Discovery of Relations among Scientific Articles using Association Rule Mining

Thanaruk Theeramunkong and Kritsada Sriphaew***
Sirindhon International Institute of Technology, Thammasat University
Tel: 02-5013505..20 ext. 2004 Fax: 02-5013505..20 ext. 2001
*Email: *thanaruk@siit.tu.ac.th, **kong@sit.tu.ac.th*

ABSTRACT – Relationship among technical documents is useful information but often hidden in a large amount of textual data. This paper presents a method to discover relations among such scientific publications using association rule mining approach. Meaningful relations among articles are extracted with the process of frequent itemset mining. As document-term representation, unigram and bigram together with binary and term-frequency schemes are investigated and compared. To measure the reliability of discovered patterns, the paper originally proposes a so-called citation matrix. The proposed method is evaluated using more than 10,000 articles from a research publication database. The experimental results show that document representations using bigrams performs better than those using unigrams with a gap of 3.7%-43.4%. The results show that term frequency performs better than binary weight for unigram, but reverse for bigram.

KEY WORDS -- Data Mining, Association Rule Mining, Document Relationship

1. Introduction

In recent years, explosive growth in research publication has made researchers hard to follow the state of the art in their area of interest. The large volume of information brings about serious hinderance for researchers to position their own work against existing works, or to find relationship (or connection) between them [1,2]. Although the publication of each work may include a list of related articles as its reference, it is still impossible to include all related works due to either intentional reasons (e.g. limitation of paper length) or unintentional reasons (e.g. naively unknown). Enormous meaningful connections that permeate the literature may remain hidden.

To find these meaningful connections, some previous approaches can be applied. As the naive and simplest approach, information retrieval (IR) can be used to discover related articles beyond those listed in the reference by exploiting an index of words/terms in articles [3]. Basically IR methods work under a so-called interactive mode in the sense that one needs to provide a set of keywords or a whole document in order to obtain an ordered list of relevant articles to those keywords or that document. IR applications include search engines (e.g. Google, MSN and Yahoo) that support their users to find a set of online texts via given keywords or a document.

In contrast to IR, both text categorization and clustering (for short, TC) passively arrange related articles into a set of groups, even different in being supervised or unsupervised approach. A connection between two articles can be assumed if they are associated in the same group. Unfortunately, most of traditional TC methods deal with a small number of classes/groups each of which contains a large number of articles. Therefore, articles in the same class are related to each other with quite broad relation. To gain more detailed connection, one can increase the number of groups. However, there are few studies on handling a large number of classes and none of them intentionally focuses on connection among documents in a class. Besides utilizing the meaningful content of articles, a number of approaches have been proposed to exploit structural information of documents, including links and their surrounding texts as clues for this generation. Growing from different society, known as literature-based discovery, the approach of discovering new relations within a bibliographic database have become popular in medical-related fields since 1986 by Swanson [4]. As a content-based approach with manual and/or semi-automatic process, a set of topical words or terms are extracted as concepts and then utilized to find a connection among two separate arguments. Due to the simplicity and practicality of this approach, it was used in several areas by its succeeding works [5].

While some approaches were successfully applied to obtain topical related papers, they are not fully automated with a lot of labor intensive tasks. Furthermore, suppose we get a set of discovered relations, it is not an easy task to evaluate them. Towards these problems, this paper presents an

automated process to discover relations in technical (or research) publication using association rule mining approach. Imitating the process of frequent itemset mining, meaningful relations among articles are extracted. Two versions of document-term representation, namely binary and term-frequency, are investigated and compared. Originally a so-called citation matrix is proposed to measure the reliability of discovered patterns. In the rest, Section 2 presents a traditional method for frequent itemset mining and its extension to a more general model, the vector space model. In Section 3, a number of document representation schemes and discovered patterns are described. Section 4 displays a novel evaluation method via a newly proposed reliability measure. A number of experiment results are given in Section 5. Finally, a conclusion is made in Section 6.

2. Universal Frequent Itemset Mining

document	terms
d_1	t_1, t_2, t_3, t_4
d_2	t_1, t_2, t_3, t_4
d_3	t_2, t_3
d_4	t_2, t_3, t_4

term	documents
t_1	d_1, d_2
t_2	d_1, d_2, d_3, d_4
t_3	d_1, d_2, d_3, d_4
t_4	d_1, d_2, d_4

Figure 1. Doc-Term orientation (left) and Term-Doc orientation (right).

In the past, association rule mining (ARM) was well-known as a process to find co-occurrences (frequent patterns) in a database. As a prominent technique in data mining, it was shown to be useful in various applications such as market basket analysis, fraud detection, data classification, etc. In the ARM process, frequent itemset mining (FIM) is the most essential task to find frequently occurred itemsets from the transactional database. The traditional transactional database used for FIM is presented in the term of item existences in the transaction. Although most of works on ARM deal with the traditional transactional database, there are some attempts to extend the weights for items or transactions in the database, called weighted association rule mining [6,7]. The items or transactions are independently weighted regarding to which type of discovered rules we would like to find. The higher weighted items or transactions will obtain higher priority for user interests. However, this approach gives a fixed weight to each item regardless of the transaction that such item occurs. This approach does not match with our application. Therefore, this section introduces a general concept extended from traditional FIM concept to mine frequent itemsets on the database with weighted item-transaction value named universal frequent itemset mining. Figure 1 shows two possible representation of database. Using different

	d_1	d_2	d_3	d_4
t_1	1	1	0	0
t_2	1	1	1	1
t_3	1	1	1	1
t_4	1	1	0	1

	d_1	d_2	d_3	d_4
t_1	4	2	0	0
t_2	4	2	4	1
t_3	2	4	2	2
t_4	2	4	0	1

Figure 2. Boolean-valued (left) and Real-valued (right) databases.

orientations of document-term database as an input for FIM, different kinds of knowledge will be discovered. With the first orientation, the discovered frequent itemset is a set of highly co-occurred terms in the document. Based on this, some text mining approaches were proposed to extract related terms from a bulb of documents [8]. With the second orientation, the discovered frequent itemset is prominently changed to be a set of documents which commonly share a number of terms. The discovered results can be assumed as a content-based relation among documents where the relation is introduced by coincident terms. This novelty was overlooked from text mining area and firstly focused in our previous work [9]. In this work, we further generalize the the transactional database with boolean values to any real values. Figure 2 shows the boolean-valued and the real-valued database. Here, the real value indicates a weight of an attribute in the transaction, e.g., a function of how often the attribute appears in the transaction, or (perhaps) the relative frequency of that attribute in the overall set of transactions. In the field of text processing, the weight can be defined in the form of vector space model (VSM), introduced by Salton [10]. In this case, such weight is defined by the term frequency *tf* of a term in the document. Note that in this work, transactions correspond to terms and items correspond to documents. Therefore, we use the term ``docset'' (document set) for itemset in ARM.

Unlike most FIM works on Boolean-valued database, this work addresses to mine frequent itemsets from a real-valued database. In the task of mining frequent itemsets, minimum support (*minsup*), a user-specified threshold, is used to filter out the itemsets of which their supports lower than this threshold, considered as infrequent itemsets. Traditionally, the support of an itemset is defined by a percentage of the number of transactions in which that itemset occurs as a subset to the total number of all transactions in a database. Extended to real-valued database, the conventional definition of support has to be generalized to take item weights into consideration instead of only item existences. To this end, the generalized support is defined as shown below.

Definition 2
Let D be a set of items where $D = \{d_1, d_2, ..., d_m\}$, and T be a set of transactions identifier (tid) where $T = \{t_1, t_2, ..., t_n\}$. Let $w(d_i,t_j)$ represent a weight

between an item d_i and a term t_j. A subset of D is called an itemset where a subset of T is called a tidset. Assume that an itemset $X = \{x_1, x_2, ..., x_k\} \subset D$ with k items, so-called k-itemset. The generalized support of X is defined as

$$sup(X) = \frac{\sum_{j-1}^{n} min_{i-1}^{k} w(x_i,t_j)}{\sum_{j-1}^{n} max_{i-1}^{m} w(d_i,t_j)}.$$

This generalized support preserves two closure properties of itemsets [11], i.e., downward closure property ("all subsets of a frequent itemset are also frequent"), and upward closure property ("all supersets of an infrequent itemset are also infrequent"). So far these properties have been applied in most existing FIM algorithms to reduce large computational time. Besides support, a so-called confidence is used for generating confident association rules. The confidence is left since it is out of scope of this work.

Using the databases in Figure 2, the calculation of the generalized support is explained for our focused task, document relation mining. D is a set of documents where $D = \{d_1, d_2, d_3, d_4\}$ and T is a set of terms where $T = \{t_1, t_2, t_3, t_4\}$. The docsets and their supports, for binary and non-binary databases, can be extracted as shown in Figure 3. Note that the closure properties of itemsets are still held, e.g., $sup(d_2) \geq sup(d_1d_2d_3)$, if $\{d_1d_2d_3\}$ is frequent then $\{d_1\}$ is also frequent, and if $\{d_2\}$ is infrequent then $\{d_1d_2d_3\}$ is also infrequent. Furthermore, the same results can certainly be achieved from mining on transactional database using traditional definition of support.

docset	support	
	Binary	VSM
$\{d_1\}$	3/3	10/12
$\{d_2\}$	3/3	10/12
$\{d_3\}$	2/3	6/12
$\{d_4\}$	2/3	3/12

docset	support	
	Binary	VSM
$\{d_1d_2\}$	3/3	6/12
$\{d_1d_3\}$	2/3	6/12
$\{d_1d_4\}$	2/3	3/12
$\{d_2d_3\}$	2/3	4/12
$\{d_2d_4\}$	2/3	3/12
$\{d_3d_4\}$	2/3	3/12

docset	support	
	Binary	VSM
$\{d_1d_2d_3\}$	2/3	4/12
$\{d_1d_2d_4\}$	2/3	3/12
$\{d_2d_3d_4\}$	2/3	3/12
$\{d_1d_2d_3d_4\}$	2/3	3/12

Figure 3. Docsets and their supports: binary and VSM models (the database in Figure 1).

3. Document Representation

A number of preprocessing techniques have been proposed such as indexing, mining, retrieval, classification and so forth. In most traditional text processing applications including information retrieval and text classification, it was showed that a bag of individual words alone is not good enough for representing the content of a text. Several enhancements have been proposed to generate more suitable representations. There are several available definitions of terms and their weights. In this section, three term definition schemes, i.e., *n*-gram, stemming, and stopwords excluding, and a term weighting scheme are described. Basically, the first three schemes concern with the attributes of term representation which form a document space while the last scheme involves with a set of correspond values of attributes in a document space.

As previously mentioned, it was well-known that single words in a text, called unigram (*1*-gram), may not be good enough to represent semantics of the text due to its ambiguity. Towards this problem, a higher *n*-gram can be applied. In *n*-gram, a term is defined by a set of any consecutive *n* words. Representing a document by *n*-grams makes us possible to handle compound nouns and hence reduces semantic ambiguity of words. For example, given a part of text ".. data mining and artificial intelligence ..", a set of unigrams (*1*-grams), say "data", "mining", "and", "artificial" and "intelligence" can be extracted. By bigram representation (*2*-grams), the following terms will be obtained i.e., "data mining", "mining and", "and artificial" and "artificial intelligence". Note that it is possible to apply a higher-gram for defining a term, but the exponential growth of the number of terms may cause a problem for mining process. Therefore, only unigram and bigram representations are preliminary taken into account.

As the second scheme, stemming is applied to reduce a variety of words due to their morphological change. For example, the words "leads", "leading", "leader" and "leadership" can be stemmed to their common root, that is "lead". With different environments and purposes in previous works on IR and TC, there is an argument in performance improvement when applying stemming for text pre-processing. To clarify this based on the purpose of this work, we therefore investigate this scheme by experiments. As the third scheme, a list of stopwords is applied to eliminate words which do not carry meaning in natural language but may frequently occur in documents. Examples of stopwords are "a", "the", "and", "to" and so on. By excluding stopwords defined in a standard stopword list [10], several works reported significant improvement of their tasks.

In addition to the above three schemes that involve how to define a set of terms, term weighting scheme is a process to set how important a term is in a document. Previously stated in Section 2, a document will be represented by a vector each element of which is a weight indicating importance of a word in the document. Term weighting scheme can be divided into two main types, i.e., binary weighting and non-binary weighting. Binary weighting is the most naïve method that indicates whether a term exists or not in the document exhibited by binary values. Another type of term weighting scheme is non-binary weighting which is proposed in several works. Basically, a commonly-used non-binary weighting that is successfully applied in many text

application areas is term-frequency (*tf*) which is a measure of how often a term found in a specific document.

The combinations of previously mentioned schemes produce different sets of document representation which are needed to be explored. By assumption, some schemes may effect to other schemes which result in different characteristics of discovered knowledge. To study this, various document representations are then generated and investigated.

4. Reliability Measure and Evaluation

To investigate efficiency and effectiveness of the proposed method in discovering document relationship, we explore an interesting and useful applications called retrieving citing/cited publication. To this end, the discovered docsets (document relationship) are compared to citing and cited information in a scientific publication database. In this work, citations resided in a publication database are used to construct the criteria for evaluating the discovered patterns. The citations usually appear in scientific research publications to refer the related publications with some other works. Anyway, some may argue that citations are a kind of fragmented information which their authors would to exhibit the relationship of their works with some previous works within their narrow specialities (not cover all related publications). Some may discuss that the author of an article cites another article because of not their related contents but some other purposes, such as paying homage to pioneers, authenticating data or even notification of unrelated works. However, in this work, the citations in a paper are assume to relate topically. For the purpose of evaluation, we propose a novel concept of citation matrix, which is used for indicating the reliability of discovered patterns based on citations.

4.1 Citation Matrix

The connection between an article and one of its cited articles can be viewed as a citation path. A citation path expresses only direct citation of an article to one of its direct citations. To extend the concept of the citation path to cover indirect citations, the transitive function can be applied on a so-called citation network. For example given a set of articles $\{d_1, d_2, d_3, d_4\} \in D$ and assume that d_1 cites to d_2, d_2 cites to d_3 and d_4 cites to d_3. In this scenario, there is an indirect citation path from d_1 to d_3 with 2 hops (and vice versa) and an indirect path from d_1 to d_4 with 3 hops (and vice versa). To describe these citations, a so-called *n*-th order citation is defined as follows.

For $x,y \in D$, y is the *n*-th order citation of x if there is at least one citation path from x to y via *n* hop(s). In the same way, x is also called the *n*-th order citation of y. In the above example, the first order citation exists between the document pairs (d_1,d_2), (d_2,d_3) and (d_3,d_4), the second order citation exists between the document pairs (d_1,d_3) and (d_2,d_4), and the third order citation exists between the document pair (d_1,d_4). In order to make an evaluation based on citation paths, a so-called *n*-th order accumulative citation matrix (for short, the *n*-th order acm) is introduced. The *n*-th order acm illustrates the citation paths between each document as follows.

The *n*-th order acm is a matrix with the size of $m \times m$ where m is the number of distinct documents. Assume that $x,y \in D$, $\delta^{(i)}$ is defined on the relation between x and y, $\delta^{(i)}(x,y) = 1$ when x is the *n*-th order citation of y where $j \leq i$ otherwise $\delta^{(i)}(x,y) = 0$. Note that $\delta^{(i)}(x,y) = \delta^{(i)}(y,x)$ and $\delta^{(i)}(x,x) = 0$. In our example, the first, second and third order acm's can be created as shown in Figure 4. Although the citation information can be extracted using the NLP techniques, but in this work used the information of hyperlinks between reference sections of one document to other documents as previously extracted by the ACM Digital Library and highly accurate to construct *n*-th order acm's.

doc.	d_1	d_2	d_3	d_4
d_1	0	1	0	0
d_2	1	0	1	0
d_3	0	1	0	1
d_4	0	0	1	0

doc.	d_1	d_2	d_3	d_4
d_1	0	1	1	0
d_2	1	0	1	1
d_3	1	1	0	1
d_4	0	1	1	0

doc.	d_1	d_2	d_3	d_4
d_1	0	1	1	1
d_2	1	0	1	1
d_3	1	1	0	1
d_4	1	1	1	0

Figure 4. The first, second and third order accumulative citation matrices (left to right).

4.2 Reliability Measure

This section defines a so-called reliability measure that can be used for evaluating the usefulness of the discovered patterns (docsets). Comparing a docset with the *n*-th order acm in order to show the reliability on *n*-th level of citations, the reliability score of the docset can be defined as follows. For a docset $X \subseteq D$,

$$S^{(i)}(X) = max_{x \in X} \left(\sum_{d \in X} \delta^{(i)}(x,d) \right)$$

The maximum score of $S^{(i)}(X)$ equals to the value of the length of such itemset subtracting with one ($|X|$-1). Naturally, the discovered patterns (docsets) may have various lengths. It is necessary to evaluate the discovered docsets F according to their lengths. To this end, the docsets are arranged into k groups based on their lengths, say F_1, F_2, F_3, ..., F_k. Here, F_i denotes a set of frequent docsets of the length i and the number of frequent itemsets with length k is denoted by $|F_k|$. Furthermore, F_1 is excluded since it is trivial.

To form a combined reliability score for a set of discovered docsets, it is necessary to consider the length of each docset as a weight for

representing difficulty to achieve the docset completely. Intuitively, the score of docsets with a longer length should be higher since it is more difficult to reach their maximum scores than those with a shorter length. That is, each score has to be calibrated by the difficulty of the patterns. In this work, we simply apply the length of a docset as its difficulty. That is, the docsets with the length of two will have the lowest difficulties. Hence the reliability score of a set of discovered docsets defines as:

$$\mathcal{R}^{(i)}(\mathcal{F}) = \frac{\sum_{X \in \mathcal{F}} \mathcal{S}^{(i)}(X)}{\sum_{j=2}^{k}((j-1) \times |\mathcal{F}_j|)}$$

where k is the maximum length of discovered frequent docsets. The $R^{(i)}(F)$ is equal to 1 when all discovered docsets achieve their maximum scores. In the other word, all documents in each docset cite to/from all other documents in the same set, and hence the discovered docset will have the most reliability based on n-th order acm. In the above example, if the discovered docsets F are $\{\{d_1d_2\}, \{d_1d_3\},\{d_2d_3\},\{d_1d_2d_3\}\}$. The reliability score is:

$$\mathcal{R}^{(1)}(\mathcal{F}) = \frac{\mathcal{S}^{(1)}(\{d_1d_2\}) + \mathcal{S}^{(1)}(\{d_1d_3\}) + \mathcal{S}^{(1)}(\{d_2d_3\}) + \mathcal{S}^{(1)}(\{d_1d_2d_3\})}{(1 \times 3) + (2 \times 1)}$$

$$= \frac{1+0+1+2}{5} = 0.8$$

5. Experiments

Two versions of n-gram, i.e., unigram and bigram, are considered. In the unigram, each single individual word found in the corpus are counted and used as transactions. This representation ignores the sequence in which the words occur and is based on the statistic about single words in isolation. In the bigram, two consecutive words are combined and used as transactions. By adapting stemming and stopwords removing, various document representations can be generated. The term weighting could be binary weight (term existence), or non-binary weight (in this experiment, it is *tf*). The conditions of stemming/nonstemming and whole terms/stopword removal are investigated. Ranking discovered docsets by their descending supports, the discovered docsets with at least two items (discarding one-length frequent patterns) is selected. To draw the trends of reliability when the discovered knowledge increases, the number of selected discovered knowledge is varied from 1000, 5000, 10000, 50000 and 100000. As our mining engine, the FP-tree algorithm, first introduced in [12], is applied with a modification of array-based technique to mine frequent itemsets. It is one of the best algorithms against other frequent itemset mining algorithms in various environments. The algorithm is modified to deal with mining frequent itemsets on VSM as described in Section 2.

5.1 Test Collection

There is no gold standard corpus, which is coincident with the objective of our approach, available as a benchmark for evaluation. We therefore need to construct the corpus by ourselves using the following reasonable method. A corpus, which is used as a test collections to find the semantic associations, is the research publications retrieved from the Web. We collected the research publications from ACM Digital Library. A corpus is retrieved by the following steps. Three classes of CCS (Computing Classification System); B:Hardware, E:Data and J:Computer, are supplied as three search keywords. In each class, top 200 articles in the PDF format and their information pages in HTML format are collected as seeds. The links of referenced publications appearing in the seeds are then crawled to gather those documents, and added them to the set of seeds. After three iterations, totally 10,825 research articles are collected and used as a corpus for our experiments. All publications are converted the ASCII text format. Since we will evaluate the discovered knowledge with the citations, the citation information resides in each text is subsequently removed by using both automatic detection in searching lexical cues such as “Reference”, “References”, “REFERENCES”, “Bibliography” and “BIBLIOGRAPHY” and manual detection. Moreover, the information pages which were collected during corpus construction are used to build citation matrix for evaluation. From this dataset, the probabilities that a document pair will hold a relation $\delta^{(i)}$ are 1.18×10^{-4}, 2.02×10^{-3}, 1.86×10^{-2} for the first, second and third order acm's, respectively.

Even removing stopwords, some extracted terms may not be important and produce a large vector. Terms with too low frequencies are assumed to be trivial. In this work, the terms with their frequencies lower than 3 (the threshold) are considered as statistically trivial and thus pruned. The number of terms is reduced approximately 80%-90%. In the next section, the experimental results are shown with various document representations, encoded by 4 digits. The first digit represents N-grams scheme, e.g. binary (“B”) or unigram (“U”). The second and third digits represent stemming and stopword removal schemes respectively where “X” is ‘not use’ but “O” means ‘use’. The fourth digit indicates the weighting scheme, e.g. binary (“B”) or *tf* (“T”). For example, “UXOT” means the dataset in which its document representation is unigram, no stemming, use stopword removal and *tf* term weighting.

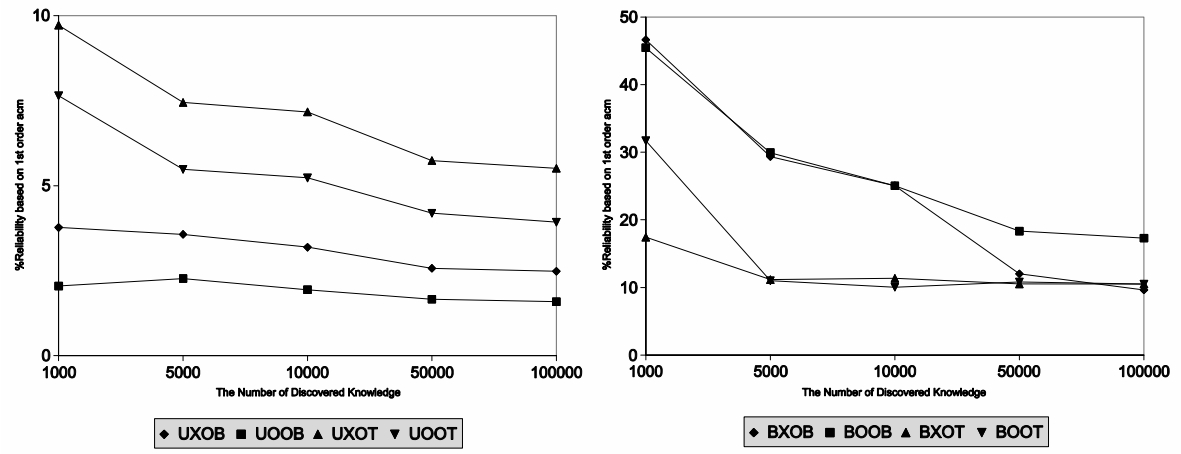

Figure 5. Reliability based on the first order acm (left: unigram, right: bigram).

5.2 Experimental Results

In every case, applying stopword removal obtains trivial higher reliability of discovered docsets. This result is consensus with some reports of the other works in IR and classification fields. Because of the obvious merit of applying stopword removal, only the representations with stopword removal are explored. The results in Figure 5 display the reliability score of discovered docsets based on the first order acm on unigram (left) and bigram (right). The unigram results show that the stemming scheme performs better than non-stemming. One possible reason is the case that unigrams are individual words and their stemmed words contain higher ambiguity in semantic than original words. When applying *tf*-weighting, the reliability score is dramatically higher than using normal binary-weighting. In most cases, the reliability score decreases when the number of discovered docsets increases. One exception is the case of unigram-based binary with stemming ("UOOB"), in which the reliability score increases when the number of discovered docsets is varied from 1000 to 5000. The reason is that the top discovered docsets in the list, ordered by descending supports, always contains only 2 or 3 documents. It is high possibility that the documents with different semantic will be bound together since their high supports are calculated from only counting the co-occurrence of stemmed words in both documents. Similar to the first experiment except only bigrams are taken into account instead of unigram, Figure 5 shows the results of four cases. Distinctly from unigrams, stemming provides a little better reliability than non-applying stemming. The reason for this inversion is that the bigrams has great influence in discarding the context ambiguity. Regardless of word ambiguity when applying stemming, the bigrams with stemming that have different semantic are less possible to occur in a set of documents. Surprisingly, the *tf*-weighting provides less reliability than binary weighting in some cases. One possible reason is the *tf* of bigrams is markedly very low since the rareness of the same bigrams in a document. Only *tf*-weighting may not provide good results because the bigrams which appear many times (high *tf*) in a document and do not have fine-grain semantic. The reliability score of bigrams is quite higher than unigrams. The bigram helps in representing the content clearly. Combining unigram and bigram is also investigated. Its reliability result falls between those of the individual two schemes. However, the trend of this reliability closes to the unigrams more than bigrams because the *tf* of unigrams is extensively larger than the *tf* of bigrams.

The evaluation of discovered docsets using the second and third order acm's is done. The results are shown in Figure 6 where the continuous lines show the reliability based on the second order acm and the dashed lines show the reliability based on the third order acm. In every cases, the reliability of the discovered knowledge is very high since the evaluation criteria is changed to become easier where the publication need not to be the direct citing/cited one. Even there is not much significant different in the reliability of different document representations with higher order acm, but the results are the same as previously discussed.

5.3 Discussion

This section gives the result of an analysis on the causes of errors. As a result, it can be concluded that there are three main reasons. The first cause is publication duplication. Some articles are written by the same group of authors but published in different places. They are very similar but most of them have no citation to each other, maybe since they were published in the same period. However, by our reliability score, this findings will trigger a zero score, even it should be fine. Anyway the reliability score increases when the result is evaluated based on higher order acm's, because of

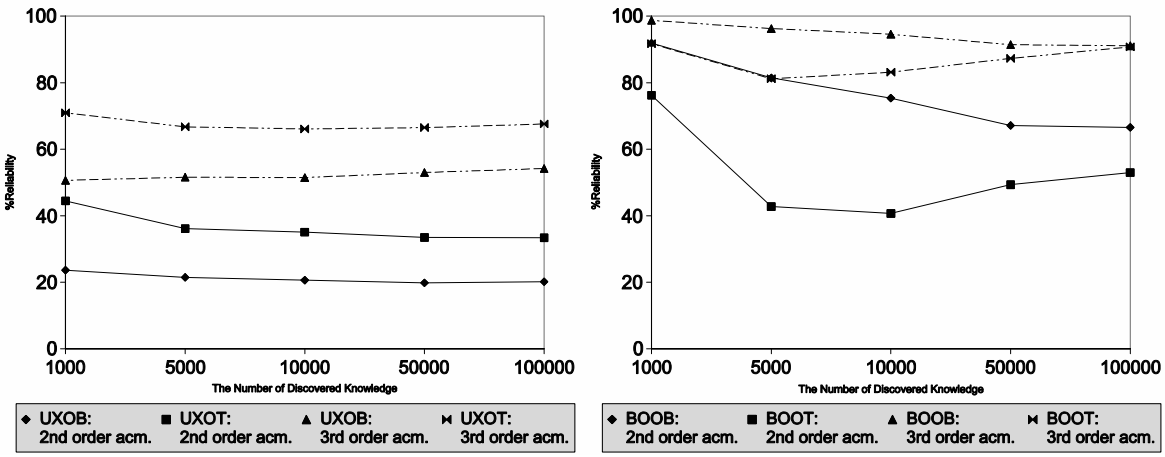

Figure 6. Reliability based on the second and third order acm's.

referenced publications. The second main error is incompleteness of citing information. This may occur due to either intentional reasons (e.g. limitation of paper length) or unintentional reasons (e.g. naively unknown). Some interesting connections are found but they are not included in the reference. This can be checked by hand but it is a quite labor-intensive task. The last reason of errors may be caused since some related articles may use different terms but with the same meaning. Some semantic approaches, such as principal component analysis or latent semantic indexing, may help in tackling this issue. A proper method of term weighting or normalization may be needed.

6. Conclusion

This paper proposes a method to discover relations among documents based on frequent itemset mining. The concept is proved in the context of discovering relations among such scientific publications. A set of meaningful relations among articles can be discovered efficiently by our modified frequent itemset mining. To measure the quality of discovered frequent document set (docset), the concept of order citation and order accumulative citation matrix are proposed. Based on these concept, the reliability score of frequent docsets is calculated. As document-term representation, unigram and bigram together with binary and vector-space-model schemes are investigated and compared. The reliability score of discovered docsets highly depends on the applied document representations used in constructing the input textual database. The experimental results show that the reliability score under the bigram model is dramatically higher than that of the unigram model with a gap of 3.7%-43.4%. This merit is totally devoted to the bigram in which it notably helps in context expressiveness. The results from applying stemming approach perform well only when combining with bigram-based. The *tf*-weighting can distinctly improve the discovered knowledge in the case of unigram-based, but performs inversely in the case of bigram-based. The reason is the *tf* of bigrams is markedly very low since the rareness of the same bigrams in a document. Many other efficiency weighting techniques can be used to solve this problems but they are still left to be investigated. Fortunately, our proposed concept on extensive frequent itemset mining is general enough to mine on any weighting techniques. As a future work, we will explore other types of term weighting and semantic approaches, such as latent semantic indexing and principal component analysis.

Acknowledgment

This work was funded by the National Electronics and Computer Technology Center (NECTEC) under research grant NT-B-22-I4-38-49-05, and the Royal Golden Jubilee Program of Thailand Resarch Fund.

References

[1] M.M. Kessler, “Bibliographic coupling between scientific papers”, *American Documentation*, Vol. 14, 1963, pp. 10-25.

[2] H. Small, “Co-Citation in the scientific literature: a new measure of the relationship between documents”, *Journal of the American Society for Information Science*, Vol. 42, 1973, pp. 676-684.

[3] R. Jelier, G. Jenster, L. Dorssers, C. van~der Eijk, E. van Mulligen, B. Mons and J.A. Kors, “Co-occurrence based meta-analysis of scientific texts: retrieving biological relationships between genes”, *Bioinformatics*, Vol. 21, No. 9, 2004, pp. 2049-2058.

[4] D. Swanson, “Medical literature as a potential source of new knowledge”, *Bulletin of the Medical Library Association*, Vol. 78, No. 1, 1990, pp. 29-37.

[5] R. Lindsay and G. M.D, “Literature-based discovery by lexical statistics”, *Journal of the American Society for Information Science*, Vol. 50, No. 7, 1999, pp. 574-587.

[6] F. Tao, F. Murtagh and M. Farid, “Weighted association rule mining using weighted support and significance framework”, *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, ACM Press, 2003, pp. 661-666.

[7] U. Yun and J.J. Leggett, “Wip: mining weighted interesting patterns with a strong weight and/or support affinity”, *Online Proceedings of 2006 SIAM Conference on Data Mining*, Bathesda, Maryland, USA, IEEE Computer Society, 2006, pp. 623-627.

[8] T. Theeramunkong, “Applying passage in web text mining”, *International Journal of Intelligence Systems*, Vol. 19, No. 1-2, 2004, pp. 149-158.

[9] K. Sriphaew and T. Theeramunkong, “Revealing topic-based relationship among documents using association rule mining”, *Artificial Intelligence and Applications*, 2005, pp. 112-117.

[10] G. Salton, “Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer”, *Addison-Wesley*, Boston, MA, 1989.

[11] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A.I. Verkamo, “Fast discovery of association rules”, *Advances in knowledge discovery and data mining*, Menlo Park, CA, USA, American Association for Artificial Intelligence, 1996, pp. 307-328.

[12] J. Han, J. Pei and Y. Yin, “Mining frequent patterns without candidate generation”, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, ACM Press, 2000, pp. 1-12.