Knowledge-Based Systems 23 (2010) 379–388

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.



## 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-