Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Text clustering using frequent itemsets

Wen Zhang [a,*], Taketoshi Yoshida [b], Xijin Tang [c], Qing Wang [a]

[a] *Lab for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, Beijing 100190, PR China*
[b] *School of Knowledge Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Tatsunokuchi, Ishikawa 923-1292, Japan*
[c] *Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, PR China*

ARTICLE INFO

Article history:
Received 28 May 2009
Received in revised form 27 October 2009
Accepted 23 January 2010
Available online 1 February 2010

Keywords:
Document clustering
Frequent itemsets
Maximum capturing
Similarity measure
Competitive learning

ABSTRACT

Frequent itemset originates from association rule mining. Recently, it has been applied in text mining such as document categorization, clustering, etc. In this paper, we conduct a study on text clustering using frequent itemsets. The main contribution of this paper is three manifolds. First, we present a review on existing methods of document clustering using frequent patterns. Second, a new method called Maximum Capturing is proposed for document clustering. Maximum Capturing includes two procedures: constructing document clusters and assigning cluster topics. We develop three versions of Maximum Capturing based on three similarity measures. We propose a normalization process based on frequency sensitive competitive learning for Maximum Capturing to merge cluster candidates into predefined number of clusters. Third, experiments are carried out to evaluate the proposed method in comparison with CFWS, CMS, FTC and FIHC methods. Experiment results show that in clustering, Maximum Capturing has better performances than other methods mentioned above. Particularly, Maximum Capturing with representation using individual words and similarity measure using asymmetrical binary similarity achieves the best performance. Moreover, topics produced by Maximum Capturing distinguished clusters from each other and can be used as labels of document clusters.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Document clustering or text clustering is one of the main themes in text mining. It refers to the process of grouping documents with similar contents or topics into clusters to improve both availability and reliability of text mining applications such as information retrieval [1], text classification [2], document summarization [3], etc. There are three kinds of problems in document clustering. The first one is how to define similarity of two documents. The second problem is how to decide appropriate number of document clusters in a text collection and the third one is how to cluster documents precisely corresponding to natural clusters.

The concept of frequent itemsets originates from association rule mining [4] which uses frequent itemsets to find association rules of items in large transactional databases. A frequent itemsets is a set of frequent items, which co-occur in transactions more than a given threshold value called minimum support. Recent studies on frequent itemsets in text mining fall into two categories. One is to use association rules to conduct text categorization [5,6] and the other one is to use frequent itemsets for text clustering [7,10–12]. The main concern of this paper is on the latter.

The motivation of adopting frequent itemsets for document clustering can be attributed to two aspects. The first one is the demand of dimensionality reduction for representation. In vector space model (VSM), bag of individual words causes huge dimensionality. Not all the documents in a collection contain all the index terms used in representation and as a result sparseness occurs in document vectors enormously. The second one comes from comprehensibility of clustering results. A frequent itemsets is a set of individual words which includes more conceptual and contextual meanings than an individual word.

The contribution of this paper is mainly three manifolds. First, we present a review of recent studies on using frequent itemsets in text clustering. Second, we propose Maximum Capturing (MC) for text clustering using frequent itemsets. MC can be divided into two components: constructing document clusters and assigning document topics. Minimum spanning tree algorithm [8] is employed to construct document clusters with three types of similarity measures. Frequency sensitive competitive learning [9] is used to normalize clusters into predefined number if necessary. Third, experiment evaluation shows that MC could produce clusters more closely related with natural clusters in document collection and, the topics assigned by MC distinguish clusters from each other and describe the common contents of documents in a cluster more appropriately than other methods.

The remainder of this paper is organized as follows: Section 2 presents a review of clustering methods using frequent pattern,

[*] Corresponding author. Tel.: +81 80 3049 6798.
E-mail addresses: zhangwen@itechs.iscas.ac.cn (W. Zhang), yoshida@jaist.ac.jp (T. Yoshida), xjtang@amss.ac.cn (X. Tang).

0950-7051/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2010.01.011

including CFWS [10], CMS [11], FTC [7] and FIHC [12]. Section 3 proposes Maximum Capturing, which comprises the process of constructing documents and assigning topics for the clusters. We also propose a normalization method to merge clusters into predefined number. Section 4 conducts experimental evaluation of the proposed method. Section 5 concludes the paper.

2. Existing clustering methods using frequent itemsets

This section reviews existing clustering methods using frequent sequences and frequent itemsets.

2.1. CFWS method

Clustering based on Frequent Word Sequence (CFWS) is proposed in [10]. CFWS uses frequent word sequence and K-mismatch for document clustering. The difference between word sequence and word itemset is that word sequence considers words' order while word itemsets ignores words' order.

Suppose we have a document collection and the items in each document are shown in Table 1. Frequent sequences are extracted from these documents as shown in Table 2. To save space of the paper, we only show the final result produced by CFWS in Table 3.

We can see from Table 3 that there are overlaps in the final clusters of CFWS. For instance, document 3 is in both cluster 1 and cluster 2, and document 4 is in both cluster 1 and cluster 3. With K-mismatch, frequent sequences of candidate clusters are used to produce final clusters. However, because of the transitivity of common items, silhouettes of final clusters will become more and more ambiguous when K-mismatch is running step by step. Consequently, all the documents in the collection may be clustered into one document cluster. That is, trivial clustering is produced.

2.2. CMS method

Document Clustering Based on Maximal Frequent Sequences (CMS) is proposed in [11]. A frequent sequence is maximal if it is not a subsequence of any other frequent sequence. The basic idea of CMS is to use maximal frequent sequences (MFS) of words as features in vector space model (VSM) for document representation and then k-means is employed to group documents into clusters.

Taking the same documents in Table 1 for example, the maximal frequent sequences are {c, e, d}, {b, e}, {b, c} and {d, a}. Thus, by VSM and Boolean weighting, 9 documents were represented in Table 4. Table 5shows the clusters produced by k-means clustering for the above document vectors.

CMS is rather a method concerning feature selection in document clustering than a specific clustering method. Its performance completely depends on the effectiveness of using MFS for document representation in clustering, and the effectiveness of k-means.

Table 1
A document collection with items in each document.

Document ID	Sequence of words
1	c, e, a
2	d, b, e
3	b, c, e, d
4	c, e, d, a
5	b, e
6	c, d, a
7	b, c, a
8	b, c
9	c, e, d

Table 2
Frequent itemsets extracted from documents in Table 1 and their corresponding documents (minimum support = 20% and minimum length of FWS = 2).

Frequent sequence	List of documents
{c, e}	1, 3, 4, 9
{b, e}	2, 5
{b, c}	3, 7, 8
{e, d}	3, 4, 9
{c, e, d}	3, 4, 9
{d, a}	4, 6

Table 3
Final clusters produced by CFWS on documents shown in Table 1 (minimum support = 20% and minimum length of FWS = 2).

Cluster ID	List of documents
1	1, 3, 4, 9
2	2, 5, 3, 7, 8
3	4, 6

Table 4
Document representation using MFS.

Document ID	Document vector
1	(0, 0, 0, 0)
2	(0, 1, 0, 0)
3	(1, 0, 1, 0)
4	(1, 0, 0, 1)
5	(0, 1, 0, 0)
6	(0, 0, 0, 1)
7	(0, 0, 1, 0)
8	(0, 0, 1, 0)
9	(1, 0, 0, 0)

Table 5
Document clusters produced by k-means with different number of clusters.

Number of clusters	Document clusters
2	(1, 3, 7, 8, 9), (2, 4, 6)
3	(1, 7, 8), (2, 5), (3, 4, 6, 9)
4	(1, 9), (2, 5), (3, 7, 8), (4, 6)
5	(1, 6), (7, 8), (3), (4, 9), (2,5)

2.3. FTC method

Frequent Term-Based Clustering (FTC) is proposed for document clustering in Beil et al. [7]. The basic motivation of FTC is to produce document clusters with overlaps as few as possible. FTC works in a bottom-up fashion. Starting with an empty set, it continues selecting one more element (one cluster description) from the set of remaining frequent itemsets until the entire document collection is contained in the cover of the set of all chosen frequent itemsets. In each step, FTC selects one of the remaining frequent itemsets which has a cover with minimum overlap with the other cluster candidates, i.e. the cluster candidate which has the smallest entropy overlap (EO) value. The documents covered by the selected frequent itemsets are removed from the collection D, and in the next iteration, the overlap for all remaining cluster candidates is recomputed with respect to the reduced collection. The final clusters produced by FTC method with the documents shown in Table 1 are shown in Table 6.

In FTC, a cluster candidate is represented by a frequent itemsets and the documents in which the frequent itemsets occur. It calculates each candidate's EO which is decided by occurrence distribu-

Table 6
Document clusters produced by FTC method with documents in Table 1 (minimum support = 30%, cov(F_i) $\geqslant$ 3).

Cluster ID	Label	List of documents
1	{c, d, e}	3, 4, 9
2	{c, d}	6
3	{c, e}	1
4	{a, c}	7
5	{d, e}	2
6	{e}	5
7	{b, c}	8

tion of the candidates' documents. Thus, FTC tends to select cluster candidate, of which its number of documents is small while occurrence frequencies of these documents are large, as a document cluster. However, it will cause large amount clusters with only one document, i.e. isolated documents.

2.4. FIHC method

Frequent Itemset-based Hierarchical Clustering (FIHC) is proposed in [12]. Two kinds of frequent item are defined in FIHC: global frequent item and cluster frequent item. FIHC develops four phases to produce document clusters: finding global frequent itemsets, initial clustering, tree construction, and pruning. Fig. 3 illustrates the working process of FIHC to produce clusters for documents shown in Table 1. To save space of the paper, we only show the final results produced by FIHC in Fig. 1.

FIHC is based on cluster profile, not pairwise similarity used in classic clustering method. FIHC provides a tree for document clusters which is easy to browse with meaningful cluster description. Its characteristics of scalability and non-sensitivity to parameters are desirable properties for clustering analysis. However, we conjecture that it has three disadvantages in practical application (we do not discuss these disadvantages in details to condense this paper). First, it cannot solve cluster conflict when assigning documents to clusters. That is, a document may be partitioned into different clusters and this partition has great influence on the final clusters produced by FIHC. Second, after a document has been assigned to a cluster, the cluster frequent items were changed and FICH does not consider this change in afterward overlapping measure. Third, in FIHC, frequent itemsets is used merely in constructing initial clusters. Other processes in FIHC, such as making clusters disjoint and pruning, are based on single items of documents and decided by initial clusters. One motivation of using frequent itemsets is to use word co-occurrence of documents, and co-occurrence of frequent itemsets can provide more information for clustering than co-occurrence of single items. For this reason, we argue that FIHC is not purely based on frequent itemsets.

3. Maximum Capturing

In this section, we propose a new method called Maximum Capturing for document clustering using frequent itemsets.

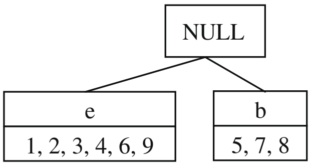


Fig. 1. Clusters produced by FIHC with documents in Table 1 (minimum global support = 30% and minimum cluster support = 60%).

3.1. The motivation

Our motivation of proposing Maximum Capturing (MC) for document clustering is to produce natural and comprehensible document clusters.

To produce natural clusters (that is, document categories given by human beings), we propose to use frequent itemsets for representation and measure similarities of documents based on co-occurrences of frequent itemsets in documents. This idea is the same as the representation method used in CMS (frequent sequences are included in frequent itemsets). Frequent itemsets is a set of words which frequently co-occur in documents. We conjecture that frequent itemsets contribute more meaning to document content and contain more semantics than individual terms, so they will improve the accuracy of similarity measure of documents and as a result, the quality of document clusters will be improved.

To make document clusters comprehensible, most frequent itemsets in a document cluster are assigned as the topic of the cluster. Because documents with largest number of common frequent itemsets are assigned into a same cluster, cluster topics will be the most representative frequent itemsets in the cluster and thus distinguish clusters from each other. By this method, the comprehensibility of topics of clusters will be improved.

3.2. The working process of Maximum Capturing

The MC proposed in this paper can be divided into two components: constructing document clusters and assigning cluster topics.

Minimum spanning tree algorithm [8] is employed to construct document clusters. We regard each document as a node in the network and the goal is to link all the documents with totally maximum similarities of document pairs.

In MC, a document D is denoted by the frequent itemsets in it. That is, $D_j = \{F_{j1}, \ldots, F_{jn}\}$, where n is the total number of frequent itemsets in D_j. We define the similarity measure between D_i and D_j in three ways: (1) number of common frequent itemset they have in common; (2) total weights of frequent itemsets they have in common; and (3) the asymmetrical binary similarity [17] between their frequent itemsets. It should be pointed out that the weight of a frequent itemset is the document frequency of the frequent itemset in the document collection (that is, $w_{F_{jk}} = df_{F_{jk}}$ $(1 \leqslant k \leqslant n)$). We use asymmetrical binary similarity in the third measure because long documents will have more frequent itemsets than short documents. We do not consider the lengths of frequent itemsets because a long frequent itemsets contains more frequent sub-itemsets than a short frequent itemsets.

Based on the above similarity measures between two documents, a similarity matrix is constructed which reflects the similarities of all the document pairs in a document collection. We define a similarity matrix A, with $a[i][j] = \mathrm{sim}[i][j]$, if $i < j$; otherwise $a[i][j] = 0$. For convenience, MC using the above three similarity measures will be abbreviated as MC-1, MC-2 and MC-3, respectively, in the remaining sections of this paper.

The basic idea of MC is that, if a document pair has the largest similarity among all the document pairs, then the two documents of the document pair should be assigned into same cluster. For instance, if d_i and d_j are in a cluster, and we find another document d_k has the maximum similarity with d_j, then d_i, d_j and d_k should be included in one cluster. The detailed procedure of MC in constructing document clusters is described in Fig. 2.

The working process of MC-1 using documents in Table 1 is shown in Fig. 3. We can see from Fig. 3 that basically, MC produces clusters by maximizing similarity values between documents within clusters. It prefers to create new cluster other than adding documents to existing clusters in order to balance the amount of

Step 1: Construct similarity matrix A;

Step 2: Find the minimum value excluding zero in A;

Step 3: Find maximum value in A, and then finding all document pairs of unclustere d documents (i.e. at least one of the two document of the document pair has not been assigned to any cluster) with the maximum value.

Step 4: If the maximum value found in step 3 is equal to the minimum value found i n Step 2, all documents in corresponding document pairs which do not attach to any cluster are used to construct a new cluster. If the maximum value found i n step 3 is not equal to the minimum value found in Step 2, then, for all found document pairs with the maximum value, the following process is conducted. First, for each document pair if two documents do not belong to any cluster, then these two documents are grouped together to construct a document cluster. Next, for each document pair if one of its two documents belongs to an existing cluster, then the other unattached document is attached to this cluster. Finall y, similarities of the found document pairs were set to be zero in A. Go to step 3.

Step 5: If there are any documents which do not attach to any cluster, then each of these documents is used to construct a new cluster.

Fig. 2. The procedure of MC in constructing document clusters.

documents in a cluster. There is no overlap among document clusters produced by MC.

In assigning topics of document clusters produced by MC, our basic intuition is that topics of document clusters should be as specific as possible and do not have overlaps among clusters, because the topic of a document cluster should distinguish the cluster from other clusters. The following procedure shown in Fig. 4 is used to produce topics for document clusters in MC.

Fig. 5 is the process of assigning topics of clusters produced by MC shown in Fig. 3. We can see that there is no overlap in topics of document clusters.

3.3. Normalizing documents into predefined number of clusters

Sometimes, we presumably know how many clusters should be produced for a document collection based on prior domain knowledge. In this condition, it is necessary to normalize documents into predefined number of clusters. The normalization process MC method is developed as follows.

First, similarity of two clusters in MC is defined as sum of similarities of documents in the two clusters shown in Eq. (1).

$$\mathrm{sim}(C_i, C_j) = \sum_{d_m \in C_i} \sum_{d_n \in C_j} \mathrm{sim}(d_m, d_n). \tag{1}$$

Second, the normalization process shown in Fig. 6, which is used to merge clusters combined with frequency sensitive competitive learning [9].

4. Experimental evaluations

In this section, a series of experiments are carried out using bench mark data sets to examine the performance of MC in comparison with the methods mentioned in Section 2.

4.1. Benchmark data sets

An English corpus and a Chinese corpus are used as the bench mark data sets in the experiments. Specifically, for English corpus, Reuters-21578 document collection (online: http://www.daviddlewis.com/resources/test-collections/reuters21578/) was applied as the benchmark data set. It appeared as Reuters-22173 in 1991 and was indexed with 135 categories by personnel from Reuters Ltd. in 1996. The documents of six categories shown in Table 7 are selected to conduct evaluation because their lengths are relatively longer than the lengths of documents in other categories, for convenience of multiword expression extraction.

For the Chinese corpus, TanCorpV1.0 is employed, which can be downloaded freely (http://www.searchforum.org.cn/tansongbo/corpus.htm). On the whole, it has 14, 150 documents in 20 categories from Chinese academic journals concerning computers, agriculture, politics. Here, documents from four categories shown in Table 8 are assigned as the Chinese bench mark data set.

In data preprocessing, stop-word[1] elimination and stemming processing[2] are conducted for the English documents. Thus, 2485 unique individual words are obtained. Morphological analysis[3] is conducted for the Chinese documents. Consequently, 281,111 individual words are obtained.

4.2. Text representation

Two methods are employed to represent the documents in both benchmark data sets: representation with individual words and representation with multiword expression. The motivation of using representation of multiword expression is to reduce the dimensionality of document vector. TF*IDF [13] is used to select individual words, and their term frequency and document fre-

[1] We obtain the stop-words from United States Patent and Trademark Office (USPTO) patent full-text and image database at %3chttp://ftp.uspto.gov/patft/help/stopword.htm%3e. It includes about 100 usual words as stop-words. The part-of-speech of English word is determined by QTAG which is a probabilistic parts-of-speech tagger and can be downloaded freely online: http://www.english.bham.ac.uk/staff/omason/software/qtag.html.

[2] Porter stemming algorithm is used for English stemming processing which can be downloaded freely online: http://tartarus.org/~martin/PorterStemmer/.

[3] Because Chinese is character based, we conducted the morphological analysis using the ICTCLAS tool. It is a Chinese Lexical Analysis System. Online: http://nlp.org.cn/~zhp/ICTCLAS/codes.html.

Step 1 Constructing similarity matrix A;

	1	2	3	4	5	6	7	8	9
1		1	3	3	1	3	3	1	3
2			4	2	2	1	1	1	3
3				6	2	3	3	3	7
4					0	4	2	1	6
5						0	1	1	1
6							3	1	3
7								3	1
8									1

Step 2 Finding the minimum value excluding zero in A as 1;

Step 3 Document pair (3, 9) has the largest similarity as 7, so MC produces cluster (3, 9) and sets similarity of (3, 9) as 0. The unclustered document set is {1, 2, 4, 5, 6, 7, 8}.

	1	2	3	4	5	6	7	8	9
1		1	3	3	1	3	3	1	3
2			4	2	2	1	1	1	3
3				6	2	3	3	3	0
4					0	4	2	1	6
5						0	1	1	1
6							3	1	3
7								3	1
8									1

Step 4 Document pairs (3, 4) and (4, 9) have the largest similarity as 6, so MC adds 4 into cluster (3, 9) as cluster (3, 4, 9) and sets similarities of (3, 4) and (4, 9) as 0. The unclustered document set is {1, 2, 5, 6, 7, 8}.

	1	2	3	4	5	6	7	8	9
1		1	3	3	1	3	3	1	3
2			4	2	2	1	1	1	3
3				0	2	3	3	3	0
4					0	4	2	1	0
5						0	1	1	1
6							3	1	3
7								3	1
8									1

Step 5 Document pairs (2, 3) and (4, 6) have the largest similarity as 4, so MC adds 2 and 6 into cluster (3, 4, 9) to produce cluster (2, 3, 4, 6, 9) and sets similarities of (2, 3) and (4, 6) as 0. The unclustered document set is {1, 5, 7, 8}.

	1	2	3	4	5	6	7	8	9
1		1	3	3	1	3	3	1	3
2			0	2	2	1	1	1	3
3				0	2	3	3	3	0
4					0	0	2	1	0
5						0	1	1	1
6							3	1	3
7								3	1
8									1

Step 6 Document pairs (1, 3) (1, 4) (1, 6) (1, 7) (1, 9) and (7, 8) have the largest similarity as 3, because either document of (7, 8) does not belong to an existing cluster, (7, 8) is used to construct a new cluster and similarity of (7, 8) is set as 0. Document 1 is added into cluster (2, 3, 4, 6, 9) because (1, 3) is found before (7, 8). Thus, we have the clusters as (1, 2, 3, 4, 6, 9) and (7, 8). The similarities of all the found document pairs in this step are set as 0. The unclustered document set is {5}.

	1	2	3	4	5	6	7	8	9
1		1	0	0	1	0	0	1	0
2			0	2	2	1	1	1	0
3				0	2	0	0	0	0
4					0	0	2	1	0
5						0	1	1	1
6							0	1	0
7								0	1
8									1

Step 7 Document pairs (5, 7) (5, 8) (5, 9) have the largest similarity as 1, which is equal to the minimum similarity excluding zero in similarity matrix. Document 5 is used to construct a unique cluster. Thus, we have the final clusters as (1, 2, 3, 4, 6, 9), (7, 8) and (5).

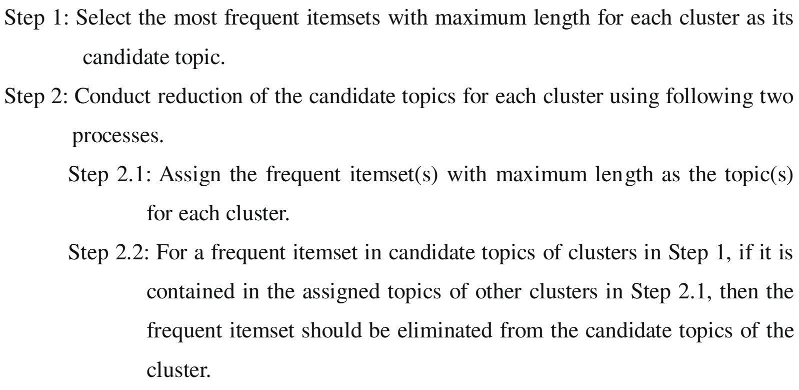
Fig. 3. The working process of MC-1 for producing clusters for the document collection shown in Table 1.

quency should be more than 2. We select those individual words whose TF*IDF values are at the top 90% for representation. The representation method is to transfer a document into a database transaction, correspondingly, according to the individual words' presence and absence in the document.

In representation with multiword expression, the method proposed in [14] is adopted to extract multiword expressions from the documents. Thus, 7927 multiword expressions are extracted from the English documents and 9074 multiword expressions are produced from the Chinese documents. Then, documents are represented with multiword expressions using the same method of individual words.

4.3. Frequent pattern extraction

FP-tree algorithm [15] is used to extract frequent itemsets from documents. BIDE algorithm [18] is used to extract frequent sequences from documents. The algorithm proposed in [19] is uti-



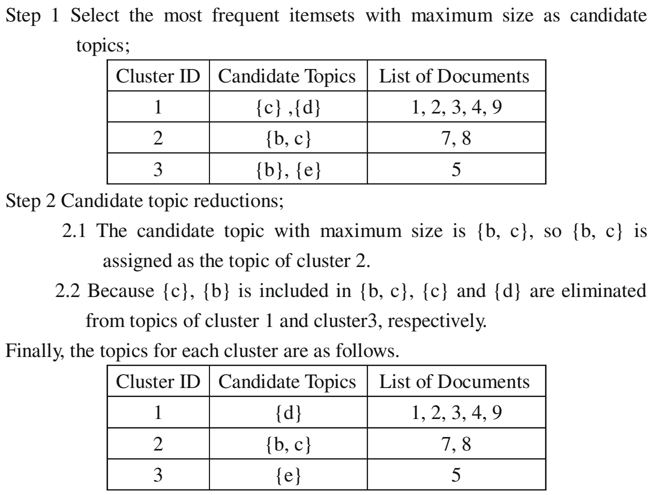
Step 1: Select the most frequent itemsets with maximum length for each cluster as its candidate topic.

Step 2: Conduct reduction of the candidate topics for each cluster using following two processes.

Step 2.1: Assign the frequent itemset(s) with maximum length as the topic(s) for each cluster.

Step 2.2: For a frequent itemset in candidate topics of clusters in Step 1, if it is contained in the assigned topics of other clusters in Step 2.1, then the frequent itemset should be eliminated from the candidate topics of the cluster.

Fig. 4. The detailed procedures of MC in assigning document topics.



Step 1 Select the most frequent itemsets with maximum size as candidate topics;

Cluster ID	Candidate Topics	List of Documents
1	{c} ,{d}	1, 2, 3, 4, 9
2	{b, c}	7, 8
3	{b}, {e}	5

Step 2 Candidate topic reductions;

2.1 The candidate topic with maximum size is {b, c}, so {b, c} is assigned as the topic of cluster 2.

2.2 Because {c}, {b} is included in {b, c}, {c} and {d} are eliminated from topics of cluster 1 and cluster3, respectively.

Finally, the topics for each cluster are as follows.

Cluster ID	Candidate Topics	List of Documents
1	{d}	1, 2, 3, 4, 9
2	{b, c}	7, 8
3	{e}	5

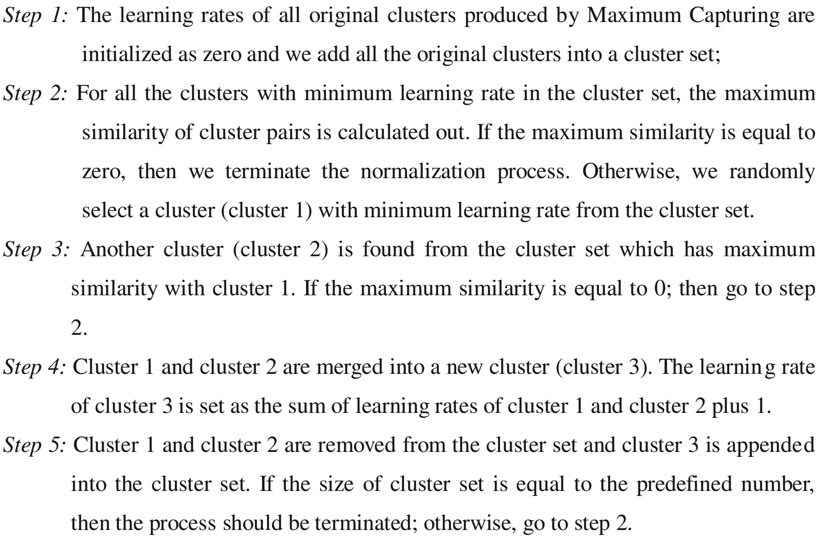
Fig. 5. The working process of MC-1 for constructing topics of clusters in Fig. 3.

Table 7
Benchmark data set selected from the English corpus.

Category	Number of documents
Coffee	102
Sugar	111
Interest	189
Money-fx	246
Trade	330
Crude	335
Total	1313

lized to extract maximal frequent sequences from documents. In order to observe the relationship between support thresholds for frequent itemset extraction and the quality of clustering results, we vary support thresholds at different proportions of the number of documents in collection as 0.01, 0.015, 0.02, 0.025 and 0.03. On one hand, large value on minimum support will bring about small number of frequent itemsets and sequences. Consequently, those frequent itemsets and sequences cannot cover all the documents in data sets. On the other hand, small value on minimum support will produce huge frequent itemsets and sequences that computer cannot handle.

Tables 9–12 show the numbers of frequent itemsets (FI), frequent word sequences (FWS) and maximal frequent itemsets (MFS) extracted from document transactions in the benchmark datasets using different minimum support (MS) values. Notice we have two versions of frequent itemsets and sequences: one is



Step 1: The learning rates of all original clusters produced by Maximum Capturing are initialized as zero and we add all the original clusters into a cluster set;

Step 2: For all the clusters with minimum learning rate in the cluster set, the maximum similarity of cluster pairs is calculated out. If the maximum similarity is equal to zero, then we terminate the normalization process. Otherwise, we randomly select a cluster (cluster 1) with minimum learning rate from the cluster set.

Step 3: Another cluster (cluster 2) is found from the cluster set which has maximum similarity with cluster 1. If the maximum similarity is equal to 0; then go to step 2.

Step 4: Cluster 1 and cluster 2 are merged into a new cluster (cluster 3). The learning rate of cluster 3 is set as the sum of learning rates of cluster 1 and cluster 2 plus 1.

Step 5: Cluster 1 and cluster 2 are removed from the cluster set and cluster 3 is appended into the cluster set. If the size of cluster set is equal to the predefined number, then the process should be terminated; otherwise, go to step 2.

Fig. 6. The normalization process of MC for merging clusters.

Table 8
Benchmark data set selected from the Chinese corpus.

Category	Number of documents
Agriculture	300
History	300
Politics	300
Economy	300
Total	1200

Table 9
The number of frequent itemsets and frequent sequences from English documents using representation of individual words.

MS	0.01	0.015	0.02	0.025	0.03
# of FI	1560	645	342	233	149
# of FWS	27,463	15,518	7798	4044	2979
# of MFS	2748	1329	923	636	478

Table 10
The number of frequent itemsets and frequent sequences from English documents using representation of multiword expressions.

MS	0.01	0.015	0.02	0.025	0.03
# of FI	2188	899	387	242	142
# of FWS	1285	540	298	183	123
# of MFS	825	342	204	122	80

Table 11
The number of frequent itemsets and frequent sequences from Chinese documents using representation of individual words.

MS	0.01	0.015	0.02	0.025	0.03
# of FI	21,576	17,635	13,993	9546	8432
# of FWS	32,899	21,947	19,529	17,694	12,983
# of MFS	12,986	9853	8763	7683	4438

Table 12
The number of frequent itemsets and frequent sequences from Chinese documents using representation of multiword expressions.

MS	0.01	0.015	0.02	0.025	0.03
# of FI	9180	7613	6381	5856	3641
# of FWS	15,217	10,334	9532	8763	6651
# of MFS	5336	3231	2149	1055	897

representation using individual words and the other is representation using multiword expressions.

We can see from Tables 9–12 that although all the frequent word sequences are included in the frequent itemsets theoretically, it is not case in practice. The reason is that when we extract frequent itemsets from documents, we ignore the term frequency of words. That is, in frequent itemsets extraction, what we care is merely the presence and absence of a term in documents, not the positions or frequency of occurrence of words in a document. However, in frequent sequence extraction, we intend to keep the information of word orders. That is, all the words in a document should be transferred into a sequence. Especially in representation using individual words, the number of frequent word sequences is far larger than the number of frequent itemsets because individual words have larger term frequency than multiword expression. The numbers of both frequent itemsets and frequent sequences of English documents are smaller than those of Chinese documents, because Chinese documents are longer than English documents in our bench mark data sets.

4.4. Evaluation methods

F-measure [16] is employed to estimate performances of the above three clustering methods after clustering results are normalized into a fixed number of predefined clusters.

The formula of *F*-measure is depicted as Eqs. (2)–(5).

$$P(i,j) = \frac{n_{ij}}{n_j} \tag{2}$$

$$R(i,j) = \frac{n_{ij}}{n_i} \tag{3}$$

$$F(i,j) = \frac{2 * P(i,j) * R(i,j)}{P(i,j) + R(i,j)} \tag{4}$$

$$F = \sum_i \frac{n_i}{n} \max_j F(i,j) \tag{5}$$

Here, n_i is the number of documents of class i, n_j is the number of documents of cluster j, and n_{ij} is the number of documents of class i in cluster $j.n$ is the total number of documents in the collection. $P(i,j)$ is the precision of cluster j in class i; $R(i,j)$ is the recall of class i in cluster j; $F(i,j)$ is the *F*-measure of cluster j in class i. In general, the larger the *F*-measure is, the better the clustering result is.

4.5. Evaluation results

4.5.1. Evaluation of MC without normalization process

Tables 13–16 are the evaluation results by *F*-measure of MC-1, MC-2, MC-3 and the clustering methods mentioned in Section 2. We conduct the evaluation on both representation of individual words and representation of multiword expression. We do not compare MC with traditional methods such as *k*-means and bisecting *k*-means because all the methods mentioned in Section 2 have been studied as more efficient than the traditional methods.

We can see from Tables 13 and 14 that, on English benchmark data set, MC, which includes MC-1, MC-2 and MC-3, significantly outperforms other methods in clustering documents. The better performance of MC implies that largest co-occurrence of frequent itemsets between documents can precisely characterize the documents in same natural cluster. We conjecture that the better performance of MC-3 than MC-1 and MC-2 is because of the different lengths of English documents and the capability of MC-3 to consider the lengths of frequent itemsets.

FIHC outperforms all the other methods except MC. We conjecture that the favorable performances of FIHC for clustering can be attributed to matching frequent items between documents and cluster candidates. However, the variations of frequent items of cluster candidates are limited in FIHC due to initial cluster construction using frequent itemsets. In the case that there are skewed initial clusters, i.e. the numbers of documents in different clusters are significantly different, large initial clusters will have more cluster frequent items than small initial clusters. Consequently, docu-

Table 13
F-measures of clustering results of English documents using representation of individual words.

MS	MC-1	MC-2	MC-3	FTC	CFWS	FIHC	CMS
0.01	0.6367	0.6746	**0.7397**	0.3712	0.3264	0.5269	0.3532
0.015	0.6697	0.6734	**0.7214**	0.3712	0.3156	0.5615	0.3449
0.02	0.6758	0.5908	**0.7128**	0.3631	0.3321	0.5012	0.3745
0.025	0.6365	0.5973	**0.6830**	0.4214	0.3221	0.5073	0.3585
0.03	0.5428	0.6120	**0.6514**	0.3869	0.3117	0.4879	0.3154

Table 14
F-Measure for clustering results of English documents using representation of multiword expressions.

MS	MC-1	MC-2	MC-3	FTC	CFWS	FIHC	CMS
0.01	0.5600	0.4938	**0.5903**	0.3683	0.3172	0.4536	0.3237
0.015	0.5408	0.4717	**0.5513**	0.3683	0.3313	0.4213	0.3103
0.02	0.5074	0.4707	**0.5287**	0.3828	0.3067	0.4118	0.3009
0.025	0.4990	0.4878	**0.5319**	0.3829	0.2835	0.3873	0.2813
0.03	**0.5180**	0.4820	0.5098	0.3587	0.2415	0.3664	0.2528

Table 15
F-Measure for clustering results of Chinese documents using representation using individual words.

MS	MC-1	MC-2	MC-3	FTC	CFWS	FIHC	CMS
0.01	**0.6817**	0.6713	0.6614	0.3514	0.3225	0.6722	0.3131
0.015	**0.6659**	0.6517	0.6437	0.3514	0.3127	0.6615	0.3216
0.02	0.6712	**0.6722**	0.6515	0.3678	0.3237	0.6312	0.3035
0.025	0.6853	0.6513	**0.6317**	0.3717	0.3315	0.6073	0.2585
0.03	0.5544	0.5618	0.5582	0.3289	0.2869	**0.5879**	0.2154

Table 16
F-measure for clustering results of Chinese documents using representation using multiword expressions.

MS	MC-1	MC-2	MC-3	FTC	CFWS	FIHC	CMS
0.01	**0.6995**	0.6813	0.6910	0.3674	0.3367	0.6042	0.3535
0.015	**0.6823**	0.6779	0.6647	0.3674	0.3423	0.6213	0.3657
0.02	0.6815	0.6215	**0.7099**	0.3354	0.3318	0.6452	0.3534
0.025	0.6023	0.5894	0.5753	0.3678	0.3482	**0.6159**	0.3478
0.03	**0.5745**	0.5513	0.5627	0.3749	0.3057	0.5618	0.3389

ments are more likely to be regarded as belonging to large initial clusters. Sometimes, the skew initial clusters may be produced by some trivial frequent itemsets and consequently, skew clustering results will be produced by FIHC. However, this disadvantage can be avoided in Maximum Capturing, especially in MC-3, where trivial frequent itemsets do not have great influence on the co-occurrence of frequent itemsets.

As mentioned in Section 3.1, FTC constructs clusters actually not based on the contents of documents but based on the distribution of occurrences of documents over all the cluster candidates. Thus, it cannot characterize the natural clusters of documents which are based on the content of documents.

CFWS is prone to produce skew clusters because *K*-mismatch will produce large overlapping between clusters and consequently overlapping coefficients between large clusters are more likely to be larger than small clusters.

CMS do not produce ideal results due its two deficiencies. The first one is that the number of maximal frequent sequence is not large enough to capture the natural clusters. The second one is that the size of maximal frequent sequence, i.e. the number of words in a maximal frequent sequence, is ignored in CMS method.

The clustering results of representation of individual words are superior over the clustering results of multiword expression. We conjecture that this outcome can be traced to that multiword expression is a larger lexical unit than individual word, and when used for representation, it ignores some important contents of documents.

From Tables 15 and 16, the conclusions drawn on English documents can also be drawn on Chinese documents. However, the performance of FIHC is better than that of English documents and the performances of MC-1, MC-2 and MC-3 are comparable to each other. We conjecture the better performance of FIHC can be traced to the same number of documents in each category of Chinese documents and the large number of frequent itemsets which bring about large divergence on frequent items. The comparable performances of MC-1, MC-2 and MC-3 imply that both the weight and the number of a frequent itemset are not important to measure the similarity of Chinese documents due to nearly same lengths of documents.

4.5.2. Evaluation of MC with normalization process

We also conduct comparative experiments to examine the performance of normalization process of MC. The normalization process of FTC is from hierarchical frequent term-based clustering [7]. The normalization process of CFWS is to combine the overlapping clusters [10]. The normalization process of FIHC is conducted by sibling merging [12]. For CMS, *k*-means is the normalization process. These experiments are conducted on English corpus (because English corpus has more categories than Chinese corpus) with minimum support as 0.01 and representation of individual words. The experimental results are shown in Tables 17–23.

We can see from Tables 17–19 that the document clusters produced by MC-1, MC-2 and MC-3 are exactly corresponding to the document categories in the data sets. Most document categories correspond to no more than two clusters. Especially on MC-3, document categories of classes 2, 3, 4, 5 and 6 are basically included in one document cluster.

For clustering results of FTC method shown in Table 20, documents of class 1, 2 and 3 are uniformly scattered over the document clusters. There is no evidence that FTC has powerful capacity to group documents in each category into a corresponding cluster. Moreover, many cluster overlaps are produced by HFTC in

Table 17
Clustering results of MC-1 in contrast to standard document classification.

MC-1	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Cluster 1	95	96	0	2	6	2
Cluster 2	1	5	6	13	197	13
Cluster 3	0	5	153	44	5	10
Cluster 4	4	1	4	8	45	164
Cluster 5	0	1	22	134	77	2
Cluster 6	2	3	4	45	0	144
Total	102	111	189	246	330	335

Table 18
Clustering results of MC-2 in contrast to standard document classification.

MC-2	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Cluster 1	1	3	14	16	164	6
Cluster 2	1	2	5	7	3	261
Cluster 3	94	99	1	1	5	13
Cluster 4	5	3	9	36	115	50
Cluster 5	0	3	13	116	42	1
Cluster 6	1	1	147	70	0	4
Total	102	111	189	245	329	335

Table 19
Clustering results of MC-3 in contrast to standard document classification.

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Cluster 1	91	11	0	12	19	100
Cluster 2	0	100	0	17	34	40
Cluster 3	0	0	173	10	27	12
Cluster 4	0	0	0	200	0	0
Cluster 5	0	0	0	0	209	0
Cluster 6	11	0	16	7	41	183
Total	102	111	189	246	330	335

Table 20
Clustering results of FTC method in contrast to standard document classification.

FTC	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Cluster 1	3	10	14	37	117	18
Cluster 2	25	18	22	43	27	53
Cluster 3	2	10	84	104	13	17
Cluster 4	50	36	36	40	49	99
Cluster 5	5	29	37	35	42	156
Cluster 6	32	17	20	41	105	10
Total	117	120	213	300	353	353

Table 21
Clustering results of CFWS method in contrast to standard document classification.

CFWS	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Cluster 1	0	0	0	0	0	3
Cluster 2	0	0	5	0	0	0
Cluster 3	0	0	0	3	0	0
Cluster 4	4	0	0	0	0	0
Cluster 5	0	0	0	0	0	0
Cluster 6	102	111	189	246	330	335
Total	106	111	194	249	330	338

Table 22
Clustering results of FIHC method in contrast to standard document classification.

FIHC	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Cluster 1	0	1	105	39	0	0
Cluster 2	79	2	2	1	1	0
Cluster 3	5	29	39	154	294	314
Cluster 4	0	1	2	6	23	2
Cluster 5	2	0	41	26	7	3
Cluster 6	16	78	0	20	5	16
Total	102	111	189	246	330	335

Table 23
Clustering results of CMS method in contrast to standard document classification.

CMS	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Cluster 1	75	62	128	155	170	204
Cluster 2	6	2	11	20	22	17
Cluster 3	5	28	12	24	41	33
Cluster 4	9	10	15	15	51	27
Cluster 5	12	5	18	20	21	13
Cluster 6	0	4	5	12	25	41
Total	102	111	189	246	329	335

normalizing small clusters into the predefined number in class 4, 5 and 6.

For clustering results of CFWS shown in Table 21, most documents are clustered into document cluster 6 due to serious overlapping of initial cluster candidates and co-occurrence transitivity discussed in Section 2.1.

For clustering results of FIHC shown in Table 22, the problem of skew clusters turns up because cluster 3 is overwhelmingly larger than other clusters. We have discussed this problem above in Section 2.4.

For clustering results of CMS shown in Table 23, it has two problems: first, too many documents are clustered into cluster 1; second, the remaining documents are uniformly scattered over other five clusters so there is no correspondence relationship between the produced clusters and natural categories.

4.5.3. Evaluation of MC in assigning topics of document clusters

Table 24 lists the topics of generated by the process of assigning topics mentioned in Section 3.2 in Maximum Capturing, for each document clusters produced MC-3, and the top 20 frequent individual words in the documents of natural categories on English data set.

We can see from Table 24 that there are no overlaps among the topics generated by Maximum Capturing despite that there is much overlap among the top 20 frequent words of each category. Moreover, the produced topics can actually identify document categories shown in Table 5, because most important words are captured in the topics. For instance, although "roaster" is not in the top 20 frequent words, it is highly relevant with topic "coffee" in documents. It is very interesting that "August" and "bank" construct a frequent itemsets and we found that in Reuters newswire, important things concerning "money" often happened in "August" such as "debt crisis", "base lending rate cut", etc. For categories which are very similar to each other such as "interest" and "money-fx", we found the items (words) in their most frequent itemsets are different, even if we can not easily differentiate them according to their top 20 frequent words.

5. Concluding remarks

In this paper, we conducted a study on document clustering using frequent itemsets. First, we present a review on methods of document clustering using frequent patterns. For each method, practical example is used to show the clustering mechanism of each method. We analyze their advantages and disadvantages in producing document clusters.

Second, MC is proposed to produce document clusters. It includes two procedures: constructing document clusters and assigning cluster topics. On constructing document clusters, our basic idea is to constrain document pairs which have the largest

Table 24
Topics generated by Maximum Capturing and most frequent words in each category.

Category	Topics produced by MC	Top 20 frequent individual words
Coffee	{Coffee, Roaster}	Coffee, export, quota, product, price, ico (international coffee organization), brazil, market, meet, mln, year, new, bag, international, consume, country, delegate, talk, Colombia, trade
Sugar	{Sugar, Ecuador}	Sugar, ton, mln, year, price, export, product, trader, product, import, beet, intervention, week, crop, cane, European, white, area, quota, market
Interest	{Rate, Pressure}	Rate, bank, pct, market, cut, billion, prime, point, year, week, dlr, effect, money, new, day, government, lend, today, lower, country
Money-fx	{Bank, August}	Bank, market, dollar, currency, rate, exchange, mln, pct, stg, billion, central, foreign, trade, west, monetary, Paris, treasury, economy, Baker, today
Trade	{Trade, Approval}	Trade, Japan, billion, dlr, year, export, Japanese, import, official, country, market, state, unit, tariff, foreign, govern, mln, deficit, agreement, Reagan
Crude	{Oil, OPEC}	Oil, price, barrel, dlr, mln, OPEC, crude, bpd, product, year, company, pct, market, Saudi, energy, industry, new, day, petroleum, output

similarity in same cluster. An extension of minimum spanning tree algorithm [8] is developed to produce document clusters. On assigning cluster topics, the most frequent itemsets of a cluster are used as the topics of that cluster and topic reduction process is used to eliminate overlaps of cluster topics. We also propose a normalization process for Maximum Capturing to merge clusters into a predefined number. Competitive learning is used here to reduce skew clusters in merging initial clusters.

Third, experiments are conducted to evaluate MC in comparison with CFWS, CMS, FTC and FIHC. An English corpus and a corpus are used in the experiments. The experimental results show that in document clustering, MC has better performance than other methods. It can group documents into clusters which appropriately correspond to the categories of documents. In assigning topics for clusters, topics produced by MC distinguish clusters from each other and have no overlaps. With the above analysis conclude that MC has favorable quality in clustering documents using frequent itemsets.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant Nos. 90718042, 60873072, and 60903050; the National Hi-Tech R&D Plan of China under Grant No. 2007AA010303, 2007AA01Z186, and 2007AA01Z179; the National Basic Research Program under Grant No. 2007CB310802. This work is also partially supported by the Foundation of Young Doctors of Institute of Software, Chinese Academy of Sciences, under Grant No. ISCAS2009-DR03. The authors would like to thank the anonymous reviewers for their help to improve this paper.

References

[1] O. Zamir, O. Etzioni, O. Madani, R.M. Karp, Fast and intuitive clustering of web documents, in: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, 1997, pp. 287–290.

[2] C.C. Aggarwal, S.G. Gates, P.S. Yu, On the merits of building categorization systems by supervised clustering, in: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999, pp. 352–356.

[3] B. Larson, C. Aone, Fast and effective text mining using linear-time document clustering, in: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, vol. 98(463), 1999, pp. 16–22.

[4] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in very large databases, in: Proceedings of the ACM SIGMOD Conference on Management of data, 1993, pp. 207–216.

[5] O.R. Zaiane, M.L. Antonie, Classifying text documents by associating terms with text categories, in: Proceedings of the 13th Australasian Database Conference, 2002, pp. 215–222.

[6] B. Liu, Wynne Hsu, Y.M. Ma, Integrating classification and association rule mining, in: Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1998, pp. 27–31.

[7] F. Beil, M. Ester, X.W. Xu, Frequent term-based text clustering, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 436–442.

[8] M.A. Harrison, Computer Algorithms: Introduction to Design and Analysis, Addison-Wesley Publishing Company, 1978. pp. 127–135.

[9] S.C. Ahalt, A.K. Krishnamurty, P. Chen, D.E. Melton, Competitive learning algorithms for vector quantization, Neural Networks 3 (1990) 277–291.

[10] Y.J. Li, S.M. Chung, J.D. Holt, Text document clustering based on frequent word meaning sequences, Data & Knowledge Engineering 64 (2008) 381–404.

[11] H. Edith, A.G. Rene, J.A. Carrasco-Ochoa, J.F. Martinez-Trinidad, Document clustering based on maximal frequent sequences, in: Proceedings of the FinTAL 2006, LNAI, vol. 4139, 2006, pp. 257–267.

[12] B. Fung, K. Wang, M. Ester, Hierarchical document clustering using frequent itemsets, in: Proceedings of the 3rd SIAM International Conference on Data Mining, 2003.

[13] K. Sparck Jones, A statistical interpretation of term specificity and its application in retrieval, Journal of Documentation 28 (1972) 11–21.

[14] W. Zhang, X.J. Tang, T. Yoshida, Improving effectiveness of mutual information for substantival multiword extraction, Expert Systems with Applications, 2009, doi:10.1016/j.eswa.2009.02.026.

[15] J.W. Han, J. Pei, Y.W. Yin, Mining frequent patterns without candidate generation, in: Proceedings of the 2000 ACM SIGMOD International Conference, 2000, pp. 1–12.

[16] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, in: KDD-2000 Workshop on Text Mining, 2000, pp. 109–110.

[17] J. Han, M. Kamber, Data Mining: Concepts and Techniques, second ed., Morgan Kaufmann Publishers, 2006.

[18] J. Wang, J. Han, BIDE: efficient mining of frequent closed sequences, in: Proceedings of the 20th International Conference on Data Engineering (ICDE'04), 2004, pp. 79–90.

[19] R.A. García-Hernández, J.F. Martínez-Trinidad, J.A. Carrasco-Ochoa, A fast algorithm to find all the maximal frequent sequence in a text, in: Proceedings of the CIARP 2004, LNCS, vol. 3287, 2004, pp. 478–486.