Data & Knowledge Engineering 69 (2010) 1208–1226

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

Editorial

An integration of WordNet and fuzzy association rule mining for multi-label document clustering

Chun-Ling Chen [a], Frank S.C. Tseng [b,*], Tyne Liang [a]

[a] Department of Computer Science, National Chiao Tung University, HsinChu 300, Taiwan, ROC
[b] Dept. of Information Management, National Kaohsiung 1st University of Science & Technology, YanChao, Kaohsiung 824, Taiwan, ROC

ARTICLE INFO

Available online 25 September 2010

Keywords:
Fuzzy association rule mining
Text mining
Document clustering
WordNet
Frequent itemsets

ABSTRACT

With the rapid growth of text documents, document clustering has become one of the main techniques for organizing large amount of documents into a small number of meaningful clusters. However, there still exist several challenges for document clustering, such as high dimensionality, scalability, accuracy, meaningful cluster labels, overlapping clusters, and extracting semantics from texts. In order to improve the quality of document clustering results, we propose an effective Fuzzy-based Multi-label Document Clustering (FMDC) approach that integrates fuzzy association rule mining with an existing ontology WordNet to alleviate these problems. In our approach, the key terms will be extracted from the document set, and the initial representation of all documents is further enriched by using hypernyms of WordNet in order to exploit the semantic relations between terms. Then, a fuzzy association rule mining algorithm for texts is employed to discover a set of highly-related fuzzy frequent itemsets, which contain key terms to be regarded as the labels of the candidate clusters. Finally, each document is dispatched into more than one target cluster by referring to these candidate clusters, and then the highly similar target clusters are merged. We conducted experiments to evaluate the performance based on Classic, Re0, R8, and WebKB datasets. The experimental results proved that our approach outperforms the influential document clustering methods with higher accuracy. Therefore, our approach not only provides more general and meaningful labels for documents, but also effectively generates overlapping clusters.

1. Introduction

The incessant flourishing of Internet invigorates various textual documents to be shared over the cyberspace astonishingly. However, it also makes users suffer from the information-overloading problem. In particular, when users pose queries to WWW search engines, they usually bewilderingly receive a small number of relevant Web pages intermingled with a large number of irrelevant Web pages.

To effectively manage and organize the result of a search engine query, there inspires the study of document clustering techniques. The aim of this study is to automatically discover the hidden similarity and the key concepts of clustered documents for users to comprehend a large amount of documents. Over the past decades, several effective document clustering algorithms have been proposed to mitigate the hassle, including the *k*-means [1], Bisecting *k*-means [2], Hierarchical Agglomerative Clustering

* Corresponding author. Present/permanent address: 1, University Road, YanChao, Kaoshiung County, Taiwan 824, ROC.
E-mail addresses: chunling@cs.nctu.edu.tw (C.-L. Chen), imfrank@ccms.nkfust.edu.tw (F.S.C. Tseng), tliang@cs.nctu.edu.tw (T. Liang).

0169-023X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2010.08.003

(HAC) [3], and Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [4]. Nevertheless, as pointed out by [5–9], there are still challenges in improving the clustering quality, which we list as follows:

- *To cope with high dimensionality*: as the volume of textual document increases, the dimensionality of term features increases as well.
- *To improve the scalability*: many document clustering algorithms work fine on small document sets, but fail to deal with large document sets efficiently.
- *To promote the accuracy*: many existing document clustering algorithms require users to specify the number of clusters as an input parameter. However, it is difficult to determine the number of clusters in advance. Moreover, an incorrect estimation of the input parameter, i.e., the number of clusters, may lead to poor clustering accuracy [6].
- *To assign meaningful cluster labels*: meaningful cluster labels will guide users in the process of browsing the retrieved results. Thus, each cluster should be labeled with an understandable description. However, most of the traditional clustering algorithms do not provide labels for clusters.
- *To enable overlapping clusters*: many well-known clustering algorithms focus on hard clustering, where each document belongs to exactly one cluster. However, a document could contain multiple subjects. By using soft clustering algorithms [9], a document would appear in multiple clusters (i.e., overlapping clusters).
- *To extract semantics from text*: the bag-of-words representation used for clustering algorithms is often unsatisfactory as it ignores the conceptual similarity of terms that do not co-occur actually [5,7].

To resolve the problems of high dimensionality, large size, and understandable cluster description, Beil et al. [8] developed the first frequent itemsets-based algorithm, namely Hierarchical Frequent Term-based Clustering (HFTC), where the frequent itemsets are generated based on the association rule mining [10]. They only considered the low-dimensional frequent itemsets as clusters. Moreover, HFTC discovers overlapping clusters, which is useful for a search engine where overlapping clusters occur like Yahoo! Directory.

However, the experiments of Fung et al. [6] showed that HFTC is not scalable. For a scalable algorithm, Fung et al. proposed the FIHC (Frequent Itemset-based Hierarchical Clustering) algorithm by using frequent itemsets derived from association rule mining to construct a hierarchical topic tree for clusters. They also proved that using frequent itemsets for document clustering can reduce the dimensionality of term features effectively. Yu et al. [11] presented another frequent itemset-based algorithm, called TDC, to improve the clustering quality and scalability. This algorithm dynamically generates a topic directory from a document set using only closed frequent itemsets and further reduces the dimensionality. But, the clusters generated by FIHC and TDC are non-overlapping. In [12], the authors proposed that document clustering methods should provide multiple subjective perspectives onto the same document to enhance their practical applicability.

Recently, WordNet [13], one of the most widely adopted thesaurus for English, has been extensively used as an ontology in grouping documents with its semantic relations of terms [5,7,14,15]. Many existing document clustering algorithms mainly transform text documents into simplistic flat bags of document representation, i.e., term vectors or bags of keywords. Once terms are treated as individual items in such simplistic representation, the semantic content of a document is decomposed and cannot be reflected. Thus, Dave et al. [14] proposed using synsets as features for document representation and subsequent clustering. However, synsets decrease the clustering performance in all experiments without considering word sense disambiguation. Meanwhile, Hotho et al. [5] used WordNet in document clustering for word sense disambiguation to improve the clustering results. Jing et al. [15] presented another application of WordNet, which described how to find mutual information between terms by using the background knowledge through WordNet. In [7], Recupero proposed a new unsupervised document clustering method by using WordNet lexical and conceptual relations to allow common clustering algorithms to perform well. In this paper, the reasons of utilizing hypernyms from WordNet are two-fold:

(1) We intend to obtain more general and conceptual labels for derived clusters.
(2) From the experimental results in [14,16], the authors found that the performance of adding hypernyms is better than adding synonymy.

Among the techniques developed for data and text mining, association rule mining [10] is one of the useful and successful techniques for discovering interesting rules. It helps users discover meaningful association rules to represent a relationship between different pairs of a set of attribute values. The form of an association rule can be represented as $X \rightarrow Y$, where X and Y are sets of items and $X \cap Y = \emptyset$, and is usually adopted for market basket analysis to describe the following meaning: customers that buy product X also buy product Y for satisfying some predefined *minimum support value* and *minimum confidence value*. In general, each itemset has an associated measure of statistical significance called *Support* value, which is the fraction of all transactions that contain the itemset. For example, an itemset X with support value, $\mathrm{supp}(X) = 0.5$, regards there are 50% of transactions in the dataset containing X. An itemset can be chosen as a *frequent itemset* if its support value is larger than or equal to the predefined *minimum support value*. The *confidence* value of an association rule, denoted $\mathrm{conf}(X \rightarrow Y) = \mathrm{supp}(X \cup Y)/\mathrm{supp}(X)$, is to measure how often items in Y appear in transactions which also contain X. Finally, a rule $X \rightarrow Y$ will be discovered whether its confidence value is larger than or equal to the predefined *minimum confidence value* or not.

However, there are still two situations to be confronted, if we use association rule mining in our approach:

(1) Some important terms that express the topics of a document may be rarely appeared in the document collection. That is, only the terms which frequently occur in the document collection can be obtained, which implies the important sparse terms may be obscured in the process of document clustering.

(2) Association rule mining often suffers from producing too many itemsets, especially when items in the dataset are highly correlated [17]. As our approach aims to consider the semantic relationships from WordNet, the situation may become severer after adding correlated hypernyms.

Considering the above two issues, we will propose an approach which stems from prior studies [18–20], by integrating fuzzy set concept [21] and association rule mining to provide significant dimensionality reduction over interesting frequent itemsets. Moreover, Kaya et al. [20] think that fuzzy association rule mining is understandable to humans because it integrates linguistic terms with fuzzy sets. By applying fuzzy association rule mining, we can discover fuzzy frequent itemsets as candidate clusters, like ($term_1$.*Low*, $term_2$.*High*) or ($term_1$.*Low*, $term_2$.*Low*), and label the terms with a linguistic term, like *Low*, *Mid*, or *High*.

In this paper, we extend our previous study [22] and further propose an effective Fuzzy-based Multi-label Document Clustering (FMDC) approach based on fuzzy association rule mining in conjunction with WordNet for clustering textual documents. In contract with our previous study, this paper illustrates how to utilize the *α-cut* concept in the process of document clustering to solve the overlapping clusters problem. The advantages of FMDC approach are listed as follows:

1. It presents a means of dynamically deriving a hierarchical organization of concepts from WordNet based on the content of each document without using training data or standard clustering techniques;
2. It extends the fuzzy data representation used in data mining by Hong et al. [18] to text mining to discover the generalized fuzzy frequent itemsets as the candidate clusters;
3. It provides an accurate measure of confidence, and adopts the *α-cut* concept to assign each document to one or more than one target cluster. Given a fuzzy set A in the universe of discourse $X=\{x_1, x_2,\ldots, x_n\}$ and any number $\alpha\in[0,1]$, the α-cut is denoted as $A_\alpha=\{x_i|\mu_A(x_i)\geq\alpha, x_i\in X\}$, where μ_A is the membership function of the fuzzy set A and it converts x_i into a membership value in the closed interval [0,1]. That is, the α-cut is the crisp set that contains all the elements of X whose membership values given by μ_A are greater than or equal to the specified value of α.
4. It can automatically determine the number of clusters by the minimum support threshold. There is no need to specify the number of clusters as an input parameter.
5. By conducting experimental evaluations on the four datasets of Classic, Re0, R8, and WebKB, it has been proven that our approach outperforms the influential document clustering methods with higher accuracy. Besides, our approach not only provides more general and meaningful labels for documents, but also generates overlapping clusters.

The subsequent sections of this paper are organized as follows. In Section 2, we review the contemporary document clustering algorithms. In Section 3, our approach will be described, together with an illustrative example. The conducted experiment will be described and the results analyzed in Section 4. Finally, we conclude and propose some future directions in Section 5.

2. Related work

The basic principle of document classification is to classify or group a set of unlabeled documents into classes or clusters. According to [2], we divide document classification into three subcategories, i.e., supervised or unsupervised, hard or soft, and partitioning, hierarchical, or frequent itemset-based. These subcategories can be shown in a tree structure as Fig. 1 depicts, which we describe as follows.

1. Supervised and Unsupervised (Clustering): in supervised document classification, a set of predefined classes are available. On the other hand, in unsupervised document classification, also called document clustering, there are no pre-determined classes available. Document clustering is the process of calculating document similarities to form clusters. The documents within a cluster are similar to each other and, simultaneously, dissimilar to the documents in the other groups.

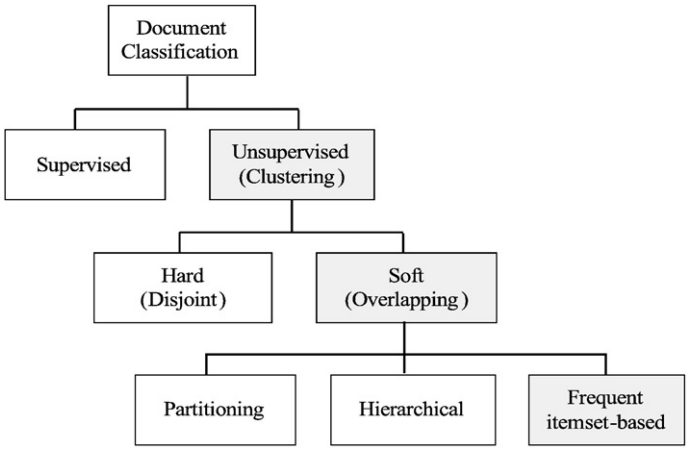


Fig. 1. A tree structure with three types of document classification.

Table 1
Summary for our approach and the other document clustering algorithms.

Authors	Problem addressed	Clustering concept	Semantic discovery	Overlapping clusters	Meaningful cluster label
Lin and Kondadadi (2001) [9]	Clustering efficiency	Soft document clustering	No	Yes	No
Beil et al. (2002) [8]	Clustering accuracy	Frequent itemset-based	No	Yes	Yes
Hotho et al. (2003) [5]	Semantic analysis for text	Partitioning	Yes	No	No
Fung et al. (2003) [6]	Quality of cluster in large document set	Frequent itemset-based	No	No	Yes
Sedding and Kazakov (2004) [16]	Semantic analysis for text	Partitioning	Yes	No	No
Yu et al. (2004) [11]	A topic directory construction and clustering accuracy	Frequent itemset-based	No	No	Yes
Wang and Hogdges (2006) [23]	Semantic analysis for text	Partitioning	Yes	No	No
Recupero (2007) [7]	Semantic analysis for text	Partitioning	Yes	No	No
Chen et al. (2009) [22]	Semantic analysis for text	Frequent itemset-based	Yes	No	Yes
Our approach (FMDC)	Clustering accuracy	Frequent itemset-based	Yes	Yes	Yes

2. Hard (Disjoint) and Soft (Overlapping): hard clustering algorithms compute the hard assignment (i.e., each document is assigned to exactly one cluster) and produce a set of disjoint clusters. Soft clustering algorithms compute the soft assignment (i.e., each document allows to appear in multiple clusters) and generate a set of overlapping clusters. For instance, a document discussing "Natural language and Information Retrieval" should be assigned to both of the clusters "Natural language" and "Information Retrieval".
3. Partitioning, Hierarchical, and Frequent itemset-based: for document clustering, partitioning-based methods exclusively partition the set of documents into a number of clusters by moving documents from one cluster to another, such as *k*-means [1] and Bisecting *k*-means [2]. Hierarchical-based document clustering is to build a hierarchical tree of clusters, whose leaf nodes represent the subset of a document collection, like Hierarchical Agglomerative Clustering (HAC) [3] and Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [4]. Besides, a new category of document clustering, namely "frequent itemset-based clustering," has been extensively developed, including FIHC [6], HFTC [8], TDC [11], and F^2IDC [22]. Frequent itemset-based clustering methods use frequent itemsets generated by the association rule mining and further cluster the documents according to these extracted frequent itemsets. These methods reduce the dimensionality of term features efficiently for very large datasets, thus they can improve the accuracy and scalability of the clustering algorithms. An advantage of frequent itemset-based clustering method is that each cluster can be labeled by the obtained frequent itemsets shared by the documents in the same cluster. Moreover, the organization of clusters generated by frequent itemset-based clustering methods could be a flat set or a hierarchical tree of clusters.

In this paper, our FMDC approach falls into the category of unsupervised, soft, and frequent itemset-based method to cluster documents with higher accuracy and creates a flat set of clusters. Table 1 summarizes the characteristics of our approach and the other influential document clustering algorithms.

3. The framework of FMDC approach

Fig. 2 shows the proposed FMDC (Fuzzy-based Multi-label Document Clustering) framework, which consists of four modules, namely *Document Analysis Module*, *TermOnto Construction Module*, *Candidate Clusters Extraction Module*, and *Overlapping Clusters Generation Module* as explained in Sections 3.1, 3.2, 3.3, and 3.4, respectively. In this framework, when receiving a set of textual documents, our first module will extract and select the key term set, and then the second module organizes it into a term forest by referring to WordNet for generating the Document Set *D*. The third module implements our fuzzy association rule mining procedure to generate the candidate cluster set. Finally, the last module constructs the Document-Cluster Matrix to produce the target clusters. The whole process will be illustrated by a comprehensive example presented in Section 3.5.

3.1. Document analysis module

There are two stages in the first module, namely *Key Term Extraction* and *Key Term Selection*, for reducing the dimensionality of the source document set:

1. *Key Term Extraction*: the whole extraction process is as follows:
   (1) First of all, each document is broken into sentences. Then, terms in each sentence are extracted as features. In this paper, a term is regarded as the stem of a single word.
   (2) The terms appeared in a predefined stop word list[1] are removed.
   (3) Remained terms are converted to their base forms by stemming. The terms with the same stem are combined for frequency counting. Finally, the frequency of each term in each document is recorded.

[1] It contains a list of 571 stop words that was developed by the SMART project.

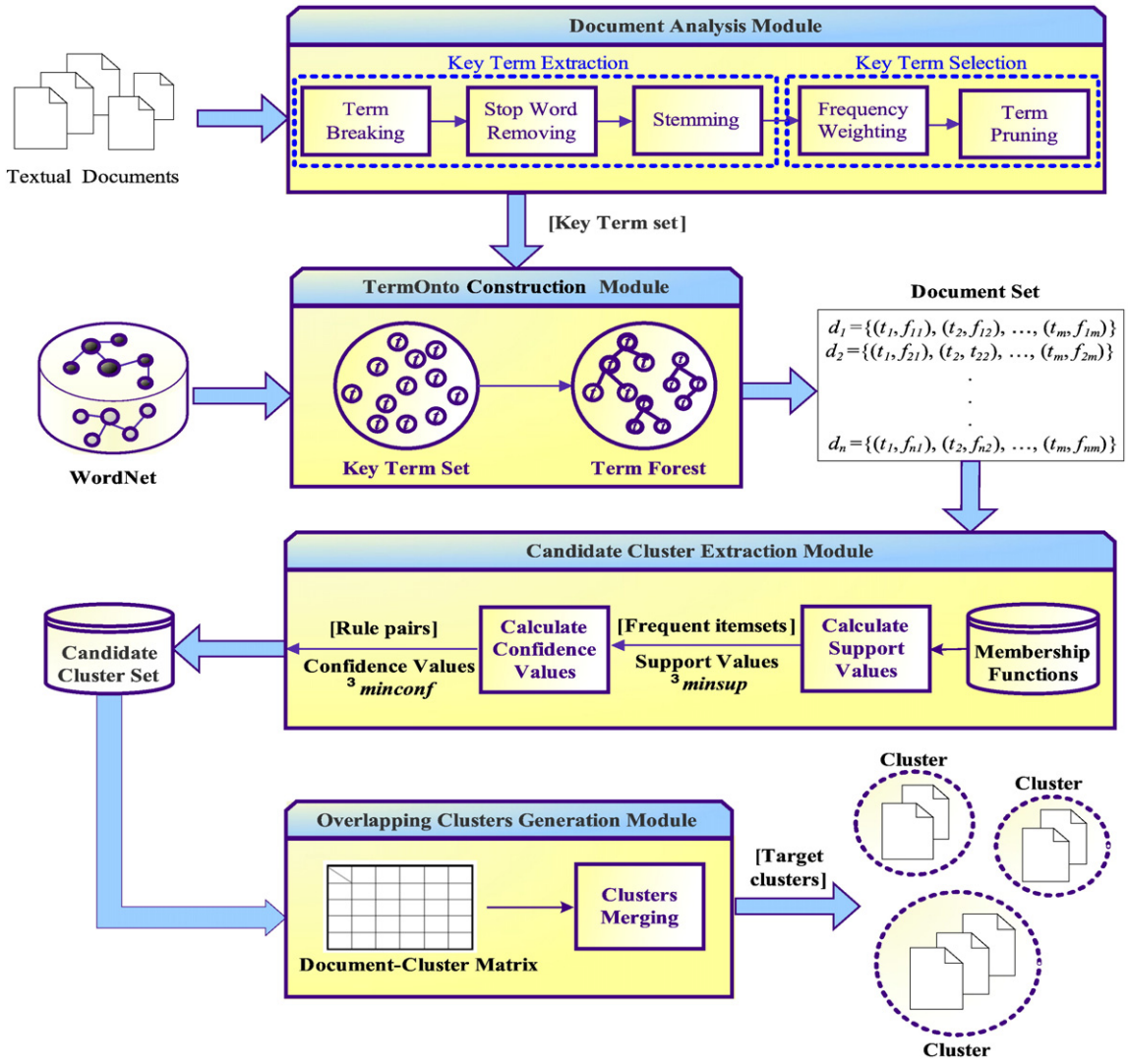


Fig. 2. The FMDC framework.

2. *Key Term Selection*: we understand that terms of low frequencies are supposed as noise and useless for identifying the appropriate cluster. Thus, we apply the tf–idf (term frequency × inverse document frequency) method to choose the key terms for the document set. A term will be discarded if its weight is less than a fixed tf–idf threshold γ. Formula (3.1) is used for the measurement of $tfidf_{ij}$ for the importance of a term t_j within a document d_i. In Formula (3.1), f_{ij} is the frequency of t_j in d_i, and $\max_{t_j \in d_i}(f_{ij})$ is the maximum frequency of all terms in d_i used for normalization to prevent bias for long documents.

$$tfidf_{ij}=0.5+0.5*\frac{f_{ij}}{\max_{t_j\in d_i}(f_{ij})}\times log\left(1+\frac{|D|}{|\{d_i|t_j\in d_i, d_i\in D\}|}\right) \tag{3.1}$$

After the weight of each term in each document has been calculated, those which satisfy the pre-specified minimum tf–idf threshold γ are retained. Subsequently, these retained terms form a set of key terms for the document set D, and we formally define them in Definitions 3.1–3.4.

Definition 3.1 (Document). A *document*, denoted $d_i=\{(t_1,f_{i1}), (t_2,f_{i2}),\ldots, (t_j,f_{ij}),\ldots, (t_m,f_{im})\}$, is a logical unit of text, characterized by a set of key terms t_j together with their corresponding frequency f_{ij}.

Definition 3.2 (Document Set). A *document set*, denoted $D=\{d_1, d_2,\ldots, d_i,\ldots, d_n\}$, also called a *document collection*, is a set of documents, where n is the total number of documents in D.

Definition 3.3 (Term Set). The *term set* of a document set $D=\{d_1, d_2,..., d_i,..., d_n\}$, denoted $T_D=\{t_1, t_2,..., t_j,..., t_s\}$, is the set of terms appeared in D, where s is the total number of terms and t_j is the stem of a single word.

Definition 3.4 (Key Term Set). The *key term set* of a document set $D=\{d_1, d_2,..., d_i,..., d_n\}$, denoted $K_D=\{t_1, t_2,..., t_j,..., t_m\}$, is a subset of the term set T_D, including only meaningful key terms, which do not appear in a well-defined stop word list, and satisfy the predefined minimum threshold of the tf–idf method.

3.2. TermOnto construction module

The objective of the second module is based on the usage of WordNet for generating a richer document representation of the given document set. As the relationships of relevant terms have been predefined in WordNet, in this module, we intend to use the hypernyms provided by WordNet as useful features for document clustering.

After key terms are extracted from the document set, they can be organized based on the hierarchical (IS–A) relationship of WordNet [13] to construct term trees. A term tree is constructed by matching a key term in WordNet and then navigating upwards for five levels of hypernyms. Eventually, all term trees can be regarded as a term forest for the document set D, which we formally define as follows.

Definition 3.5 (Term Tree). A *term tree* of term t, denoted $\mathcal{J}=(W, H, I, t)$, is a 4-tuple consisting of a set of hypernyms $I=\{h_1,..., h_r\}$ of a key term $t_j \in W$, together with their reference function $H: 2^{|W|} \rightarrow 2^I$ in W, where W represents the WordNet and H links the set of hypernyms up to five levels in W. We denote $h_1 \leq h_2$, when h_2 is the hypernym of h_1 defined in W.

Definition 3.6 (Term Forest). A *term forest* of a set of key terms $\{t_1, t_2, \ldots, t_i, \ldots, t_m\}$, denoted $\mathcal{F}=\{\mathcal{J}_1, \mathcal{J}_2,..., \mathcal{J}_i,..., \mathcal{J}_m\}$, is a set of term trees, where m is the total number of key terms in D.

Using hypernyms can help our approach magnify hidden similarities to identify related topics, which potentially leads to better clustering quality [5,16]. For example, a document talking about 'sale' may not be associated with a document about 'trade' by the clustering algorithm, if there are only 'sale' and 'trade' in the key term set. But, if a more general term 'commerce' is added to both documents, their semantic relationship can be revealed.

Hence, we enriched the representation of each document with hypernyms based on WordNet to find semantically-related documents. Based on the key terms appeared in a document, the representation of this document is enriched by associating them with the term trees accordingly. By simply combining these expanded hypernyms, we obtain a new key term set $K_D=\{t_1, t_2,..., t_m,$

Algorithm 1. Basic algorithm to obtain the designated representation of all documents

Input: A document set D; A well-defined stop word list; WordNet W; The minimum tf-idf threshold γ.

Output: The formal representation of all documents in D.

1. Extract the term set $T_D = \{t_1, t_2,..., t_j,..., t_s\}$
2. Remove all stop words from T_D
3. Apply Stemming for T_D
4. For each $d_i \in D$ do //key term selection
   For each $t_j \in T_D$ do
   (1) Evaluate its $tfidf_{ij}$ weight // defined by Formula (4.1) in Section 4
   (2) Retain the term if $tfidf_{ij} \geq \gamma$
5. Form the key term set $K_D = \{t_1, t_2,..., t_j,..., t_m\}$, where $m \leq s$
6. For each $t_j \in K_D$ do //refer to W
   $\mathcal{J}_j = (W, H, I, t_j)$ // find the set of hypernyms $I = \{h_1,..., h_r\}$ and their links H
7. Form the Term Forest $\mathcal{F} = \{\mathcal{J}_1, \mathcal{J}_2, ..., \mathcal{J}_i, ..., \mathcal{J}_m\}$
8. For each $d_i \in D$ do //document enrichment step
   For each $t_j \in K_D$ do
   (1) If (h_j is hypernyms of t_j) then //refer to W
   (a) $hf_{ij} \rightarrow hf_{ij} + f_{ij}$
   (2) If (h_j is not in K_D) then
   (b) $K_D \rightarrow K_D \cup \{h_j\}$
9. For each $d_i \in D$ do //in order to decrease noise from hypernyms , tf-idf method is executed again
   For each $t_j \in K_D$ do
   (1) Evaluate its $tfidf_{ij}$ weight
   (2) Retain the term if $tfidf_{ij} \geq \gamma$
10. Form the new key term set $K_D = \{t_1, t_2,..., t_m, h_1,..., h_r\}$
11. For each $d_i \in D$, record the frequency f_{ij} of t_j and the frequency hf_{ij} of h_j in d_i to obtain the final representation of $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}),..., (t_m, f_{im}), (h_1, hf_{i1}),..., (h_r, hf_{ir})\}$

Fig. 3. The detailed description of Algorithm 1.

$h_1,\ldots, h_r\}$, where $h_1,\ldots, h_r$ are newly-added hypernyms derived from WordNet. Thus, the enriched document d_i now can be extended into $d_i=\{(t_1,f_{i1}),(t_2,f_{i2}),\ldots,(t_m,f_{im}),(h_1,hf_{i1}),\ldots,(h_r,hf_{ir})\}$. Notice that the weight of some key terms may be 0 if they do not appear in d_i. The frequency hf_{ij} of hypernym h_j is a value accumulated from the frequencies of its descent terms appearing in d_i.

We use Algorithm 1, as shown in Fig. 3, to generate the extended representation of each document for later mining process.

3.3. Candidate clusters extraction module

After the above processes, documents are converted into structured term vectors. Then, the fuzzy data mining algorithm is executed to generate fuzzy frequent itemsets and output a candidate clusters set. In the following, we define the membership functions in Section 3.3.1 and present our fuzzy association rule mining algorithm for texts in Section 3.3.2.

3.3.1. The membership functions

Each pair (t_j, f_{ij}) of a document d_i can be transformed into a fuzzy set $F_{ij}=w_{ij}^{Low}/t_j.Low + w_{ij}^{Mid}/t_j.Mid + w_{ij}^{High}/t_j.High$ with its frequency being represented by three fuzzy regions, namely *Low*, *Mid*, and *High*, to depict its grade of membership within d_i. Each fuzzy value w_{ij}^r has a corresponding membership function, denoted $w_{ij}^r(f_{ij})$, to convert the key tem frequency f_{ij} into a value of the range [0, 2], where r can be *Low*, *Mid*, and *High*, and the corresponding membership functions $w_{ij}^r(f_{ij})$ are defined by Formulas (3.2), (3.3), and (3.4), respectively. The derived membership functions are shown in Fig. 4.

$$w_{ij}^{Low}\left(f_{ij}\right)=\begin{cases}0, f_{ij}=0 & a=0,\\ 1+f_{ij}-a\big/_{b-a},\ a\le f_{ij}\le b & b=\min\left(f_{ij}\right),\\ 2,\ b<f_{ij}<c & c=\dfrac{\lceil avg\left(f_{ij}\right)+\min\left(f_{ij}\right)\rceil}{2},\\ 1+f_{ij}-d\big/_{c-d},\ c\le f_{ij}\le d & d=avg\left(f_{ij}\right)\\ 1, f_{ij}>d & \end{cases} \tag{3.2}$$

$$w_{ij}^{mid}\left(f_{ij}\right)=\begin{cases}0, f_{ij}=0 & a=min\left(f_{ij}\right),\\ 1, f_{ij}<a & \\ 1+f_{ij}-a\big/_{b-a},\ a\le f_{ij}\le b & b=\dfrac{\lceil avg\left(f_{ij}\right)+\min\left(f_{ij}\right)\rceil}{2}\\ 2,\ b<f_{ij}<c & c=avg(f_{ij}),\\ 1+f_{ij}-d\big/_{c-d},\ c\le f_{ij}\le d & \\ 1, f_{ij}>d & d=avg(f_{ij})+\dfrac{\lceil max\left(f_{ij}\right)-\mathrm{avg}\left(f_{ij}\right)\rceil}{4}\end{cases} \tag{3.3}$$

$$w_{ij}^{high}\left(f_{ij}\right)=\begin{cases}0, f_{ij}=0 & a=avg(f_{ij}),\\ 1, f_{ij}<a & \\ 1+f_{ij}-a\big/_{b-a},\ a\le f_{ij}\le b & b=avg(f_{ij})+\dfrac{\lceil max\left(f_{ij}\right)-\mathrm{avg}\left(f_{ij}\right)\rceil}{4},\\ 2,\ b<f_{ij}<c & c=avg(f_{ij})+\dfrac{\lceil max\left(f_{ij}\right)-\mathrm{avg}\left(f_{ij}\right)\rceil}{2},\\ 1+f_{ij}-d\big/_{c-d},\ c\le f_{ij}\le d & d=max(f_{ij})\end{cases} \tag{3.4}$$

In Formulas (3.2), (3.3), and (3.4), $min(f_{ij})$ is the minimum frequency of terms in D, $max(f_{ij})$ is the maximum frequency of terms in D, and $avg(f_{ij})=\frac{\left\lceil\sum_{i=1}^{n} f_{ij}\right\rceil}{|K|}$, where $f_{ij}\neq min(f_{ij})$ or $max(f_{ij})$, and $|K|$ is the number of summed key terms.

3.3.2. The fuzzy association rule mining algorithm for texts

To generate the *target cluster set* $C_D=\left\{c_1^1, c_2^1,\ldots, c_l^q,\ldots c_f^q\right\}$ for a document set D, a *candidate cluster set* $\tilde{C}_D=\left\{\tilde{c}_1^1,\ldots,\tilde{c}_{l-1}^2,\tilde{c}_l^q,\ldots,\tilde{c}_k^q\right\}$, where k is the total number of candidate clusters, will be generated after the mining process. We call each c_i^q as a *target cluster* in the following. A *candidate cluster* $\tilde{c}=(\tilde{D}_c,\tau)$ is a two-tuple, where $\tilde{D}_c$ is a subset of D, such that it includes those documents which contain all the key terms in $\tau=\{t_1,t_2,\ldots,t_q\}\subseteq K_D$, $q\ge 1$, where K_D is the key term set of D, and q is the number of key terms contained in τ. In fact, τ is a fuzzy frequent itemset for describing $\tilde{c}$. To illustrate, $\tilde{c}$ can also be denoted as $\tilde{c}^q_{(t_1,t_2,\ldots,t_q)}$ or $\tilde{c}^q_{(\tau)}$, and will be used interchangeably hereafter. For instance, the candidate cluster $\tilde{c}^1_{(trade)}=(\{d_2,d_3\},\{\text{trade}\})$ means the term "trade" appeared in documents d_2 and d_3.

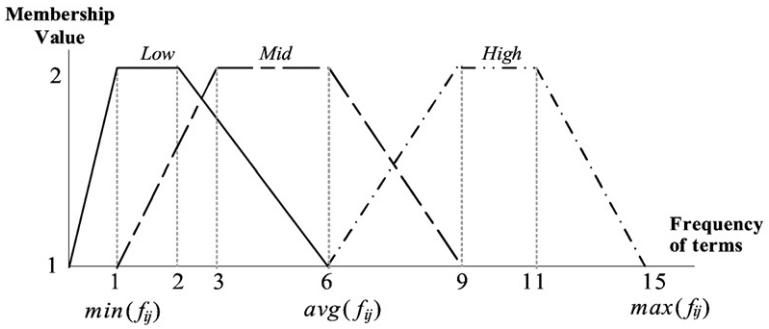


Fig. 4. The predefined membership functions.

Algorithm 2. Basic algorithm to obtain the fuzzy frequent itemsets from the document set.

Input: A set of documents $D = \{d_1, d_2, \ldots, d_n\}$, where $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \ldots, (t_j, f_{ij}), \ldots, (t_m, f_{im})\}$; A set of membership functions (as defined in Section 3.2.1); The minimum support value θ; The minimum confidence value λ.

Output: A set of candidate cluster.

1. For each $d_i \in D$ do
   For each $t_j \in d_i$ do
   (1) $f_{ij} \rightarrow F_{ij} = w_{ij}^{Low}/t_j.Low + w_{ij}^{Mid}/t_j.Mid + w_{ij}^{High}/t_j.High$ //using membership functions
2. For each $t_j \in K_D$ do
   For each $d_i \in D$ do
   (1) $count_j^{Low} = \sum_{i=1}^{n} w_{ij}^{Low}, count_j^{Mid} = \sum_{i=1}^{n} w_{ij}^{Mid}, count_j^{High} = \sum_{i=1}^{n} w_{ij}^{High}$
3. For each $t_j \in K_D$ do
   (1) $max\text{-}count_j = max(count_j^{Low}, count_j^{Mid}, count_j^{High})$
4. $L_1 = \{max\text{-}R_j | support(t_j) = \frac{max\text{-}count_j}{|D|} \geq \theta, 1 \leq j \leq m\}$ // $|D|$ is the number of documents.
5. For $(q = 2; L_{q-1} \neq \varnothing; q{+}{+})$ do // Find fuzzy frequent q-itemsets L_q
   (1) $C_q =$ **apriori_gen**(L_{q-1}, θ) // similar to the *a priori* algorithm
   (2) For each candidate q-itemsets τ with key terms $(t_1, t_2, \ldots, t_q) \in C_q$ do
      (a) For each $d_i \in D$ do
         $w_{i\tau} = min\{w_{ij}^{max\text{-}R_j} | j = 1, 2, \ldots, q\}$ // $w_{ij}^{max\text{-}R_j}$ is the fuzzy membership value of the maximum region of t_j in d_i.
      (b) $count_\tau = \sum_{i=1}^{n} w_{i\tau}$
   (3) $L_q = \{\tau \in C_q | support(\tau) = \frac{count_\tau}{|D|} \geq \theta, 1 \leq j \leq q\}$
6. For all the fuzzy frequent q-itemsets τ containing key terms $(t_1, t_2, \ldots, t_q)$, where $q \geq 2$ do // construct the strong fuzzy frequent itemsets
   (1) form all possible association rules
   $$\tau_1 \wedge \ldots \wedge \tau_{k-1} \wedge \tau_{k+1} \wedge \ldots \wedge \tau_q \rightarrow \tau_k, \; k = 1 \text{ to } q.$$
   (2) Calculate the confidence values of all possible association rules
   $$confidence(\tau) = \frac{\sum_{i=1}^{n} w_{i\tau}}{\sum_{i=1}^{n} (w_{i1} \wedge \ldots \wedge w_{ik-1}, w_{ik+1} \wedge \ldots \wedge w_{iq})}$$
   (3) $\tilde{C}_D = \{\tau \in L_q | confidence(\tau) \geq \lambda\}$
7. $\tilde{C}_D \rightarrow \{L_1\} \cup \tilde{C}_D$

Procedure **apriori_gen**(L_{q-1}, θ)
1. for each itemset $l_1 \in L_{q-1}$ do
   for each itemset $l_2 \in L_{q-1}$ do
   (1) if ($l_1[1] = l_2[1]$ $l_1[2] = l_2[2]$ ... $l_1[k\text{-}2] = l_2[k\text{-}2]$ $l_1[k\text{-}1] = l_2[k\text{-}1]$) then
      $C_q = \{c | c = l_1 \; l_2\}$
2. Return C_q

Fig. 5. The detailed description of Algorithm 2.

$$W = \begin{array}{c} \\ d_1 \\ d_2 \\ \vdots \\ d_n \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & \cdots & t_p \end{array} \\ \begin{bmatrix} w_{11}^{\max-R_j} & w_{12}^{\max-R_j} & \cdots & w_{1p}^{\max-R_j} \\ w_{21}^{\max-R_j} & w_{22}^{\max-R_j} & \cdots & w_{2p}^{\max-R_j} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1}^{\max-R_j} & w_{n2}^{\max-R_j} & \cdots & w_{np}^{\max-R_j} \end{bmatrix}_{n\times p} \end{array}$$

Fig. 6. A formal illustration of Document-Term Matrix.

Algorithm 2 shown in Fig. 5 generates fuzzy frequent itemsets based on predefined membership functions and the minimum support value θ, from a large textual document set, and obtains a candidate cluster set according to the minimum confidence value λ. Since each discovered fuzzy frequent itemset has an associated fuzzy count value, it can be regarded as the degree of importance that the itemset contributes to the document set.

In Algorithm 2, two confidence values of a rule pair is used to measure the strength of association among the key terms $(t_1, t_2, \ldots, t_q)$ of the fuzzy frequent q-itemsets. We take the candidate cluster $\tilde{c}^2_{(sale,\ trade)}$ as an example. Since its confidence values of the rule pair "If sale = *Low*, then trade = *Mid*" and "If trade = *Mid*, then sale = *Low*" are both greater than the minimum confidence value λ, $\tilde{c}^2_{(sale,\ trade)}$ is put in the candidate cluster set $\tilde{C}_D$. Finally, the candidate cluster set $\tilde{C}_D$ will be output.

3.4. Overlapping clusters generation module

The objective of the last module is to assign each document to multiple clusters $\{\tilde{c}^q_1, \ldots, \tilde{c}^q_i\}$, where $i \geq 1$ and $q \geq 1$. For assigning documents to the target clusters, each candidate cluster $\tilde{c}^q_{(\tau)} = \tilde{c}^q_{(t_1, t_2, \ldots, t_q)}$ with fuzzy frequent itemset τ is considered in the clustering process. The τ will be regarded as a reference point for generating a target cluster. In order to represent the degree of importance of a document d_i in a candidate cluster $\tilde{c}^q_l$, an $n \times k$ Document-Cluster Matrix (DCM) will be constructed to calculate the similarity of terms in d_i and $\tilde{c}^q_l$. To achieve this goal, we define Document-Term Matrix and Term-Cluster Matrix by Definitions 3.7 and 3.8, respectively. Based on these definitions, we can further define the Document- Cluster Matrix (DCM) of a document set D by Definition 3.9.

Definition 3.7 (Document-Term Matrix, DTM). A *Document-Term Matrix* (*DTM*), denoted $W = \left[w_{ij}^{max-Rj}\right]$, for a document set D, is an $n \times p$ matrix, such that $w_{ij}^{\max-R_j}$ is the weight (fuzzy membership value of the maximum region) of term t_j in document d_i and $t_j \in L_1$ and can be calculated from the Steps 4 and 5 of Algorithm 2. A formal illustration of DTM can be found in Fig. 6.

Definition 3.8 (Term-Cluster Matrix, TCM). A *Term-Cluster Matrix* (*TCM*) for a document set $D = \{d_1, d_2, \ldots, d_n\}$ is a $p \times k$ matrix, such that $1 \leq j \leq p$, $1 \leq l \leq k$, defined as $G = \left[g_{jl}^{max-R_j}\right]$, where

$$g_{jl}^{max-R_j} = \frac{score(\tilde{c}^q_l)}{\sum_{i=1}^{n} w_{ij}^{max-R_j}} \qquad \text{where} \qquad \left\{ \begin{array}{l} score(\tilde{c}^q_l) = \sum_{d_i \in \tilde{c}^q_l, t_j \in L_1} w_{ij}^{\max-R_j}, \text{and} \\ w_{ij}^{max-R_j} = \text{the weight(fuzzy membership value of the maximum region) of term } t_j \text{ in document } d_i \in \tilde{c}^q_l. \end{array} \right\} \tag{3.5}$$

$$G = \begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_p \end{array} \begin{array}{c} \begin{array}{cccccc} \tilde{c}^1_1 & \cdots & \tilde{c}^1_{l-1} & \tilde{c}^q_l & \cdots & \tilde{c}^q_k \end{array} \\ \begin{bmatrix} g_{11}^{\max-R_j} & \cdots & g_{1l-1}^{\max-R_j} & g_{1l}^{\max-R_j} & \cdots & g_{1k}^{\max-R_j} \\ g_{21}^{\max-R_j} & \cdots & g_{2l-1}^{\max-R_j} & g_{2l}^{\max-R_j} & \cdots & g_{2k}^{\max-R_j} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g_{p1}^{\max-R_j} & \cdots & g_{pl-1}^{\max-R_j} & g_{pl}^{\max-R_j} & \cdots & g_{pk}^{\max-R_j} \end{bmatrix}_{p\times k} \end{array}$$

Fig. 7. A formal illustration of Term-Cluster Matrix.

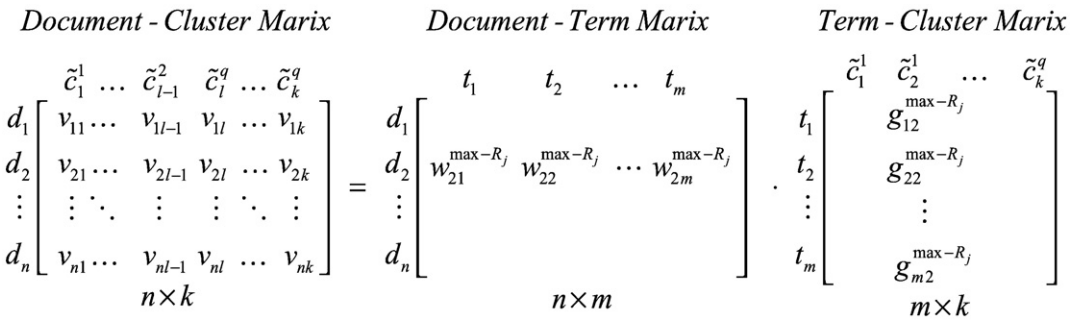


Fig. 8. Document-Cluster Matrix.

A formal illustration of TCM can be found in Fig. 7. Each entry $g_{jl}^{\max - R_j}$ represents the degree of importance of key term t_j in a candidate cluster $\tilde{c}_l^q$.

Definition 3.9 (Document-Cluster Matrix, DCM). A *Document-Cluster Matrix* (*DCM*) for a document set $D = \{d_1, d_2, \ldots, d_n\}$ is an $n \times k$ matrix $V = [v_{il}]$ defined by the inner product of its DTM and TCM, where

$$v_{il} = row_i(W) \cdot col_1(G) = \left[w_{i1}^{max-R_j} w_{i2}^{max-R_j} \cdots w_{ip}^{max-R_j} \right] \begin{bmatrix} g_{1l}^{\max-R_j} \\ g_{2l}^{\max-R_j} \\ \vdots \\ g_{pl}^{\max-R_j} \end{bmatrix} = \sum_{p=1}^{p} w_{ip} g_{pl},\ 1 \le i \le n \text{ and } 1 \le l \le k.$$

A formal illustration of DCM can be found in Fig. 8.

Then, based on DCM, we apply intersection of fuzzy set theory to compute the membership degree of each document in one candidate cluster with the other candidate clusters. Hence, we define one matrix, namely Multiple Clusters Matrix (MCM), as follows.

Definition 3.10 (Multiple Clusters Matrix, MCM). A *Multiple Clusters Matrix* (*MCM*), denoted $M = [m_{ig}]$, is an $n \times C_k^2$ matrix, such that $m_{ig} = min\{m_{il}, m_{ij}\}$ is the membership degree of document d_i in intersection of two candidate clusters $\tilde{c}_l^q \cap \tilde{c}_j^q$, where $l, j \in \{1, 2, \ldots, k\}$, $l \neq j$, and $q = 1$. A formal illustration of MDM can be found in Fig. 9.

Moreover, we apply the α-cut threshold [21,23,24] determined by Formula (3.6) to evaluate the minimum value which satisfies the restrictive condition, and it can appropriately provide flexibility to overlapping clusters.

$$\alpha < \min_{1 \le g \le C_2^k} \left\{ \max_{1 \le i \le n} \left[m_{ig} \right] \right\} \tag{3.6}$$

Then, based on the obtained DCM, an unassigned document d_i might belong to more than one target cluster by using Formula (3.7).

$$c_l^q = \{ d_i \mid v_{il} > max\{(\rho - \alpha), \alpha\} \text{where } \rho = max\{v_{i1}, v_{i2}, \ldots, v_{ik}\} \in \tilde{c}_l^q \} \tag{3.7}$$

$$M = \begin{array}{c} \begin{matrix} \tilde{c}_1^1 \cap \tilde{c}_2^1 & \tilde{c}_1^1 \cap \tilde{c}_3^1 & \cdots & \tilde{c}_{k-1}^1 \cap \tilde{c}_k^1 \end{matrix} \\ \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1C_k^2} \\ m_{21} & m_{22} & \cdots & m_{2C_k^2} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nC_k^2} \end{bmatrix}_{n \times C_k^2} \end{array} \begin{array}{l} d_1 \\ d_2 \\ \vdots \\ d_n \end{array}$$

Fig. 9. A formal illustration of Multiple Clusters Matrix.

Finally, to avoid low clustering accuracy, the inter-cluster similarity between two target clusters c_x^q and c_y^q, $c_x^q \neq c_y^q$, is calculated to merge the small target cluster into the similar target cluster. The inter-cluster similarity measurement between two target clusters is defined by Formula (3.8).

$$Inter\_Sim\left(c_x^q, c_y^q\right) = \frac{\sum_{i=1,\ d_i \in c_x^q, c_y^q}^{n} v_{ix} \times v_{iy}}{\sqrt{\sum_{i=1,\ d_i \in c_x^q}^{n} (v_{ix})^2 \times \sum_{i=1,\ d_i \in c_y^q}^{n} \left(v_{iy}\right)^2}} \tag{3.8}$$

where v_{ix} and v_{iy} stand for two entries, such that $d_i \in c_x^q$ and $d_i \in c_y^q$, in DCM, respectively. The range of *Inter-Sim* is [0, 1]. If the *Inter-Sim* value is close to 1, then both clusters are regarded nearly the same. In the following, the minimum *Inter-Sim* will be used as a threshold δ to decide whether two target clusters should be merged.

Algorithm 3 shown in Fig. 10 is used to assign each document to the fitting target clusters, and finally builds a target cluster set for output.

3.5. An illustrative example

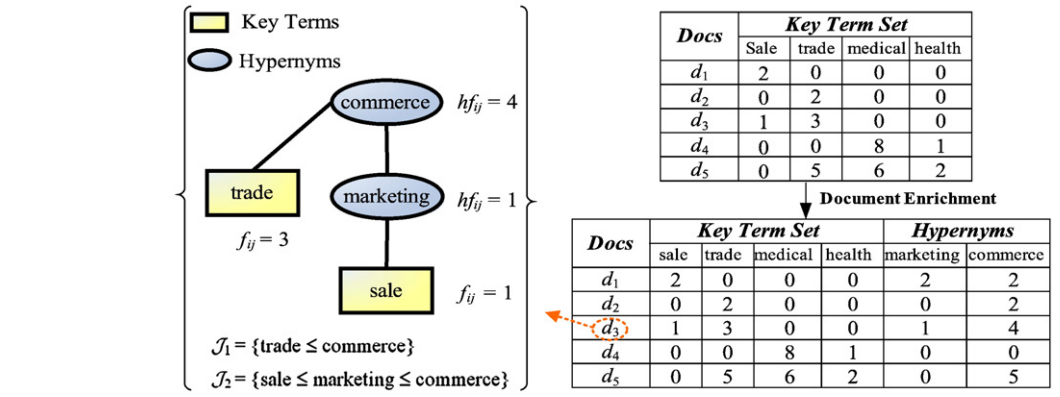
Suppose we have a document set $D = \{d_1, d_2, \ldots, d_5\}$ and its key term set $K_D = \{$sale, trade, medical, health$\}$. Fig. 11 illustrates the process of Algorithm 1 to obtain the representation of all documents. Notice that we use a tabular representation, where each

Algorithm 3. Basic algorithm to obtain the target clusters

Input: A document set $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$; The key term set $K_D = \{t_1, t_2, \ldots, t_j, \ldots, t_m\}$;
The candidate cluster set $\tilde{C}_D = \{\tilde{c}_1^1, \ldots, \tilde{c}_{l-1}^1, \tilde{c}_l^q, \ldots, \tilde{c}_k^q\}$; A minimum *Inter-Sim* threshold δ;
Output: The target cluster set $C_D = \{c_1^1, c_2^1, \ldots, c_i^q, \ldots, c_f^q\}$

1. Build $n \times p$ document-term matrix $W = \left[w_{ij}^{\max - R_j}\right]$. // $w_{ij}^{\max - R_j}$ is the weight (fuzzy value) of t_j in d_i and $t_j \in L_1$.
2. Build $p \times k$ term-cluster matrix $G = \left[g_{jl}^{\max - R_j}\right]$. // $g_{jl}^{\max - R_j} = \frac{score(\tilde{c}_l^q)}{\sum_{i=1}^{n} w_{ij}^{\max - R_j}}$, $1 \le j \le p$, $1 \le l \le k$,
   and, $score(\tilde{c}_l^q) = \sum_{d_i \in \tilde{c}_l^q, t_j \in \tau} w_{ij}^{\max - R_j}$, where $w_{ij}^{\max - R_j}$ is the weight (fuzzy value) of t_j in d_i and $t_j \in L_1$.
3. Build $n \times k$ document-cluster matrix $V = W \cdot G = [v_{il}] = \sum_{p=1}^{p} w_{ip} g_{pl}$.
4. Build $n \times C_k^2$ multiple clusters matrix $M = [m_{ig}]$
5. Decide the α-cut threshold $\alpha < \min_{1 \le g \le C_k^2} \left\{ \max_{1 \le i \le n} \left[m_{ig}\right] \right\}$
4. Based on V, assign d_i to target cluster s
   $c_l^q = \left\{ d_i \,\middle|\, v_{il} > \max\{(\rho - \alpha),\ \alpha\} \text{ where } \rho = \max\{v_{i1}, v_{i2}, \ldots, v_{ik}\} \in \tilde{c}_l^q \right\}$
6. Clusters merging
   (1) For each $c_l^q \in C_D$ do
      (a) If (c_l^q = null) then { remove this target clusters c_l^q from C_D }
   (2) For each pair of target clusters $(c_x^q, c_y^q) \in C_D$ do
      (a) Calculate the *Inter_sim*
      (b) Store the results in the Inter -Cluster Similarity matrix I.
   (3) If (one of the *Inter_sim* value in $I \ge \delta$) then
      (a) Select (c_x^q, c_y^q) with the highest *Inter_sim*.
      (b) Merge the smaller target cluster into the larger target cluster .
      (c) Repeat Step (2) to update I
7. Output C_D

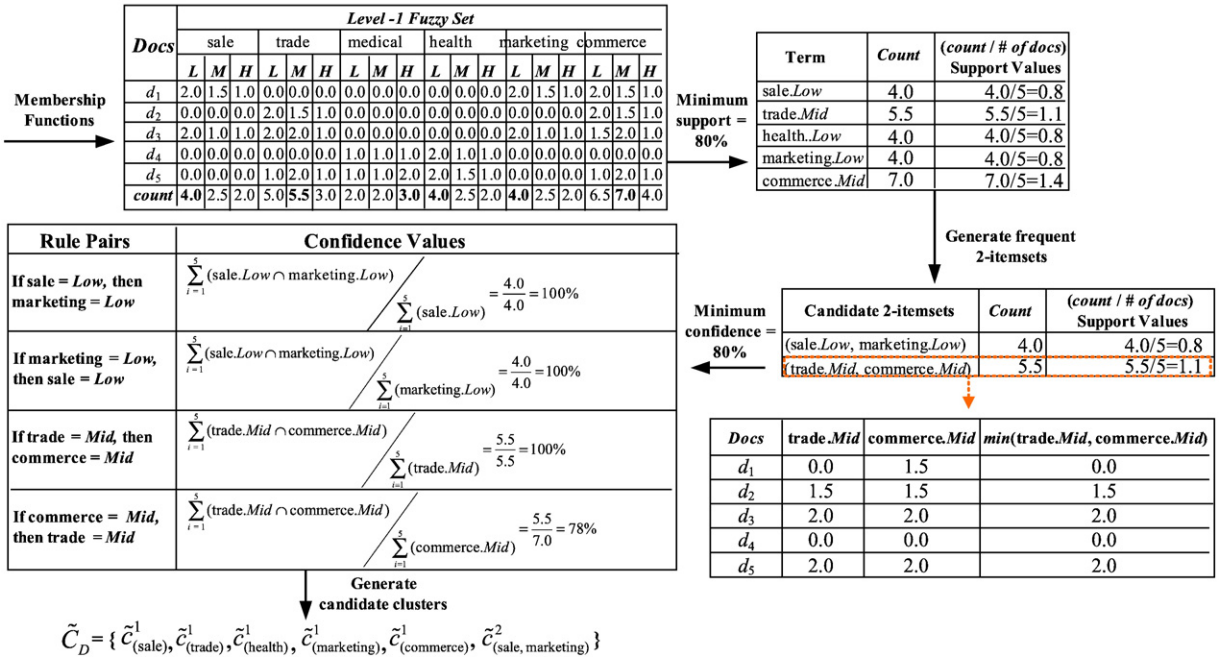
Fig. 10. The detailed description of Algorithm 3.



Docs	Key Term Set			
	Sale	trade	medical	health
d_1	2	0	0	0
d_2	0	2	0	0
d_3	1	3	0	0
d_4	0	0	8	1
d_5	0	5	6	2

Docs	Key Term Set				Hypernyms	
	sale	trade	medical	health	marketing	commerce
d_1	2	0	0	0	2	2
d_2	0	2	0	0	0	2
d_3	1	3	0	0	1	4
d_4	0	0	8	1	0	0
d_5	0	5	6	2	0	5

Fig. 11. The process of Algorithm 1 of this example.



Docs	Level -1 Fuzzy Set																	
	sale			trade			medical			health			marketing			commerce		
	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
d_1	2.0	1.5	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	1.5	1.0	2.0	1.5	1.0
d_2	0.0	0.0	0.0	2.0	1.5	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	1.5	1.0
d_3	2.0	1.0	1.0	2.0	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	1.0	1.0	1.5	2.0	1.0
d_4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	2.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
d_5	0.0	0.0	0.0	1.0	2.0	1.0	1.0	1.0	2.0	2.0	1.5	1.0	0.0	0.0	0.0	1.0	2.0	1.0
count	**4.0**	2.5	2.0	5.0	**5.5**	3.0	2.0	2.0	**3.0**	**4.0**	2.5	2.0	**4.0**	2.5	2.0	6.5	**7.0**	4.0

Term	Count	(count / # of docs) Support Values
sale.*Low*	4.0	4.0/5=0.8
trade.*Mid*	5.5	5.5/5=1.1
health..*Low*	4.0	4.0/5=0.8
marketing.*Low*	4.0	4.0/5=0.8
commerce.*Mid*	7.0	7.0/5=1.4

Candidate 2-itemsets	Count	(count / # of docs) Support Values
(sale.*Low*, marketing.*Low*)	4.0	4.0/5=0.8
(trade.*Mid*, commerce.*Mid*)	5.5	5.5/5=1.1

Rule Pairs	Confidence Values
If sale = *Low*, then marketing = *Low*	$\sum_{i=1}^{5}(\text{sale.}Low \cap \text{marketing.}Low) \Big/ \sum_{i=1}^{5}(\text{sale.}Low) = \frac{4.0}{4.0} = 100\%$
If marketing = *Low*, then sale = *Low*	$\sum_{i=1}^{5}(\text{sale.}Low \cap \text{marketing.}Low) \Big/ \sum_{i=1}^{5}(\text{marketing.}Low) = \frac{4.0}{4.0} = 100\%$
If trade = *Mid*, then commerce = *Mid*	$\sum_{i=1}^{5}(\text{trade.}Mid \cap \text{commerce.}Mid) \Big/ \sum_{i=1}^{5}(\text{trade.}Mid) = \frac{5.5}{5.5} = 100\%$
If commerce = *Mid*, then trade = *Mid*	$\sum_{i=1}^{5}(\text{trade.}Mid \cap \text{commerce.}Mid) \Big/ \sum_{i=1}^{5}(\text{commerce.}Mid) = \frac{5.5}{7.0} = 78\%$

Docs	trade.*Mid*	commerce.*Mid*	*min*(trade.*Mid*, commerce.*Mid*)
d_1	0.0	1.5	0.0
d_2	1.5	1.5	1.5
d_3	2.0	2.0	2.0
d_4	0.0	0.0	0.0
d_5	2.0	2.0	2.0

Fig. 12. The process of Algorithm 2 of this example.

entry denotes the frequency of a key term (the column heading) in a document d_i (the row heading), to make our presentation more concise. Moreover, rectangle nodes represent actual key terms appearing in the document set; spheroid nodes represent newly-added hypernyms. In this example, the key term 'sale' has the parent nodes 'marketing' and 'commerce'. Similarly, 'trade' and 'marketing' have the same parent node 'commerce'.

Consider the representation of all documents generated from Fig. 11, the membership functions defined in Fig. 4, the minimum support value 80%, and the minimum confidence value 80% as inputs. The fuzzy frequent itemsets discovery procedure is depicted in Fig. 12.

Moreover, consider the candidate cluster set $\tilde{C}_D$ was already generated in Fig. 12. Now, suppose the minimum *Inter-Sim* value is 0.5. Fig. 13 illustrates the process of Algorithm 3, together with the final results.

4. Experimental evaluation

In this section, we experimentally evaluated the performance of the proposed algorithm by comparing with that of FIHC, *k*-means, Bisecting *k*-means, and UPGMA algorithms. We make use of the FIHC 1.0 tool[2] to generate the results of FIHC. Moreover, Steinbach et al. [2] compared the performance of some influential clustering algorithms, and the results indicated that UPGMA and

[2] http://ddm.cs.sfu.ca/dmsoft/Clustering/products/.

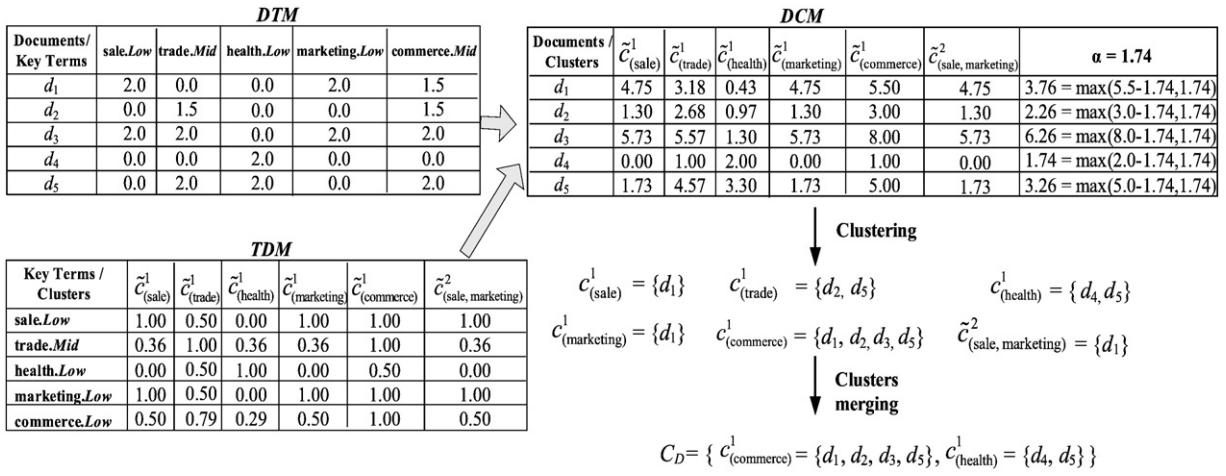


Fig. 13. The process of Algorithm 3 of this example.

Bisecting *k*-means are the most accurate clustering algorithms. Therefore, the CLUTO-2.1.2a[3] Clustering tool is applied to generate the results of *k*-means,[4,5] Bisecting *k*-means,[6] and UPGMA.[7] The produced results are then fetched into the same evaluation program to ensure a fair comparison. All the experiments were performed on a P4 3.2 GHz Windows XP machine with 1 GB memory. The implementation of our approach was written with Java 1.5 to allow reusability of the written code.

4.1. Datasets

To test the proposed approach, we used four different kinds of datasets: Classic, Re0, R8, and WebKB, which are widely adopted as standard benchmarks for the text categorization task. Moreover, these datasets are not specially designed to combine with WordNet for facilitating the clustering result. Table 2 summarizes the statistics of these datasets. The detailed information of these datasets is described as follows:

- Classic[8]: this document set is a combination of the four classes CACM, CISI, CRAN, and MED abstracts. Classic dataset includes 3203 CACM documents, 1460 CISI documents from information retrieval papers, 1398 CRANFIELD documents from aeronautical system papers, and 1033 MEDLINE documents from medical journals.
- *Re0*[9]: *Re0* is a text document dataset, derived from Reuters-21578 text categorization test collection Distribution 1.0. *Re0* includes 1504 documents with 13 classes.
- R8[10]: R8 is a subset of the Reuters-21578[11] text categorization collections. It considers only the documents associated with a single topic and includes 7674 documents with 8 most frequent classes.
- WebKB[12]: this dataset consists of web pages collected by the WebKB project of the CMU text learning group [25]. These pages are manually classified into seven categories: Student, Faculty, Staff, Department, Course, Project, and Other. In our test, we select the four most popular entity-representing categories: course, faculty, project, and student.

4.2. The evaluation metric

In these datasets, each document is pre-classified into single category, i.e., natural class. The class information is utilized in the evaluation method for measuring the accuracy of the clustering result. In our test, the standard evaluation measure, namely Overall *F*-measure [6], is used to evaluate the generated clustering results. The evaluation measure is widely used to evaluate the performance of clustering algorithms.

[3] http://glaros.dtc.umn.edu/gkhome/views/cluto/.
[4] The command is vcluster -clmethod = direct -crfun = i2 -sim = cos -rowmodel = maxtf -colmodel = idf -clabelfile = <X>.mat.clabel<X>.mat<K>.
[5] <X> is the name of the dataset being tested (ex. R8, WebKB etc.), and<K> is the number of clusters desired in the final solution. Vcluster is the name of the Cluto clustering program that clusters data from .mat files as input.
[6] The command is vcluster -clmethod = agglo -crfun = upgma -sim = cos -rowmodel = maxtf -colmodel = idf -clabelfile = <X>.mat.clabel<X>.mat<K>.
[7] The command is vcluster -clmethod = rbr -crfun = i2 -sim = cos –cstype = best -rowmodel = maxtf -colmodel = idf -clabelfile = <X>.mat.clabel<X>.mat<K>.
[8] ftp://ftp.cs.cornell.edu/pub/smart/.
[9] The preprocessed datasets can be downloaded. http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download/.
[10] The preprocessed datasets can be downloaded. http://web.ist.utl.pt/~acardoso/datasets/.
[11] http://www.daviddlewis.com/resources/testcollections/.
[12] The preprocessed datasets can be downloaded. http://www.cs.technion.ac.il/~ronb/thesis.html.

Table 2
Statistics for our test datasets.

Datasets	Documents	Classes	Class size			Document length
	Total	Total	Max	Average	Min	Average
Classic	7094	4	3203	1774	1033	39
Re0	1504	13	608	116	11	69
R8	7674	8	3923	959	51	48
WebKB	4199	4	1641	1050	504	124

Document clustering is a process of partitioning a set of documents into a set of meaningful subclasses, called clusters. Hence, we define a set of document clusters generated from clustering results, denoted C, and another set is natural classes, denoted L, which each document is pre-classified into a single class. Both sets are derived from the same document set D. Let $|D|$ be the number of all documents in the document set D; $|c_i|$ be the number of documents in the cluster $c_i \in C$; $|l_j|$ be the number of documents in the class $l_j \in L$; $|c_i \cap l_j|$ be the number of documents both in a cluster c_i and a class l_j. Then, the two standard evaluation measures are defined as follows.

4.2.1. Overall F-measure

The F-measure is often employed to evaluate the accuracy of clustering results. Fung et al. [6] measured the quality of a clustering result C using the weighted sum of such maximum F-measures for all natural classes according to the cluster size. This measure is called the overall F-measure of C, denoted $F(C)$, which is defined as follows.

$$F(C) = \sum_{l_j \in L} \frac{|l_j|}{|D|} \max_{c_i \in C} \{F\}, \text{ where } F = \frac{2PR}{P+R}, \mathrm{P} = \frac{|\mathrm{c_i} \cap l_j|}{|\mathrm{c_i}|} \text{ and } R = \frac{|\mathrm{c_i} \cap l_j|}{|l_j|} \tag{4.1}$$

In general, the higher values of $F(C)$ indicate the better clustering quality. Notice that overall F-Measure favors for the hard assignment generated by clustering algorithms [26]. In order to demonstrate the performance of our approach, we present experiments in which we generated hard assignment (this has been called *hardening* the clusters) and then evaluated the output of our algorithm. The hardening scheme is simply performed by assigning each document to the cluster which has a maximum membership degree among all the document clusters. Thus, it can be employed to evaluate the performance of our approach by comparing with the other hard clustering methods. Thus, we use overall F-Measure to evaluate the clustering quality of FMDC and the other compared algorithms.

4.3. Parameters selection

Table 3 summarizes the parameters for our proposed method and the other algorithms to compare the clustering performance. Since k-means, Bisecting k-means, and UPGMA may generate different clustering results each time with randomly chosen initial value. Therefore, the final result of these three algorithms is an average from five runs performed on a given dataset.

Before applying FMDC, we first consider the feature selection strategy. In order to select the most representative features, we use Formula (3.1) to obtain the key terms with weights higher than the predefined thresholds γ. Table 4 shows the keyword

Table 3
List of all parameters for our algorithms and the other four algorithms.

Parameter name	FMDC	FIHC	k-means	Bi. k-means	UPGMA
Datasets	Classic, Re0, R8, WebKB				
Stop word removal	Yes				
Stemming	Yes				
Length of the smallest term	Three				
Weight of the term vector	TF	tf–idf	tf–idf	tf–idf	tf–idf
Levels of hypernyms	h_1, h_2, h_3, h_4, h_5				
Cluster count k	5, 10, 15, 30, 45, 60, 80, 100				

Table 4
Keyword statistics of our test datasets.

Data set	# of terms	# of terms after pre-processing	# of terms after enriching	γ threshold	
				FMDC without WordNet	FMDC with WordNet
Classic	40,291	40,279	41,931	0.60	0.65
Re0	2886	2678	3507	0.60	0.65
R8	16,810	16,790	18,692	0.60	0.65
WebKB	42,503	34,310	36,622	0.60	0.65

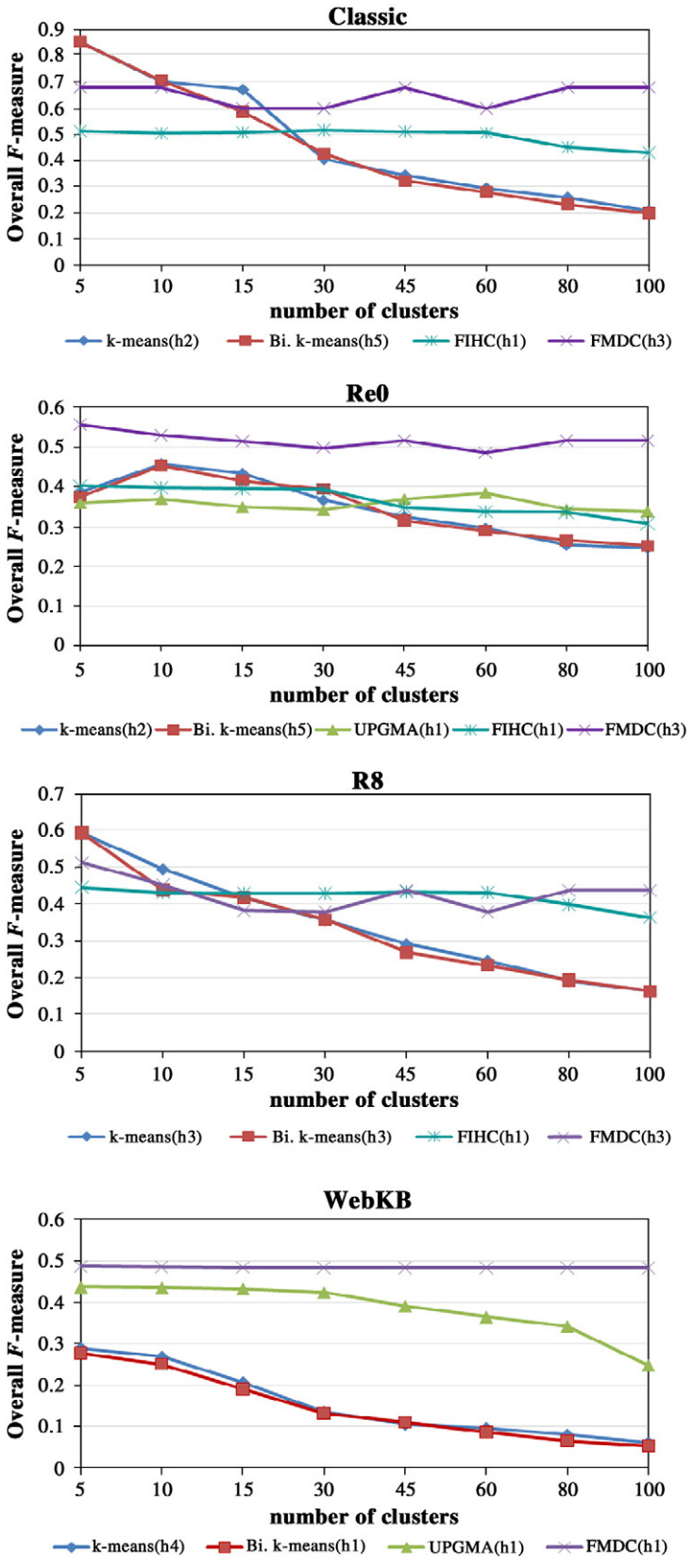


Fig. 14. Overall F-measure comparison for five clustering algorithms on the four datasets.

Table 5
Average overall F-measure comparison for five clustering algorithms on the four datasets.

Datasets	FMDC(h)	FIHC(h)	k-means(h)	Bi. k-means (h)	UPGMA(h)
Classic	0.65(3)*	0.49(1)	0.47(2)	0.45(5)	N.A.
Re0	0.53(3)*	0.36(1)	0.35(2)	0.34(5)	0.36(1)
R8	0.44(3)*	0.42(1)	0.34(3)	0.33(3)	N.A.
WebKB	0.48(1)*	N.A.	0.16(4)	0.15(1)	0.38(1)

N.A. means not scalable to run * means the best competitor.

statistics of our test datasets and the suggested thresholds for each dataset. Documents were then represented as TF (Term Frequency) vectors, and unimportant terms were discarded. This process implies a significant dimensionality reduction without loss of clustering performance.

The two algorithms, FMDC and FIHC, all have two main parameters for the adjustment of accuracy quality. This first one is mandatory and is denoted MinSup, which means the minimum support for frequent itemsets generation. The other one is optional, and is denoted KCluster, which represents the number of clusters.

4.4. Experimental results

The experiments were conducted by the following steps. First, we evaluated our approach, FMDC, on the four selected datasets described in Section 4.1 and compared its accuracy with that of FIHC, the standard k-means, Bisecting k-means, and UPGMA. Second, we verified if the use of WordNet can improve the clustering accuracy on these compared algorithms and generated conceptual labels for the derived clusters. Third, the dataset Reuters was chosen to evaluate the efficiency and scalability of FMDC.

4.4.1. Comparison of FMDC with other algorithms

Fig. 14 presents the obtained overall F-Measure values for FMDC and the other algorithms by comparing eight different numbers of clusters on four datasets. For each algorithm, we run each dataset enriched with the top 5 levels of hypernyms. We tested each algorithm's clustering results with the value h, the levels of hypernyms, from 1 to 5 and selected the best results. We chose the MinSup threshold from the elements in {25%, 28%, 30%, 32%, and 35%} to run FMDC with WordNet for all datasets. Moreover, we use the minimum support, ranging from 3% to 6% for FIHC for all datasets. Notice that UPGMA is not available for large data sets because some experimental results cannot be generated for UPGMA. Since FIHC is not available for the documents of long average length, there is no experimental result generated on the WebKB dataset.

By Table 5, it is obvious that the average overall F-measure values of FMDC with WordNet are superior to that of the other algorithms on all datasets. Although the average accuracy of Bisecting k-means and FIHC shown in Fig. 14 are slightly better than that of the FMDC in several cases. We argue that the exact number of clusters in a document set is usually unknown in real case, and FMDC is robust enough to produce stable, consistent and high quality clusters for a wide range number of clusters. This can be realized by observing the average overall F-measure values of all test cases. From Fig. 14, we also observed that the clustering accuracy of k-means, Bisecting k-means, and UPGMA are sensitive when the number of clusters changes. These algorithms require users to specify the number of clusters as an input parameter, which may imply poor clustering accuracy when we input an incorrect parameter [6].

4.4.2. The effect of the enriched document representation

As described in the second module of our approach, when enriching the document representation, we use the hypernyms from WordNet as useful features for clustering. We demonstrate the effect of adding hypernyms in our approach. In the following, all algorithms are tested by the baseline method and the addition of hypernyms of various levels.

Table 6 shows the average overall F-measure results obtained by all algorithms on Classic and Re0 datasets. The results for R8 and WebKB datasets are shown in Table 7. In Tables 6 and 7, "Baseline" means that no hypernyms are added; "h_1" corresponds to the addition of direct hypernyms; "h_2" stands for the addition of hypernyms of first and second levels, and so on. We chose the minimum support values, ranging from 4% to 8%, to run the baseline result of FMDC for all datasets. The evaluation results in Tables 6 and 7 confirm that the average overall F-measure values of WordNet-based FMDC performance are superior to that of the

Table 6
The effect of enriching the document representation on Classic and Re0 datasets.

Dataset	Classic					Re0				
	FMDC	FIHC	k-means	Bi. k-means	UPGMA	FMDC	FIHC	k-means	Bi. k-means	UPGMA
Baseline	0.48	0.47	0.45	0.46	N.A.	0.55	0.38	0.36	0.35	0.40
h_1	0.63	**0.49**	0.46	0.44	N.A.	0.52	**0.36**	0.34	**0.34**	**0.36**
h_2	0.64	0.49	**0.47**	0.44	N.A.	0.52	0.35	**0.35**	0.34	0.35
h_3	**0.65**	0.48	0.47	**0.45**	N.A.	**0.53**	0.36	0.35	0.34	0.35
h_4	0.61	0.45	0.45	0.44	N.A.	0.51	0.36	0.35	0.34	0.35
h_5	0.62	0.45	0.45	0.45	N.A.	0.51	0.36	0.33	0.34	0.35

N.A. means not scalable to run boldface entries highlight the best competitor in each column from h_1 to h_5 (the row headings).

Table 7
The effect of enriching the document representation on R8 and Webkb datasets.

Dataset	R8					Webkb				
	FMDC	FIHC	*k*-means	Bi. *k*-means	UPGMA	FMDC	FIHC	*k*-means	Bi. *k*-means	UPGMA
Baseline	0.53	0.52	0.35	0.34	N.A.	0.43	N.A.	0.15	0.15	0.35
h_1	0.36	**0.42**	**0.34**	**0.33**	N.A.	**0.48**	N.A.	0.15	**0.15**	**0.38**
h_2	0.37	0.41	0.34	0.33	N.A.	0.43	N.A.	0.15	0.14	0.38
h_3	**0.44**	0.37	0.34	0.33	N.A.	0.37	N.A.	0.15	0.14	0.38
h_4	0.43	0.37	0.33	0.33	N.A.	0.33	N.A.	**0.16**	0.14	0.38
h_5	0.43	0.36	0.33	0.32	N.A.	0.33	N.A.	0.15	0.14	0.38

N.A. means not scalable to run boldface entries highlight the best competitor in each column from h_1 to h_5 (the row headings).

Table 8
Cluster labels generated by FMDC algorithm on Re0 dataset.

FMDC without WordNet	FMDC with WordNet
Bank, dollar, currency, growth, industry market, nation, rate, rise, rose, sell, and trade	Activity, agent, assemblage, commerce, (commodity, good), currency, deficiency, forecast, growth, merchant, nation, part, rate, and record, (bush, rose, and shrub)

other algorithms when adding hypernyms of the first, second, and third levels on almost all datasets, except for WebKB dataset. The performance of FMDC with the addition of direct hypernyms is better than that of FMDC with higher levels of hypernyms on WebKB dataset. Due to the longer average length of documents in WebKB dataset, we think that higher levels of hypernyms may add more noise to the clustering process and decrease the clustering accuracy.

From Tables 6 and 7, the use of WordNet for FMDC induces better clustering results at least 5% higher than the other algorithms on Classic and WebKB datasets, particularly the improvement of Classic dataset. However, adding hypernyms may not be beneficial for the clustering task. The reason is that using hypernyms as additional features in the document enrichment process inevitably introduces a lot of noise into these datasets. In contrast to the other WordNet-based algorithms, our approach can ameliorate the effect of adding hypernyms by filtering out noise for clustering on Classic and WebKB datasets.

However, comparing with the baseline method, the use of WordNet decreases the clustering accuracy on Re0 and R8 datasets for our approach and the other compared algorithms. For the obtained results, the reasons could be:

(1) It is not likely to work well for text, such as documents in Reuters-21578, which is guaranteed to be written in concise and efficiently [27].
(2) Word sense disambiguation was not performed to determine the proper meaning of each polysemous term in documents [5].

To understand the reason why WordNet enhanced FMDC to perform better, a sample of the cluster labels generated by FMDC without WordNet and FMDC with WordNet on Re0 dataset can be found in Table 8. Thanks to the rich semantic network representation provided by WordNet, FMDC generates more general and meaningful labels for clusters. For example, the label 'commerce' produced by FMDC with WordNet is a more general concept than the labels 'sell' and 'trade' generated by FMDC without WordNet.

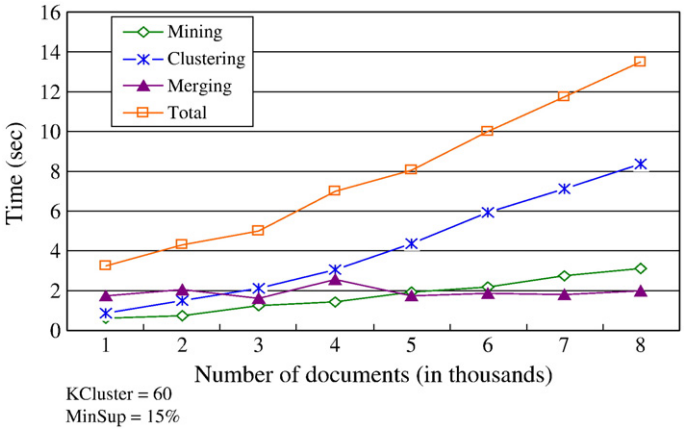


Fig. 15. The detailed time cost analysis of FMDC on Reuters dataset.

4.4.3. Efficiency and scalability

Our algorithm, FMDC, involves three major phases: finding fuzzy frequent itemsets, initial clustering, and clusters merging. Fig. 15 shows the scalabilities of FMDC on different sizes of Reuters datasets, ranging from 1 K to 8 K documents.

5. Conclusion and future work

The importance of document clustering emerges from the massive volumes of textual documents created. Although numerous document clustering methods have been extensively studied in these years, there still exist several challenges for increasing the clustering quality. Particularly, most of the current document clustering algorithms, including FIHC, do not consider the semantic relationships among the terms nor search an organization of documents into overlapping clusters. In this paper, we derived a fuzzy-based document clustering approach that combines fuzzy association rule mining with WordNet to alleviate these problems. In the total processes, we begin with the process of document pre-processing and further enrich the initial representation of all documents by using hypernyms of WordNet in order to exploit the semantic relations between terms. Then, fuzzy association data mining algorithm automatically generates fuzzy frequent itemsets and regards them as candidate clusters. Finally, each document is dispatched into more than one cluster by referring to these candidate clusters, and then highly similar clusters are merged.

The key advantage conferred by our proposed algorithm is that the generated clusters, labeled with conceptual terms, are easier to understand than clusters annotated by isolated terms. In addition, the extracted cluster labels may help for identifying the content of individual clusters. Moreover, the other advantage is that overlapping clusters occur naturally in many applications such as the Yahoo! directory.

Our experiments reveal that the proposed algorithm has better accuracy quality than that of FIHC, *k*-means, Bisecting *k*-means, and UPGMA methods based on the comparison on four datasets. Our primary findings are as follows:

(1) Our approach is successful in avoiding the expansion of terms with noisy features on Classic and WebKB datasets.
(2) FIHC performs better for documents of short average length, but worse for documents of long average length.
(3) The other document clustering algorithms, like *k*-means, Bisecting *k*-means, and UPGMA, are sensitive when the number of clusters changes.

Our future work will focus on the following two aspects:

(1) Combining the syntactic analysis: for finding the important terms in a document, terms with different part-of-speech (POS) and syntactic attributes should be set at different weights according to their relatedness in a document. There are a lot of syntactic analysis tools that can be used to tag all terms in the document set, i.e., Qtag[13] parser. We will further study whether our proposed algorithm with a syntactic analysis tool can improve the clustering results.
(2) Incrementally updating the cluster tree: when the number of documents increases sequentially in a document set, it is inefficient to reform the cluster tree for each new insertion. That is, it is admirable to reflect the current state of the whole document set by incrementally updating the cluster tree [28–30]. Therefore, we intend to propose an efficient incremental clustering algorithm for assigning a new document to the most similar existing cluster in the future. Some recent researches on data mining concerning data streaming [31–33] may be applicable for such incremental clustering development.

Acknowledgements

This research was partially supported by National Science Council, Taiwan, ROC, under Contract No. NSC 98-2410-H-327-020-MY3 and No. NSC 98-2221-E-009-145.

References

[1] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.
[2] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, Proc. of the 6th ACM SIGKDD int'l conf. on Knowledge Discovery and Data Mining (KDD), 2000.
[3] P. Willett, Recent trends in hierarchic document clustering: a critical review, Information Processing & Management 24 (5) (1988) 577–597.
[4] C.D. Michenerand, R.R. Sokal, A quantitative approach to a problem in classification, Evolution 11 (1957) 130–162.
[5] A. Hotho, S. Staab, G. Stumme, Wordnet improves text document clustering, Proc. of SIGIR Int'l Conf on Semantic Web Workshop, 2003.
[6] B. Fung, K. Wang, M. Ester, Hierarchical document clustering using frequent itemsets, Proc. of SIAM Int'l Conf. on Data Mining (SDM'03), 2003, pp. 59–70, May.
[7] D.R. Recupero, A new unsupervised method for document clustering by using WordNet lexical and conceptual relations, Information Retrieval 10 (6) (2007) 563–579.
[8] F. Beil, M. Ester, X. Xu, Frequent term-based text clustering, Proc. of Int'l Conf. on knowledge Discovery and Data Mining (KDD'02), 2002, pp. 436–442.
[9] K. Lin, R. Kondadadi, A word-based soft clustering algorithm for documents, Computers and Their Applications (2001) 391–394.
[10] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, Proc. of ACM SIGMOD Int'l Conf. on Management of Data, 1993, pp. 207–216.
[11] H. Yu, D. Searsmith, X. Li, J. Han, Scalable construction of topic directory with nonparametric closed termset mining, ICDM'04, 2004, pp. 563–566.
[12] A. Hotho, A. Maedche, S. Staab, Ontology-based text document clustering, Kunstliche Intelligenz 16 (4) (2002) 48–54.

[13] http://www.english.bham.ac.uk/staff/omason/software/qtag.html.

[13] G.A. Miller, WordNet: a lexical database for English, Communications of the ACM 38 (11) (1995) 39–41.
[14] K. Dave, S. Lawrence, D.M. Pennock, Mining the peanut gallery: opinion extraction and semantic classification of product reviews, Proc. of the 12th Int'l Conf. on World Wide Web, 2003.
[15] L. Jing, L. Zhou, M.K. Ng, J.Z. Huang, Ontology-based distance measure for text clustering, Proc. Of SIAM Int'l Conf. on Data Mining, 2006.
[16] J. Sedding, D. Kazakov, WordNet-based text document clustering, Proc. of COLING-2004 Workshop on Robust Methods in Analysis of Natural Language Data, 2004.
[17] B. Liu, W. Hsu, Y. Ma, Pruning and summarizing the discovered associations, Proc. of the ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, 1999, pp. 125–134.
[18] T.P. Hong, K.Y. Lin, S.L. Wang, Fuzzy data mining for interesting generalized association rules, Fuzzy Sets and Systems 138 (2) (2003) 255–269.
[19] M.J. Martín-Bautista, D. Sánchez, J. Chamorro-Martínez, J.M. Serrano, M.A. Vila, Mining web documents to find additional query terms using fuzzy association rules, Fuzzy Sets and Systems 148 (1) (2004) 85–104.
[20] M. Kaya, R. Alhajj, Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rule mining, Applied Intelligence 24 (1) (2006) 7–15.
[21] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[22] C.L. Chen, F.S.C. Tseng, T. Liang, An integration of fuzzy association rules and WordNet for document clustering, Proc. of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-09), 2009, pp. 147–159.
[23] Y. Wang, J. Hodges, Document clustering with semantic analysis, Proc. of the 39th Annual Hawaii Int'l Conf. on System Sciences, 2006.
[24] H.J. Zimmermann, Fuzzy Set Theory and Its Application, 2nd Revised Edition, Kluver Academic Publisher, Boston, 1991.
[25] M. Craven, D. DiPasquo, A. McCallum, T. Mitchell, K. Nigam, S. Slattery, Learning to extract symbolic knowledge from the world wide web, Proc. of AAAI-98, 1998.
[26] N.O. Andrews, E.A. Fox, Recent developments in document clustering, Technical Report TR-07-35, Computer Science, Virginia Tech, 2007.
[27] S. Scott, S. Matwin, Text classification using WordNet hypernyms, Proc. Of Worksh. Usage of WordNet in NLP Systems at COLING-98, 1998, pp. 38–44.
[28] A. Pons-Porrata, R. Berlanga-Llavori, J. Ruiz-Shulcloper, Topic discovery based on text mining techniques, Information Processing and Management 43 (3) (2007) 752–768.
[29] R. Huang, W. Lam, An active learning framework for semi-supervised document clustering with language modeling, Data and Knowledge Engineering 68 (1) (2009) 49–67.
[30] S. Guha, A. Meyerson, N. Mishra, R. Motwani, L. O'Callaghan, Clustering data streams: theory and practice, IEEE Transactions on Knowledge and Data Engineering 15 (3) (2003) 515–528.
[31] S. Lühr, M. Lazarescu, Incremental clustering of dynamic data streams using connectivity based representative points, Data and Knowledge Engineering 68 (1) (2009) 1–27.
[32] H. Li, H. Chen, Mining non-derivable frequent itemsets over data stream, Data and Knowledge Engineering 68 (5) (2009) 481–498.
[33] N. Manerikar, T. Palpanas, Frequent items in streaming data: an experimental evaluation of the state-of-the-art, Data and Knowledge Engineering 68 (4) (2009) 415–430.

Chun-Ling Chen received her Ph.D degree, in computer science, from National Chiao Tung University, Taiwan, ROC. Currently, she is a postdoctoral research fellow of the Institute of Statistical Science, Academia SINICA, Taiwan, ROC. Her research interests include database, object-oriented conceptual modeling, information retrieval, text mining, and machine learning.

Frank S.C. Tseng received his B.S., M.S. and Ph.D. degrees, all in computer science and information engineering, from National Chiao Tung University, Taiwan, ROC, in 1986, 1988, and 1992, respectively. He joined the faculty of the Department of Information Management, Yuan-Ze University, Taiwan, ROC, on August 1995. From 1996 to 1997, he was the chairman of the department. Currently, he is a professor of the Department of Information Management, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, ROC. His research interests include database theory and applications, information retrieval, XML technologies for Internet computing, data/document warehousing, and data/text mining. Dr. Tseng is a member of the IEEE Computer Society and the Association for Computing Machinery.

Tyne Liang received her Ph.D. degree from National Chiao Tung University, Taiwan, ROC, majored in computer science. Currently, she is an associate professor of the Dept. of Computer Science, National Chiao Tung University, Taiwan, ROC. Her research interests include information retrieval, natural language processing, web mining, and inter-connection networking.