Applying Latent Semantic Indexing in Frequent Itemset Mining for Document Relation Discovery

Thanaruk Theeramunkong[1], Kritsada Sriphaew[1,2], and Manabu Okumura[2]

[1] Sirindhorn International Institute of Technology, Thammasat University,
131 Moo 5 Tiwanont Rd., Bangkadi, Muang, Pathumthani 12000, Thailand
thanaruk@siit.tu.ac.th, kong@siit.tu.ac.th

[2] Precision and Intelligence Laboratory, Tokyo Institute of Technology,
4259 Nagatsuta Midori-ku Yokohama 226-8503, Japan
oku@pi.titech.ac.jp

Abstract. Word-based relations among technical documents are immensely useful information but often hidden in a large amount of scientific publications. This work presents a method to apply latent semantic indexing in frequent itemset mining to discover potential relations among scientific publications. In this work, two weighting schemes, tf and tfidf are investigated with the exploitation of latent semantic indexing. The proposed method is evaluated using a set of technical documents in a publication database by comparing the extracted document relations with their references (citations). To this end, the paper uses order accumulative citation matrices to evaluate the validity (quality) of discovered patterns. The results also show that the proposed method successfully discovers a set of document relations, comparing to the original method that uses no latent semantic indexing.

1 Introduction

Fast increasing of research publication has caused the difficulty for researchers to grasp movement or change in their area of interest. Such information overload becomes serious hindrance for researchers to position their own works against existing ones, or to find useful relations (or connections) among them. Although the publication of each work may include a list of related articles (documents) as its reference (called citation), 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., naïvely unknown). Enormous meaningful connections that permeate the literatures may remain hidden. Recently, there have been two different approaches to find relations among research documents. As the first approach, the citation-based method uses expansion of bibliography or citation information in scientific publication to find indirect relations, including measurement of impact factor [1], characterization of the citation [2], support of browsing citation graph [3] and so forth. For the task of relation discovery, two basic properties of citation, called bibligraphic coupling [4] and co-citation [5], can

T. Washio et al. (Eds.): PAKDD 2008, LNAI 5012, pp. 731–738, 2008.

be focused. Those previous works stated that any two documents tend to have relation with each other if they are citing to one or more documents in common (bibliographic coupling) or they are both cited by one or more documents in common (co-citation). As the second approach, the word- or term-based method exploits words or terms in a document as potential clues to detect relations between the document and other related documents. This method (later called word-based approach) discovers a set of documents with similar contents (topics) using either word co-occurrences or shared vocabularies, such as done in information retrieval, text categorization and text clustering. However, the process to find relations among two documents is computationally expensive since all combinations need to be considered for any possible relation [6]. Towards this problem, some recent works [7,8] have applied association rule mining (ARM) techniques to find n-ary document relations where a support can be set to avoid exploring all document combinations. Even such works could achieve discovery of high-quality relations to some extents, they still have some limitations due to direct use of words and terms in documents.

In this paper, we propose a method to apply latent semantic indexing in the process of discovering hidden relations among two documents. Two main objectives are (1) to study how well the word-based approach with different weighting (tf and tfidf) performs in finding relations among documents using ARM techniques, and (2) to study how much latent semantic indexing improves the conventional approach in finding useful hidden relations.

2 Frequent Itemset Mining

In the past, association rule mining (ARM) and frequent itemset mining (FIM) was known as a process to find co-occurrences (frequent patterns) in a database. In general, the conventional transactional database is presented in the term of item existences in the transaction. Although most ARM works deal with a this kind of databases, there are some attempts to extend the original framework to be able to assign the weights for items or transactions in the database, called weighted association rule mining [9]. In those works, 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 such item occurs. Unlike those works, our approach utilizes the term-document orientations, where the discovered frequent itemset is a set of documents which share a large number of terms as done in [7,8]. Note that a transaction corresponds to a term while an item corresponds to a document. Therefore, a “docset” (document set) is used in place of the term “itemset” in the traditional FIM approaches. The discovered results can be assumed as a term-based relation among documents where the relation is introduced by coincident terms. In Figure 1, two examples of the real-valued databases are defined in the form of well-known vector space model (VSM). The left part indicates how often

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

	d_1	d_2	d_3	d_4
t_1	$4\times \log 4/2 = 1.20$	$2\times \log 4/2 = 0.60$	$0\times \log 4/2 = 0.00$	$0\times \log 4/2 = 0.00$
t_2	$4\times \log 4/3 = 0.50$	$2\times \log 4/3 = 0.25$	$4\times \log 4/3 = 0.50$	$0\times \log 4/3 = 0.00$
t_3	$2\times \log 4/3 = 0.25$	$0\times \log 4/3 = 0.00$	$2\times \log 4/3 = 0.25$	$2\times \log 4/3 = 0.25$
t_4	$0\times \log 4/2 = 0.00$	$4\times \log 4/2 = 1.20$	$0\times \log 4/2 = 0.00$	$1\times \log 4/2 = 0.30$

Fig. 1. the term-document database with tf (left) and tfidf (right) term weightings

a term occurs in each document (called term frequency - tf) while the right part shows term frequency multiplied by the inverse document frequency (tfidf).

Traditionally, the support of a docset is defined by a ratio between the number of terms that exist in all documents in the docset and the total number of distinct terms in a database. To expand this concept to a real-valued database, the definition of support is generalized as follows. Let $\mathcal{D}$ be a set of documents (items) where $\mathcal{D} = \{d_1, d_2, ..., d_m\}$, and T be a set of terms (transactions) where $\mathcal{T} = \{t_1, t_2, ..., t_n\}$. Also let $w(d_i, t_j)$ represent a weight of a term t_j in a document d_i. A subset of $\mathcal{D}$ is called a docset whereas a subset of $\mathcal{T}$ is called a termset. Furthermore, a docset $X_k = \{x_1, x_2, ..., x_k\} \subset \mathcal{D}$ with k documents is called k-docset. The support of X_k is defined as follows.

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

By representing the data to be mined as shown in Figure 1, the new definition of support employs the *min* operation to find the weight of each term for a docset by selecting a minimum weight of such term among all documents in the docset. The *max* operation is applied for finding the maximum weight of each term in the database. The support of a docset will then be calculated from the ratio between the sum of all term weights for a docset and the sum of maximum weights of all terms in the database. While this definition can be applied for general real-valued databases, it also can used for the traditional FIM on boolean-valued databases with the same result. An example of docsets and their supports, for tf and tfidf databases, can be computed as shown in Figure 2. Besides support, a so-called confidence is used for generating confident association rules. Here, the confidence is left since it is out of scope in this work. Note that similar to conventional ARM, these generalized supports preserve two closure properties, 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"). For example, $sup(d_1) \geq sup(d_1d_2)$ and $sup(d_2) \geq sup(d_1d_2)$. The mathematical proof can be found in [8].

3 Representation and Latent Semantic Indexing

To represent document representation, term weighting can be performed to set importance level of a term in a document. This work uses two most common non-binary weightings: term-frequency (tf) and term-frequency-inverse-document-frequency (tfidf). Moreover, latent semantic indexing is applied to reveal hidden meaning in a document or a query. In this latent semantic space, a query and a

Docset	Generalized support	
	tf	tfidf
$\{d_1\}$	10/14 = 0.71	1.95/3.15 = 0.62
$\{d_2\}$	8/14 = 0.57	2.05/3.15 = 0.65
$\{d_3\}$	6/14 = 0.43	0.75/3.15 = 0.24
$\{d_4\}$	3/14 = 0.21	0.55/3.15 = 0.17
$\{d_1 d_2\}$	4/14 = 0.29	0.85/3.15 = 0.27
$\{d_1 d_3\}$	6/14 = 0.43	0.75/3.15 = 0.24
$\{d_1 d_4\}$	2/14 = 0.14	0.25/3.15 = 0.08

Docset	Generalized support	
	tf	tfidf
$\{d_2 d_3\}$	2/14 = 0.14	0.25/3.15 = 0.08
$\{d_2 d_4\}$	1/14 = 0.07	0.30/3.15 = 0.10
$\{d_3 d_4\}$	2/14 = 0.14	0.25/3.15 = 0.08
$\{d_1 d_2 d_3\}$	2/14 = 0.14	0.25/3.15 = 0.08
$\{d_1 d_2 d_4\}$	0/14 = 0.00	0.00/3.15 = 0.00
$\{d_2 d_3 d_4\}$	0/14 = 0.00	0.00/3.15 = 0.00
$\{d_1 d_3 d_4\}$	2/14 = 0.14	0.25/3.15 = 0.08
$\{d_1 d_2 d_3 d_4\}$	0/14 = 0.00	0.00/3.15 = 0.00

Fig. 2. Docsets and their generalized supports (tf vs. tfidf)

document may have high cosine similarity even if they do not share any common words or terms but their terms are semantically similar. Applied the concept of Singular Value Decomposition (SVD), LSI can also be viewed as a method for dimensionality reduction by a least-squared method [10]. SVD (also LSI) translates an input matrix A and represents it as $A^{'}$ in a lower dimensional space such that the 'distance' between the two matrices as measured by minimizing the 2-norm (Euclidean distance), $||A - A^{'}||_2$. It is possible to project an n-dimensional space of word-document matrices onto a k-dimensional space where n is the number of word types in the collection and k is relatively very small compared to n, say 100 and 150. The SVD projection is done by decomposing a document-by-term matrix $A_{t \times d}$ into the product of three matrices, $T_{t \times n}$, $S_{n \times n}$ and $D_{d \times n}$ as follows.

$$A_{t \times d} = T_{t \times n} \times S_{n \times n} \times D^T_{d \times n}$$

Here, t is the number of terms, d is the number of documents, $n = \min(t, d)$, T and D have orthonormal columns, i.e. $T \times T^T = I$ and $D^T \times D = I$, and S is a diagonal matrix, where $si, j = 0$ for $i \neq j$. Moreover, in some situations $rank(A) = r$ where $r \leq n$. In these situations, the diagonal elements of S are $\sigma_1, \sigma_2, ..., \sigma_n$ where $\sigma_i > 0$ for $1 \leq i \leq r$ and $\sigma_i = 0$ for $r < i \leq n$. For details of how to derive $T_{t \times n}$, $S_{n \times n}$ and $D_{d \times n}$, can be found in [10]. In this work, we investigate the best combination of the four schemes.

4 The Evaluation Method

To evaluate the result, we introduce an automatic evaluation where citation graph is used to evaluate our system based on its ability to find the relations that exist in the citation graph. Although human judgment is the best method for evaluation, it is a labor-intensive and time-consuming task. To do this, a citation graph is applied. Conceptually citations among documents in scientific publication collection form a citation graph, where a node corresponds to a document and an arc corresponds to a direct citation of a document to another document. Based on this citation graph, an indirect citation can be defined using the concept of transitivity. The formulation of direct and indirect citations can be given in the terms of the u-th order citation and the v-th order accumulative citation matrix as follows.

doc.	d_1	d_2	d_3	d_4	d_5	d_6
d_1	1	1	0	0	0	0
d_2	1	1	1	0	1	0
d_3	0	1	1	1	1	0
d_4	0	0	1	1	0	1
d_5	0	1	1	0	1	0
d_6	0	0	0	1	0	1

1-OACM

doc.	d_1	d_2	d_3	d_4	d_5	d_6
d_1	1	1	1	0	1	0
d_2	1	1	1	1	1	0
d_3	1	1	1	1	1	1
d_4	0	1	1	1	1	1
d_5	1	1	1	1	1	0
d_6	0	0	1	1	0	1

2-OACM

doc.	d_1	d_2	d_3	d_4	d_5	d_6
d_1	1	1	1	1	1	0
d_2	1	1	1	1	1	1
d_3	1	1	1	1	1	1
d_4	1	1	1	1	1	1
d_5	1	1	1	1	1	1
d_6	0	1	1	1	1	1

3-OACM

Fig. 3. The 1-, 2- and 3-OACMs

Definition 1 (the *u*-th order citation). For $x, y \in \mathcal{D}$, y is the u-th order citation of x iff the number of arcs in the shortest path between x to y in the citation graph is u (≥ 1). Conversely, x is called the u-th order citation of y.

Definition 2 (the *v*-th order accumulative citation matrix). Given a set of n distinct documents, the v-th order accumulative citation matrix (for short, v-OACM) is an $n \times n$ matrix, each element of which represents the citation relation δ^v between two documents x, y where $\delta^v(x, y) = 1$ when x is the u-th order citation of y and $u \leq v$, otherwise $\delta^v(x, y) = 0$. Note that $\delta^v(x, y) = \delta^v(y, x)$ and $\delta^v(x, x) = 1$.

For example, given a set of six documents $d_1, d_2, d_3, d_4, d_5, d_6 \in \mathcal{D}$ and a set of six citations d_1 to d_2, d_2 to d_3 and d_5, d_3 to d_5, and d_4 to d_3 and d_6, d_2 is the first, d_3 and d_5 is the second, d_4 is the third, and d_6 is the fourth order citations of the document d_1. The 1-, 2- and 3-OACMs can be created as shown in Figure 3. The 1-OACM can be straightforwardly constructed from the set of the first-order citation (direct citation). The $(v+1)$-OACM (mathematically denoted by a matrix A^{v+1}) can be recursively created from the operation between v-OACM (A^v) and 1-OACM (A^1) according to the following formula.

$$a_{ij}^{v+1} = \vee_{k=1}^{n} (a_{ik}^{v} \wedge a_{kj}^{1}) \tag{1}$$

where $\vee$ is an OR operator, $\wedge$ is an AND operator, a_{ik}^{v} is the element at the i-th row and k-th column of the matrix A^v and a_{kj}^{1} is the element at the k-th row and j-th column of the matrix A^1. Here, a v-OACM is a symmetric matrix.

The shorter the specific range is, the more restrict the evaluation is. With the concept of v-OACM stated in the previous section, we can realize this generalized evaluation by a so-called v-th order validity (for short, *v-validity*), where v corresponds to the specific range mentioned above. The formulation of the v-validity of a docset X ($X \subset \mathcal{D}$), denoted by $S^v(X)$, is defined as follows.

$$\mathcal{S}^v(X) = \frac{max_{x \in X}(\sum_{y \in X, y \neq x} \delta^v(x, y))}{|X| - 1} \tag{2}$$

Here, $\delta^v(x, y)$ is the citation relation defined by Definition 2. In the equation, we can observe that the v-validity of a docset is ranging from 0 to 1, i.e., $0 \leq \mathcal{S}^v(X) \leq 1$. The v-validity achieves the minimum (i.e., 0) when there is no citation relation among any document in the docset. On the other hand, it achieves the maximum (i.e., 1) when there is at least one document that has a citation relation with all documents in a docset. Intuitively, the validity of a bigger docset tends to have lower validity than a smaller one. Moreover, given a set of discovered docsets $\mathcal{F}$, its v-validity (later called *set v-validity*)), denoted by $\overline{\mathcal{S}}^v(\mathcal{F})$, can be defined as follows.

$$\overline{\mathcal{S}}^v(\mathcal{F}) = \frac{\sum_{X \in \mathcal{F}} w_X \times \mathcal{S}^v(X)}{\sum_{X \in \mathcal{F}} w_X} \tag{3}$$

where w_X is the weight of a docset (X). In this work, w_X is set to $|X| - 1$, the maximum value that the validity of a docset X can gain. For example, given the 1-OACM in Figure 3 and $\mathcal{F} = \{d_1d_2, d_1d_2d_3\}$, the set 1-validity of $\mathcal{F}$ (i.e., $\overline{\mathcal{S}}^1(\mathcal{F})$) equals to $\frac{(1\times\frac{1}{1})+(2\times\frac{2}{2})}{1+2} = \frac{3}{3} = 1$.

5 Experimental Settings and Results

A set of experiments are made to investigate how efficiently universal frequent itemset mining helps in discovering document relation among scientific research publications. In this work, an evaluation material is constructed from a collection of scientific research publications in the ACM Digital Library[1]. This dataset was originally used in [7]. As a seed of evaluation dataset, 200 publications are retrieved from each of the three computer-related classes, coded by B (Hardware), E (Data) and J (Computer) classes. Then the publications referred by these newly collected publications are also gathered and appended into the dataset. In total there are 10,817 publications collected as the evaluation material and used to generate citation graph under 1-OACM. As the result, only 36,626 citation edges are remained with an average of 7 citations (including both cite to and cited from other publications) per publication. For mining, we applied FP-tree algorithm, originally introduced in [11] and used the BOW library [12] as a tool for constructing an attribute-value database. The 524 stopwords and terms with very low frequency (less than 3 times) are omitted. Table 1 shows the validity of discovered document relations when either tf or tfidf are considered and LSI is applied with a thresholds of either 0.5, 0.7 or 1.0.

From the result shown in Table 1, some interesting characteristics can be observed. First, in most cases of the original space (w/o LSI), tfidf performs better than tf even there are few exceptions. The result implies that tfidf helps us obtain good representation for document relation discovery. Moreover, the result of 1-OACM becomes lower when N increases. This implies that better relations are located at higher ranks. In addition, with a higher-OACM, the method can achieve up to 90-100 % validity and has the same trend that the validity drops when N increases. Second, for both tf and tfidf, the 1-OACM performance of discovering document relations improves from 14.29 % to around 40 % for top-1000 documents when LSI is applied. Focusing on the 2-OACM and

[1] http://www.portal.acm.org

Table 1. Set validity of top-N rankings of discovered docsets when either tf or tfidf is used and LSI is applied with a thresholds of either 0.5, 0.7 or 1.0

		1-OACM		2-OACM		3-OACM	
Methods	N	tf	tfidf	tf	tfidf	tf	tfidf
w/o LSI	1000	14.29	25.00	85.71	100.00	100.00	100.00
	5000	37.59	38.03	87.23	95.77	95.62	97.18
	10000	18.22	38.97	58.94	87.66	87.13	93.81
	50000	6.16	16.24	35.91	60.52	75.68	94.05
	100000	4.37	14.36	31.22	55.83	74.49	93.08
LSI$_{\delta=0.5}$	1000	41.51	42.86	90.57	85.71	94.34	91.43
	5000	23.80	25.90	66.47	67.94	84.01	83.76
	10000	19.92	23.01	64.44	67.26	86.06	85.02
	50000	14.12	17.89	59.80	64.13	90.15	89.13
	100000	11.40	14.48	56.81	60.57	90.39	90.13
LSI$_{\delta=0.7}$	1000	47.14	44.15	90.00	80.32	95.71	85.64
	5000	25.95	28.28	69.09	70.86	85.98	85.72
	10000	22.26	25.59	67.80	70.64	87.52	86.95
	50000	14.77	19.91	60.76	66.72	91.43	91.27
	100000	12.09	16.06	57.51	61.73	91.52	90.98
LSI$_{\delta=1.0}$	1000	44.68	45.42	85.11	81.25	90.43	87.08
	5000	26.55	28.95	70.23	71.42	86.86	86.43
	10000	23.67	27.85	69.27	72.66	88.54	89.15
	50000	15.27	19.79	61.05	66.58	91.75	91.29
	100000	12.53	16.45	57.35	62.03	91.67	91.90

3-OACM performance, LSI is helpful to improve the validity of the discovered relations, especially for the cases of tf. In the cases of tfidf, LSI is helpful to improve validity of discovered document relations especially in the case of the 1-OACM. However, it is not useful for the 2-OACM and 3-OACM performance. This implies that LSI is helpful to increase the performance of discovering direct citations but not indirect citations. One implication is that the tfidf seems to be a good representation. Third, a stronger LSI (LSI with a higher threshold) performs better than a softer LSI (LSI with a lower threshold). This implies that LSI is useful to grasp the semantics of documents and then help increasing the discovery performance.

6 Conclusions

This work presents a new approach to discover document relations using association rule mining techniques with latent semantic indexing. Extended from the conventional frequent itemset mining, a so-called generalized support is proposed. The generalized support can serve a mining process of frequent itemsets from an attribute-value database where the values are weighted by real values, instead of boolean values as done in conventional methods. The quality of discovered document relations is measured under the concepts of the u-th order citation and the v-th order accumulative citation matrix. By experiments, we found out that tfidf seems better than tf and latent semantic indexing is helpful in discovering meaningful document relations. As future works, it is necessary to explore other suitable term weightings and normalization techniques. More explorations are needed for different data collections.

Acknowledgement

This work has been supported by Thailand Research Fund under project number BRG50800013. This work was also supported by NECTEC under project number NT-B-22-I4-38-49-05 and Royal Golden Jubilee (RGJ) Ph.D. program of the Thailand Research Fund (TRF).

References

1. Garfield, E.: Citation analysis as a tool in journal evaluation. Science 178(4060), 471–479 (1972)
2. An, Y., Janssen, J., Milios, E.E.: Characterizing and mining the citation graph of the computer science literature. Knowl. Inf. Syst. 6(6), 664–678 (2004)
3. Chen, C.: visualising semantic spaces and author co-citation networks in digital libraries. Information Processing and Management 35(3), 401–420 (1999)
4. Kessler, M.M.: Bibliographic coupling between scientific papers. American Documentation 14, 10–25 (1963)
5. Small, H.: Co-Citation in the scientific literature: a new measure of the relationship between documents. Journal of the American Society for Information Science 42, 676–684 (1973)
6. Theeramunkong, T.: Applying passage in web text mining. Int. J. Intell. Syst. (1-2), 149–158 (2004)
7. Sriphaew, K., Theeramunkong, T.: Revealing topic-based relationship among documents using association rule mining. Artificial Intelligence and Applications, 112–117 (2005)
8. Sriphaew, K., Theeramunkong, T.: Quality evaluation from document relation discovery using citation information. IEICE Transactions on Information and Systems E90-D(8), 1131–1140 (2007)
9. Yun, U., Leggett, J.J.: Wip: mining weighted interesting patterns with a strong weight and/or support affinity. In: Proceedings of 2006 SIAM Conference on Data Mining, pp. 623–627. IEEE Computer Society Press, Bathesda, Maryland, USA (2006)
10. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407 (1990)
11. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Chen, W., Naughton, J., Bernstein, P.A. (eds.) 2000 ACM SIGMOD Intl. Conference on Management of Data, pp. 1–12. ACM Press, New York (2000)
12. McCallum, A.K.: Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering (1996)