Mining of Association Relations in Text

Gleb Sizov, Pinar Öztürk

Department of Computer Sciences

Norwegian University of Science and Technology

Trondheim, Norway

Abstract

We present a method that adopts ideas from association rule mining in database (DB) systems to facilitate reuse of organizational knowledge captured in textual form. Many organizations possess large collections of textual reports representing the episodic memory of an organization, e.g. medical patient records, industrial accident reports, lawsuit records and investigation reports. Effective (re)use of expert knowledge contained in these reports may increase the productivity of the organizations. Our method may support employees in discovering information in textual reports that can be useful for dealing with a new situation or writing a new report. Association rule mining is used in DB systems to discover associations between items in database records. We set out with the hypothesis that a similar approach may be used to discover implicit relations between text units in textual reports. We developed the *SmoothApriori* algorithm that finds association relations between sentences which may correspond to a cause-effect type of relation or have a more implicit nature. We evaluated the ability of SmoothApriori to restore sentences that were removed, for test purpose, from air investigation reports. SmoothApriori restored significantly more information than any of our baselines.

1 Introduction

Success of an organization is proven to highly depend on its knowledge which is generated and accumulated by its employees over years. However, unless made explicit and shareable, organizations have the risk of losing this knowledge because employees may change jobs at any time, or retire. Knowledge management (KM) deals with methods for handling knowledge in an organization. Although no single agreement upon definition of knowledge management exists, four subtasks are typically maintained to underly the KM process: (1) construction/creation, (2) storage/retrieval, (3) transfer, and (4) application [4]. This paper focuses only on the retrieval and application processes.

As it regards knowledge, there is an ongoing discussion about its definition and the different types of knowledge, which can be traced back to ancient Greek philosophers

This paper was presented at the NIK-2012 conference; see http://www.nik.no/.

[10]. We will not dwell into this discussion but rather concentrate on one certain type of knowledge which is identified in cognitive psychology and artificial intelligence as *episodic memories* [18, 12]. These memories are about concrete situations with reference to time, space and physical objects in the real world and reflect the reasoning of an expert when solving a problem. Experts document such "knowledge" to have an overview of their work and for sharing with colleagues. Such documentation might also be necessary for evidence purpose in case of legal problems, and because of governmental regulations. Consequently, many companies and institutions have large collections of textual reports documenting their organizational experience on a particular task, a client or a problem. Industrial accident reports, law suit reports, electronic patient records and investigation reports are the most intuitive examples of such documents. It has been conjectured in recent years that these reports constitute the episodic memory of organizations [7] and effective use of the knowledge contained there can save substantial time and resources. For example, accident reports can be used to identify possible risks and prevent future accidents, law suit reports constitute precedences for future cases, and patient records might help to diagnose and find an appropriate treatment for a patient with similar symptoms.

Information technology plays an indispensable role in the implementation of the KM. Modern technology allows huge storage capacity while finding relevant documents in the corporate memory and the use of knowledge contained in these remain a major challenge. In this paper we present a model for using organizations' episodic memories captured in text. We tested the model on aviation accident reports. Let us describe the task a bit more. When an accident occurs, the investigation is conducted. An expert is assigned the task of analyzing the context of the accident and writing down his/her understanding of why it happened, and possibly what could have been done in order to hinder it happening. The expert may benefit from studying past accident reports similar to the current one in some critical aspects. Past reports can be useful for finding additional details that might be relevant for the current situation and provide hypotheses for possible causes of the accident.

Traditional information retrieval (IR) systems provide techniques to retrieve documents that contain the same or similar keywords and key phrases as in a user query. The result returned by an IR system usually consists of a list of documents ranked according to the number and importance of matching keywords. Some of the modern IR systems will also highlight the matched keywords in the retrieved documents or even extract sentences that contain them. However, a problem solving user is often not interested in information similar to the query but rather information that bears an interesting relation to the query, e.g. explanation, cause, effect, additional details, solution and so on. Even though this information is likely to exist in the documents that match the query, finding the snippets of the document that capture such *associative* knowledge by manual inspection is both a time consuming task and is critically dependent on the degree of expertise of the person. This paper presents the *SmoothApriori* algorithm which automatically identifies association relations between sentences and thus allows retrieval of sentences associated with the query. The algorithm is inspired by association rules mining in database systems. Previously, association rules have been used in information retrieval to obtain relations between words and short phrases [16]. We modify the classical association rule mining algorithm, Apriori, to enable discovery of association rules between larger text units such as sentences.

The rest of the paper is organized as follows. Section 2 formally introduces the

concept of association rules. In section 3, the Apriori algorithm is described. Section 4 reviews the previous work done on association rule mining for textual data. In section 5, Apriori is extended to SmoothApriori which is able to mine sentence-based rules. Section 6 describes the experiment in which SmoothApriori is evaluated on air investigation reports. The conclusion and the future work directions are presented in section 7.

2 Association rules

An association rule is a directed relation between two sets of items, from the *antecedent* to the *consequent*. Association rules were first introduced in [1] for the purpose of discovering interesting relations between items in retail transaction "baskets" where each basket is represented as a record in a database. An example of a rule in the market domain is $\{milk, butter\} \Rightarrow \{bread, sausage\}$. The intuitive interpretation of this rules is that customers who buy milk and butter tend to buy bread and sausage as well.

Formally, let $I = \{i_1, i_2, ..., i_n\}$ be the set of all items in a certain domain. Let D be the collection of records, where each record $R \subseteq I$ is a subset of items in I. A rule has the form $A \Rightarrow B$, where $A \subset I$ and $B \subset I$ are non-overlapping itemsets, subsets of items in I. The number of possible rules grows exponentially with the number of items in I but only relatively few of them are interesting. The aim is to eliminate all uninteresting rules without missing interesting ones. Interestingness can be defined using a variety of measures. The most well-known ones are *confidence* and *support*. Support of an itemset A is the proportion of records in a database D that contain A:

$$support(A) = \frac{|\{R \in D \mid A \subseteq R\}|}{|D|} \tag{1}$$

Support of a rule $A \Rightarrow B$ is equal to the support of $A \cup B$: $support(A \cup B)$. Confidence of a rule $A \Rightarrow B$ is the proportion of records that contain A which also contain B:

$$confidence(A \Rightarrow B) = \frac{support(A \cup B)}{support(A)} \tag{2}$$

Rules are considered interesting if their support and confidence values are above predefined thresholds. The threshold values are usually determined experimentally and have direct influence on the number of rules that are considered interesting. For example, lowering the support threshold leads to generation of more rules.

When the support and confidence values are included, the rule example we had earlier about the market domain becomes $\{milk, butter\} \Rightarrow \{bread, sausage\}$ (0.02, 0.1) where 0.02 is the support and 0.1 is the confidence. The interpretation of the rule is that in 2% of all transactions customers buy milk, butter, bread and sausage together, and in 10% of transactions where customers buy milk and butter they also buy bread and sausage. In addition to support and confidence a variety of other interest measures have been proposed. A thorough study of such measures can be found in the work by Tan et al. [17]

3 Association rules mining

There are many ways to mine association rules. The most naive one is to generate all possible rules and then to apply some criteria (e.g., support and confidence) to select the most interesting ones. This straightforward approach of iterating over all possible rules and computing support and confidence is not feasible for all but very small datasets.

Therefore, many algorithms have been proposed to avoid the complete enumeration of every possible rule. Instead of generating association rules in the first place, these algorithms first produce *frequent itemsets*, i.e. itemsets with support equal or greater than the minimum support threshold. Generation of high-confidence rules from frequent itemsets is then a trivial task and can be accomplished by iterating through all possible splits of a frequent itemset and computing confidence of the splits. For description and comparison of association rule mining algorithms see Hipp et al. [11].

Our work concerns the classical frequent set generation algorithm Apriori proposed by Agrawal [2]. Being quite efficient, this algorithm is easy to implement and extend for mining rules in text. Our algorithm, SmoothApriori, is an extension of the Apriori algorithm. Therefore, we describe first the Apriori algorithm (see algorithm 1).

Algorithm 1 Structure of Apriori algorithm.

```
1:  function APRIORI(items I, records D, minsup)
2:      C_1 = {{i} | i ∈ I}                          ▷ generate 1-itemsets
3:      k = 1
4:      repeat
5:          for X ∈ C_k do
6:              for R ∈ D do
7:                  X.support = X.support + match(X, R)
8:              end for
9:          end for
10:         L_k = {X ∈ C_k | X.support ≥ minsup}
11:         k = k + 1
12:         C_k = generate-candidates(L_{k-1})
13:     until L_{k-1} ≠ ∅
14:     return ⋃_k L_k
15: end function
```

The core of the algorithm consists of three procedures: generation of candidate itemsets (lines 2 and 12), computation of support for each candidate (lines 5-9), and selection of itemsets with support greater or equal than the minimal support threshold *minsup* (line 10). To complete the algorithm it is necessary to define the functions *match* (line 7) and *generate-candidates* (line 12). In Apriori, *match* returns 1 if an itemset X a is subset of a record R, and 0 otherwise.

There are several alternatives for the candidate generation procedure. The one we used is proposed by Manilla et al. [13] and is defined as follows:

$$C'_k = \{X \cup X' \mid X, X' \in L_{k-1}, |X \cap X'| = k-2\} \quad (3)$$

$$C_k = \{X \in C'_k \mid |\{P \in L_{k-1} \mid P \subset X\}| = k\} \quad (4)$$

where equation 3 generates candidate k-itemsets by joining two frequent (k-1)-itemsets that differ only by one item and equation 4 selects k-itemsets from the candidates each of which contains k frequent (k-1)-itemsets. The main idea of this two step procedure is to eliminate candidate k-itemsets that can't possibly be frequent because they contain a non-frequent (k-1)-itemset. This idea is supported by the downward closure lemma.

Lemma 1 (Downward closure lemma) *If an itemset X is not frequent, then any itemset that contains X is guaranteed to be not frequent.*

This lemma allows to reduce the number of candidates sufficiently, making Apriori tractable for real-life datasets. In practice, the running time will very much depend on a minimal support threshold relative to the number of items, records, number of items in a record, and the sparsity of records.

4 Association rules mining in text

Association rules are mined based on the co-occurrence of items in database records, where the number of items is usually far less than the number of records and the subsets of items typically appear in multiple records. When a text collection is used instead of a database, the data should first be converted into an item-record like representation. In this representation, a record may correspond to a document, a paragraph, a sentence or any other text unit. Similarly, an item may correspond to a word, a phrase, a sentence or any piece of text smaller than a record. The decision regarding the granularity of records and items is made based on the available dataset and the task at hand.

Current research typically uses keywords as items. For example, the FACT system [8] was used to mine Reuters news articles to discover rules such as $\{Israel\} \Rightarrow \{Palestina, Iraq\}$. In the work by Ahonen et al. [3] the mined keyword-based rules are used for extraction of descriptive phrases. Chen et. al [5] have used keyword-based rules for identification of frequent phrases and then used these phrases to mine phrase-based rules. Feature generalization is another common application of keyword-based rules. In the work by Rossi et al. [15] features are generalized by merging two or more keywords that are connected by strong association relations. Feature generalization has been also investigated by Wiratunga et al. [19], where association rules transform a term-document matrix to a less sparse one by sharing values of strongly connected terms with each other, thus bringing semantically related documents closer. Information extraction techniques may be applied prior to association rule mining in order to extract textual features/items of interest from a text. An example of this approach can be found in the work by Ghani et al. [9]. The system developed by Ghani et al. [9] uses structured information extracted from web-pages. Some of the features used are address, company name, company type, income data, number of employees and so on. These features are structured and abstracted prior to association rule mining.

As far as we are aware of, existing work on mining association rules from text documents operates on a word or phrase level only. In contrast, we aim to mine rules where items in the antecedent and consequent parts are sentences. Sentences, unlike individual words and short phrases carry larger and more specific pieces of information and thus give us the possibility to obtain more meaningful and useful relations. The problem is that a larger piece of text such as a sentence rarely appears more than once in the entire collection of documents. Consequently, rule mining techniques that rely on counting identical sentences in documents would not be able to identify any rules. On the other hand, due to the variability of a natural language, syntactically different sentences (i.e. because of synonymous words and different grammatical structure) may have identical meaning. To make it even more challenging, many sentences may not have an identical counterpart even on the semantic level, but still bear some degree of similarity to other sentences. This motivated us to investigate the use of similarity between items in association rule mining process.

The idea of accommodating similarity between items in mining of association rules was first proposed by Nahm et al. [14]. Nahm developed the *SoftApriori* algorithm for addressing the problem of synonymous items at the word/phrase level. This problem does

not appear in databases because same items are usually denoted by the same word in all records, e.g. *milk* is *milk* in all its occurrences. However, words with different denotations but the same connotation are common in natural language text. For example, "Win XP", "Windows XP" and "MS Windows XP" all have the same meaning but will be counted as different items when computing support. This issue needs to be resolved in order for rules to gain enough support. SoftApriori handles such "noise" in the extracted textual features by collapsing terms with the same meaning into a single item (e.g., to "Windows XP" in the example above). To do this, the algorithm relies on the similarity between textual items which is represented as a binary value, i.e. 0 - similar or 1 - not similar. Intuitively, it merges similar items and then uses the merged ones as items in the original Apriori. It is worth mentioning that the algorithm does not do merging of items beforehand but rather handles it dynamically when computing the *match* function (line 7 in algorithm 1). Originally SoftApriori was not intended for mining sentence-based rules, but it can be used for this purpose. However, treating similarity as a binary value is a very rough approximation. In our evaluation, presented in section 6, the performance of SoftApriori is compared to our algorithm.

5 Smooth association rules

We propose a modification of the Apriori algorithm which we dubbed SmoothApriori. For computation of the support value, it makes use of the similarity between items where the actual similarity values are used instead of reducing them to binary similar/not similar values like SoftApriori [14] does. There are no specific requirements for the similarity measure used in SmoothApriori. The choice of the measure is dictated by the type of items used.

SmoothApriori follows the structure of the Apriori algorithm (see algorithm 1) but replaces the *match* function with *smooth-match*. Computation of this function involves finding a maximum weight bipartite matching between items of a candidate itemset and items in a record. The smooth-match function can be though of as the bottleneck of this matching. Formally, the task of maximum weight bipartite matching is defined as follows:

Definition 1 (Maximum weight bipartite matching) *Given a bipartite graph* $G = (A,B,E)$ *with vertices divided into disjoint sets* A *and* B *such that each edge* $e \in E$ *connects a vertex in* A *to a vertex in* B *and has a weight* $w(e)$*, find a matching (i.e. a set of edges without common vertices) with maximum weight where the weight of a matching* M *is given by* $w(M) = \sum_{e \in M} w(e)$

To compute the smooth-match function for a candidate itemset X and a record R a bipartite graph is constructed with the bipartition (X,R) where each edge e connects an item $x \in X$ to an item $r \in R$ has a weight equal to the similarity between x and r: $w(e) = similarity(x,r)$. Given the maximum weight bipartite matching M for this graph, the smooth-match function is defined as follows:

$$\text{smooth-match}(X,R) = \min_{e \in M} w(e) \tag{5}$$

For example, consider the bipartite graph in figure 1 where the maximum weight bipartite matching consists of edges with weights $similarity(x_1,r_2) = 0.8$ and $similarity(x_2,r_1) = 0.7$. The sum, 1.5, is greater than for any other alternative matching in this graph. The value for the smooth-match function is the minimum similarity in this matching, that is 0.7.

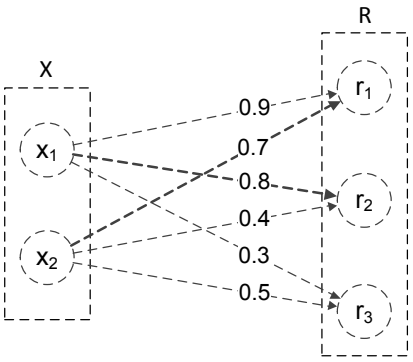

Figure 1: Maximum weight bipartite matching between items in a candidate itemset X and items in a record R.

The support computed by SmoothApriori is referred to as *smooth-support*. Analogous to the support in the Apriori algorithm (line 7 in algorithm 1), it is the sum of smooth-match values with all the records divided by the number of records. For example, given an itemset X and one record R from figure 1 with the smooth-match 0.7, the smooth-support will be $0.7/1 = 0.7$. If we add another record with the smooth-match 0.5, the smooth-support becomes the average of these values $(0.5+0.7)/2 = 0.6$. *Smooth-confidence* is calculated the same way as shown in equation 2 but using smooth-support instead of support values.

Downward closure lemma (see lemma 1), which makes Apriori efficient, applies for SmoothApriori as well.

Lemma 2 (Smooth-support lemma) *If an itemset X has smooth-support below minsup, then any itemset that contains X is guaranteed to have smooth-support below minsup.*

The correctness of the smooth-support lemma directly follows from the correctness of the smooth-match lemma.

Lemma 3 (Smooth-match lemma) *For any itemset X, an itemset $Y \subset X$ and a record R the following is always true: smooth-match$(Y,R) \geq$ smooth-match(X,R)*

Proof 1 *By definition smooth-match takes the minimum similarity of all item pairs matched by maximum weighted bipartite matching. Let us consider an item $i \in X$ which has the minimum similarity with a matched item from R of all items in X. Since $Y \subset X$, i is either in Y or in $X \setminus Y$. If $i \in Y$, then i has the lowest match similarity of all items in Y as well, thus smooth-match$(Y,R) =$ smooth-match(X,R). If $i \in X \setminus Y$ then all items in Y have greater match similarity than i, thus smooth-match$(Y,R) >$ smooth-match(X,R).*

The complexity of the SmoothApriori algorithm is greater than Apriori because of the cost of maximum bipartite matching which can be computed with the Hungarian algorithm with the complexity $O(V^3)$ where $V = max(|X|,|R|)$. In addition, similarity between items is calculated beforehand.

6 Evaluation

In this section we describe experimental evaluation of SmoothApriori compared to SoftApriori and two baseline algorithms

Dataset

In our experiment we used air investigation reports from the Transportation Safety Board of Canada[1]. Each report in this collection documents an aircraft accident including details about what, where and when it happened, qualification and behaviour of the pilot, weather conditions, aircraft technical details, communication with the controllers on the ground, possible causes, risk factors and the safety measures taken after the accident.

In our experiment we extract sentences only from "Findings as to Risk" section, which on average consists of 4 sentence. Only 208 reports of 650 contain this section and thus are used in the experiment. The reports were downloaded from Transportation Board of Canada website as html documents. The preprocessing steps applied for each document are (1) extraction of text and structure from html, (2) splitting the text into words and sentences (3) stemming of words. All the preprocessing steps were accomplished using ANNIE components in GATE NLP platform [6].

Evaluation task

The purpose of the evaluation task is to estimate the quality of association rules produced SmoothApriori. The quality should determine how meaningful is a rule from the domain perspective and the usefulness of it in some user task. For our evaluation we assume the accident analysis and report authoring task. Given the initial description of the accident, the system discovers rules which may contribute to the explanation of the underlying cause or may point to some critical information lacking from the initially available situation description. Ideally, domain experts should determine to what extent the produced rules are able to do so. Although we have manual evaluation in our plans, currently we opted for automatic evaluation.

The idea behind our automatic evaluation procedure is to take a report and remove half of the sentences from it. Then use the system to discover rules of which antecedents are subsets of the remaining sentences. Finally match the consequent sentences of these rules with the removed sentences. The closer the match is between the consequences in the rule set and the removed sentences, the higher the quality of the discovered rules will be. The evaluation workflow follows a 10-fold cross validation procedure where each tested system (see section 6) is evaluated on each of the 10 bins using the following procedure:

1. Mark one bin as a test set and use the remaining 9 bins as a training set.
2. Feed the training set to the tested system.
3. For each report in the test set:
    - 3.1. Randomly (order of sentences is not preserved) split the report into two equally-sized sets of sentences: *available sentences*, *missing sentences*.
    - 3.2. Provide the available sentences to the tested system as a *query*.
    - 3.3. The tested system returns a set of sentences, which we refer to as *restored sentences*.
    - 3.4. Match restored sentences with missing sentences by maximum weighted bipartite matching.
    - 3.5. Calculate recall, precision and F-measure scores for the match.

[1] Air Investigation Reports are available at `http://goo.gl/k9mMV`

The same training-testing and available-missing splits are used when testing all the systems. In step 3.4, restored sentences are matched with missing sentences using the approach similar to *smooth-support* (see section 5 for details), but instead of the minimum, the sum of similarities for all matched restored and missing sentences is taken. In step 3.5, this sum is divided by the number of missing and restored sentences to calculate recall and precision respectively. F-measure is computed as the harmonic mean of precision and recall.

Textual similarity

SoftApriori and SmoothApriori require similarity between sentences for mining association rules, i.e. edge weights in figure 1. In addition, evaluation procedure outlined in section 6 also requires similarity to match restored and missing sentences. We implemented a textual similarity based on overlap between sets of stemmed terms extracted from sentences, excluding stop words. The reason why we use such a simple form of textual similarity is to avoid uncontrollable interactions between association rule mining algorithms and the details of similarity measures such as term weighting, parsing accuracy, quality of utilized lexical resources etc. This makes the analysis and interpretation of the experimental results a less dubious task. In our future work we plan to test SmoothApriori with more sophisticated similarity measures.

Test systems

The following four systems were implemented and compared:

SmoothApriori-based system Antecedents of rules generated by SmoothApriori (minimum support and confidence thresholds 0.01 and 0.3)from the training set are matched with available sentences of a test report using *smooth-match* function. If the match is greater than 0.3, the consequents of the rules are retained as restored sentences.

SoftApriori-based system The same as for SmoothApriori-based system but rules generated by SoftApriori (minimum support and confidence thresholds 0.01 and 0.1) are used. The similarity threshold used in the mining of rules was set to 0.5.

Similarity-based baseline For each available sentence in a test report the most similar sentence from the training set is selected.

Random baseline Sentences are selected randomly from the training set. The number of sentences selected is half the size of a random report.

All the thresholds are chosen manually to maximize the performance of the systems on one of the cross validation bins.

Results and discussion

The results are summarized in table 1. Overall, the scores are low and the variance is high due to the difficulty of the task. The random baseline scored under 0.1% with relative variance more than 100%, which indicates that the evaluation measure does not allow to increase scores by chance. Similarity-based baseline is under 1%, significantly lower than any of the association rule mining systems. This can be explained by peculiarities of the ”Findings as to Risk” section that briefly enumerates findings without creating

System	Precision	Recall	F-score
Random baseline	0.0007 ± 0.0001	0.0011 ± 0.0002	$\mathbf{0.0009} \pm 0.0001$
Similarity baseline	0.0099 ± 0.0024	0.0068 ± 0.0010	$\mathbf{0.0078} \pm 0.0013$
SoftApriori	0.0244 ± 0.0111	0.0186 ± 0.0059	$\mathbf{0.0194} \pm 0.0068$
SmoothApriori	0.0518 ± 0.0142	0.0484 ± 0.0118	$\mathbf{0.0455} \pm 0.0103$

Table 1: Evaluation results, mean $\pm$ variance.

groups of related sentences. SoftApriori scored around 2% and SmoothApriori more than doubles this score. The p-value for paired t-test of SoftApriori and SmoothApriori scores is 0.00008, which makes the difference in scores statistically significant.

In order to see how meaningful the rules generated by SmoothApriori are, we printed out some of the rules with top confidence.

1. Typically, flight crews receive only limited training in stall recognition and recovery, where recovery is initiated at the first indication of a stall. $\Longrightarrow$ Such training does not allow pilots to become familiar with natural stall symptoms, such as buffet, or allow for practise in recovering from a full aerodynamic stall.

2. Transport Canada was not aware of the proposed ferry flight and therefore could not intervene. $\Longrightarrow$ This increases the time that aircraft are at risk of collision.

3. The pilot not flying was not wearing protective headgear. This increases the time that aircraft are at risk of collision. $\Longrightarrow$ This increased the risk of injury and, in this case, the risk of drowning.

4. Therefore, search and rescue efforts did not begin until one hour after the flight's planned estimated time of arrival. $\Longrightarrow$ There was no emergency locator transmitter (ELT) signal received.

5. The air traffic control environment has no standard, reliable method to alert controllers that new weather sequences have been issued. $\Longrightarrow$ Consequently, controllers may not be aware of new weather information that should be passed to flight crew.

6. Therefore, search and rescue efforts did not begin until one hour after the flight's planned estimated time of arrival. $\Longrightarrow$ The emergency locator transmitter signal was not detected, primarily because the antenna had been broken during the accident.

7. The company did not provide an adequate level of supervision and allowed the flight to depart without an autopilot. $\Longrightarrow$ The company operations manual did not reflect current company procedures.

Many of the top confidence rules automatically generated by SmoothApriori are indeed quite interesting and make sense. Some rules, e.g. 1 and 5, connect sentences that follow each other in the same report. In general, subsequent sentences are often related to maintain coherence in the discourse. Naturally, some of these relations go beyond coherence and indicate interesting association relations. The fact that the algorithm was able to discover such relations is a positive indicator of the rules' quality, especially considering that the algorithm was unaware of the original sentence order in the report.

Another observation is that many rules, e.g 2, 3, 4, 6, contain sentences that are not similar syntactically but have a meaningful semantic relation in a domain-specific context.

This is the desired effect of applying association rule mining to text and the one that discriminates it from a similarity-based approach.

Almost all the rules with high confidence are one-to-one rules despite the fact that SmoothApriori is able to discover many-to-many rules. Only rule 3 contains more than one sentence in it's antecedent. In this regard, we may want to encourage many-to-many rules by modifying the interest measure or switch to mining of one-to-one or many-to-one rules, thus substantially reducing the computational cost.

7 Conclusion and future work

We have introduced SmoothApriori algorithm for mining association rules from text. SmoothApriori is based on the idea of using similarity between items in association rule mining. Compared to previously proposed SoftApriori algorithm, which also makes use of similarity, our algorithm is able to utilize similarity values directly rather than reducing them to binary similar/not similar values. Both algorithm were implemented and tested in the proposed evaluation task, where rules generated by the algorithms are used to restore missing sentences in reports. The results show significantly better results for SmoothApriori. When inspected manually, rules generated by the algorithm seem to be meaningful and lead to interesting observations.

We plan to continue our work in development of association rule mining algorithms for textual data and their applications in knowledge management. In particular, we would like to experiment with a variety of interest and similarity measures available. The algorithm itself needs optimization in order to handle large itemsets, which is vital for large documents. Another promising direction is visualization of rules. Post-processing, such as clustering and redundancy removal, might be necessary to avoid information overload when presenting rules to the user.

References

[1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of the ACM SIGMOD international conference on Management of data*, SIGMOD '93, pages 207–216. ACM, 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proc. of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499. Morgan Kaufmann, 1994.

[3] H. Ahonen, O. Heinonen, M. Klemettinen, and A. I. Verkamo. Applying data mining techniques for descriptive phrase extraction in digital document collections. In *Proc. of the IEEE Forum on Research and Technology Advances in Digital Libraries*, pages 2–11, 1998.

[4] M. Alavi and D. Leidner. Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS quarterly*, pages 107–136, 2001.

[5] X. Chen. Personalized knowledge discovery: mining novel association rules from text. In *Proc. of the 6th SIAM International Conference on Data Mining*, pages 588–592. SIAM, 2006.

[6] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Annual Meeting of the ACL*, 2002.

[7] O. El Sawy, G. Gomes, and M. Gonzalez. Preserving institutional memory: the management of history as an organizational resource. In *Academy of Management Best Paper Proceedings*, volume 37, pages 118–122, 1986.

[8] R. Feldman. Mining associations in text in the presence of background knowledge. In *Proc. of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96*, pages 343–346, 1996.

[9] R. Ghani, R. Jones, D. Mladenic, K. Nigam, and S. Slattery. Data mining on symbolic knowledge extracted from the web. In *Workshop on Text Mining at the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 56, 2000.

[10] T. Hellström and S. Raman. The commodification of knowledge about knowledge: Knowledge management and the reification of epistemology. *Social Epistemology*, 15(3):139–154, 2001.

[11] J. Hipp, U. Güntzer, and G. Nakhaeizadeh. Algorithms for association rule mining – a general survey and comparison. *SIGKDD Explor. Newsl.*, 2(1):58–64, 2000.

[12] J. Kolodner. Towards an understanding of the role of experience in the evolution from novice to expert. *International Journal of Man-Machine Studies*, 19(5):497–518, 1983.

[13] H. Mannila, H. Toivonen, and I. A. Verkamo. Efficient algorithms for discovering association rules. In *AAAI Workshop on Knowledge Discovery in Databases (KDD-94)*, pages 181–192, Seattle, Washington, 1994. AAAI Press.

[14] U. Y. Nahm and R. J. Mooney. Mining soft-matching association rules. In *Proc. of the 11th international conference on Information and knowledge management*, CIKM '02, pages 681–683, 2002.

[15] R. Rossi and S. Rezende. Generating features from textual documents through association rules, 2011.

[16] M. Song, I. Song, X. Hu, and R. Allen. Integration of association rules and ontologies for semantic query expansion. *Data & Knowledge Engineering*, 63(1):63–75, 2007.

[17] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right objective measure for association analysis. *Information Systems*, 29(4):293–313, 2004.

[18] E. Tulving. Precis of elements of episodic memory. *Behavioral and Brain Sciences*, 7(2):223–68, 1984.

[19] N. Wiratunga, I. Koychev, and S. Massie. Feature selection and generalisation for retrieval of textual cases. In *Advances in Case-Based Reasoning, 7th European Conference*, pages 806–820. Springer, 2004.