financial_institution/bank#1',0)} → {('eu/nnp',2)}	0.13	0.52	0.0000
12	{('russia/nnp',0)} → {('military/armed_forces#1',2)}	0.14	0.62	0.0000
13	{('iraq/nnp',0)} → {('military/armed_forces#1',2)}	0.11	0.57	0.0000
14	{('sarkozy/nnp',0)} → {('military/armed_forces#1',2)}	0.13	0.56	0.0000
15	{('uk/nnp',0)} → {('military/armed_forces#1',2)}	0.21	0.58	0.0000
16	{('europe/nnp',0)} → {('military/armed_forces#1',1)}	0.18	0.58	0.0000
17	{('russia/nnp',0)} → {('head/chief#4',2)}	0.11	0.50	0.0000
18	{('russia/nnp',0)} → {('president_of_the_united_states/united_states_president#1',2)}	0.12	0.54	0.0000
19	{('russia/nnp',0)} → {('head/chief#4',1)}	0.11	0.50	0.0000
20	{('eu/nnp',0)} → {('military/armed_forces#1',2)}	0.21	0.55	0.0000
21	{('china/nnp' 'market/marketplace#1',0)} → {('military/armed_forces#1',1)}	0.11	0.55	0.0593
22	{('market/marketplace#1',0)} → {('military/armed_forces#1',2)}	0.23	0.59	0.0593
23	{('market/marketplace#1',0)} → {('military/armed_forces#1',1)}	0.23	0.59	0.0593
24	{('china/nnp' 'market/marketplace#1',0)} → {('military/armed_forces#1',2)}	0.12	0.59	0.0593
25	{('company#1',0)} → {('market/marketplace#1',2)}	0.15	0.50	0.0600
26	{('investor#1',1) ('uk/nnp',0)} → {('military/armed_forces#1',2)}	0.10	0.85	0.0600
27	{('investor#1' 'uk/nnp',0)} → {('military/armed_forces#1',1)}	0.12	0.72	0.0600
28	{('investor#1',0)} → {('military/armed_forces#1',1)}	0.28	0.69	0.0600
29	{('investor#1',0)} → {('military/armed_forces#1',2)}	0.22	0.55	0.0600
30	{('investor#1' 'uk/nnp',0)} → {('military/armed_forces#1',2)}	0.11	0.67	0.0600
31	{('week/hebdomad#1',0)} → {('military/armed_forces#1',1)}	0.11	0.52	0.0607
32	{('investor#1' 'president_of_the_united_states/united_states_president#1' , 0)} → {('military/armed_forces#1',1)}	0.14	0.83	0.0611
33	{('china/nnp',0) ('president_of_the_united_states/united_states_president#1' , 1)} → {('military/armed_forces#1',2)}	0.11	0.75	0.0622
34	{('china/nnp' 'president_of_the_united_states/united_states_president#1' , 0)} → {('military/armed_forces#1',1)}	0.11	0.71	0.0622
35	{('president_of_the_united_states/united_states_president#1' , 0)} → {('military/armed_forces#1',1)}	0.23	0.63	0.0622
36	{('investor#1' 'head/chief#4',0)} → {('military/armed_forces#1',1)}	0.12	0.72	0.0635
37	{('conflict/struggle#1',0)} → {('military/armed_forces#1',2)}	0.10	0.65	0.0637
38	{('year#3',0)} → {('military/armed_forces#1',1)}	0.11	0.55	0.0638
39	{('head/chief#4',0)} → {('military/armed_forces#1',1)}	0.20	0.64	0.0670
40	{('head/chief#4',0)} → {('military/armed_forces#1',2)}	0.17	0.55	0.0670
41	{('commercial_enterprise/business_enterprise#2',0)} → {('military/armed_forces#1',2)}	0.18	0.70	0.0674
42	{('occupation/business#1',0)} → {('military/armed_forces#1',2)}	0.11	0.60	0.0703
43	{('military/armed_forces#1',1) ('president_of_the_united_states/united_states_president#1' , 0)} → {('investor#1',2)}	0.14	0.62	0.0735
44	{('time#5',0)} → {('military/armed_forces#1',1)}	0.11	0.60	0.0742
45	{('time#5',0)} → {('military/armed_forces#1',2)}	0.10	0.55	0.0742
46	{('military/armed_forces#1',1) ('head/chief#4',0)} → {('investor#1',2)}	0.12	0.62	0.0784
47	{('military/armed_forces#1' 'head/chief#4',0)} → {('investor#1',2)}	0.10	0.61	0.0784
48	{('country/state#1',0)} → {('investor#1',2)}	0.12	0.62	0.0817
49	{('president_of_the_united_states/united_states_president#1' , 0)} → {('investor#1',2)}	0.19	0.53	0.0870
50	{('company#1' 'president_of_the_united_states/united_states_president#1' , 0)} → {('military/armed_forces#1',1)}	0.13	0.74	0.0873
51	{('company#1' 'president_of_the_united_states/united_states_president#1' , 0)} → {('military/armed_forces#1',2)}	0.10	0.58	0.0873
52	{('president_of_the_united_states/united_states_president#1' 'head/chief#4',0)} → {('investor#1',2)}	0.11	0.71	0.0919
53	{('head/chief#4',0)} → {('investor#1',2)}	0.17	0.55	0.0968
54	{('depository_financial_institution/bank#1',0)} → {('military/armed_forces#1',2)}	0.13	0.52	0.0973
55	{('company#1',0)} → {('military/armed_forces#1',2)}	0.16	0.53	0.1124
56	{('company#1',0)} → {('military/armed_forces#1',1)}	0.17	0.56	0.1124

Table 1. Complete Set Of Rules

Rule#	Rule
1	{('europe/nnp', 0)} → {('market/marketplace#1', 1)}
2	{('china/nnp', 0)} → {('military/armed_forces#1', 1)}
3	{('russia/nnp', 0)} → {('military/armed_forces#1', 1)}
4	{('iraq/nnp', 0)} → {('military/armed_forces#1', 1)}
5	{('uk/nnp', 0)} → {('military/armed_forces#1', 1)}

Table 2. The 5 first rules with no semantic similarity

Rule#	Rule
21	{('china/nnp' 'market/marketplace#1', 0)} → {('military/armed_forces#1', 1)}
22	{('market/marketplace#1', 0)} → {('military/armed_forces#1', 2)}
23	{('market/marketplace#1', 0)} → {('military/armed_forces#1', 1)}
24	{('china/nnp' 'market/marketplace#1', 0)} → {('military/armed_forces#1', 2)}
25	{('company#1', 0)} → {('market/marketplace#1', 2)}

Table 3. The 5 rules with the lowest semantic similarity

Rule#	Rule
52	{('president_of_the_united_states/united_states_president#1' 'head/chief#4', 0)} → {('investor#1', 2)}
53	{('head/chief#4', 0)} → {('investor#1', 2)}
54	{('depository_financial_institution/bank#1', 0)} → {('military/armed_forces#1', 2)}
55	{('company#1', 0)} → {('military/armed_forces#1', 2)}
56	{('company#1', 0)} → {('military/armed_forces#1', 1)}

Table 4. The 5 rules with the highest semantic similarity

Rule#	Rule
33	{('china/nnp', 0) ('president_of_the_united_states/united_states_president#1', 1)} → {('military/armed_forces#1', 2)}

Table 5. Potentially interesting rule

A problem that could affect the usefulness of semantic similarity, is the difficulty of assigning the correct sense to a word. An evaluation of a number of random texts from the document collection showed a precision of only about 35%, which is similar to what has been reported in previous work [1]. In addition, in some cases it was not possible to determine if the correct sense was assigned to a word. The reason for this is that the senses in WordNet are very fine-grained, and it is difficult to spot the difference (also reported by Hovy et al. [7]).

The implication of this problem to the results of this project, is that care must be taken when looking at the association rules since some of the terms may be present due to erroneous word sense disambiguation. However, many words which are disambiguated incorrectly will be filtered out during the rule mining process because their support in the document collection as a whole is too low.

One of the problems with interestingness when mining for association rules in web newspapers, is that what may seem like an interesting rule, really is a coincidence. Consider for example the rule given in the problem description, namely $\{$('Bomb', 0)$\} \rightarrow \{$('Terror', 1)$\}$. At first glance, this rule may seem interesting. But after further inspection it may become clear that the news article containing the word 'terror' is in no way related to the article containing 'bomb', instead it may relate to a totally different event and the association rule is totally coincidental.

7 Conclusions And Further Work

In this paper we have extended the previous work on mining association rules in temporal document collections by performing mining based on *semantics* as well as studying the impact of additional techniques for ranking of rules. Based on result from experiments we have illustrated the usefulness of employing semantics in this context, and shown that the impact of using semantic similarity for ranking rules is questionable at best.

Future work will go in two directions: 1) further development of appropriate metrics for rule quality, and 2) improvement of the actual rule mining, so that larger document collections can be mined as well as reducing processing time for smaller collections.

Acknowledgments The authors would like to thank Trond Øivind Eriksen and Kjell-Inge Skogstad who developed the basic ideas for mining association rules in temporal document collections and implemented the TTM Testbench.

References

1. S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 2003.
2. S. Basu, R. J. Mooney, K. V. Pasupuleti, and J. Ghosh. Evaluating the novelty of text mined rules using lexical knowledge. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, 2001.
3. S. Chakrabarti. *Mining the Web - Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, 2003.
4. M. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall, 2003.
5. R. Feldman and J. Sanger. *The Text Mining Handbook*. Cambridge, 2007.
6. J. D. Holt and S. M. Chung. Efficient mining of association rules in text databases. In *Proceedings of CIKM'99*, 1999.
7. E. H. Hovy, M. P. Marcus, M. Palmer, L. A. Ramshaw, and R. M. Weischedel. OntoNotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, 2006.
8. 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.
9. J. J. Jiang and D. W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*, 1997.
10. 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.
11. B. Lent, R. Agrawal, and R. Srikant. Discovering trends in text databases. In *Proceedings of KDD'1997*, 1997.
12. M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC '86)*, 1986.
13. 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.
14. C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 999.
15. Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of KDD'05*, 2005.
16. K. Nørvåg, K.-I. Skogstad, and T. Eriksen. Mining association rules in temporal document collections. In *Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems (ISMIS'06)*, 2006.
17. 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.
18. P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of KDD'2002*, 2002.
19. K. Toutanova and C. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 2000.
20. 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.
21. Y. Zhao, C. Zhang, and S. Zhang. Discovering interesting association rules by clustering. In *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, 2004.