Knowl Inf Syst (2011) 28:687–708
DOI 10.1007/s10115-010-0364-2

REGULAR PAPER

An integration of fuzzy association rules and WordNet for document clustering

Chun-Ling Chen · Frank S. C. Tseng · Tyne Liang

Received: 27 August 2009 / Revised: 21 March 2010 / Accepted: 4 November 2010 /
Published online: 27 November 2010

Abstract With the rapid growth of text documents, document clustering technique is emerging for efficient document retrieval and better document browsing. Recently, some methods had been proposed to resolve the problems of high dimensionality, scalability, accuracy, and meaningful cluster labels by using frequent itemsets derived from association rule mining for clustering documents. In order to improve the quality of document clustering results, we propose an effective Fuzzy Frequent Itemset-based Document Clustering (F^2IDC) approach that combines fuzzy association rule mining with the background knowledge embedded in WordNet. A term hierarchy generated from WordNet is applied to discover generalized frequent itemsets as candidate cluster labels for grouping documents. We have conducted experiments to evaluate our approach on Classic4, Re0, R8, and WebKB datasets. Our experimental results show that our proposed approach indeed provide more accurate clustering results than prior influential clustering methods presented in recent literature.

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

C.-L. Chen · T. Liang
Department of Computer Science, National Chiao Tung University,
HsinChu 300, Taiwan, ROC
e-mail: chunling@cs.nctu.edu.tw

T. Liang
e-mail: tliang@cs.nctu.edu.tw

Present Address:
F. S. C. Tseng (✉)
Department of Information Management, National Kaohsiung 1st University of Science and Technology,
1, University Road, YenChao, Kaoshiung County 824, Taiwan, ROC
e-mail: imfrank@ccms.nkfust.edu.tw

1 Introduction

With the rapid growth of text documents, document clustering has become one of the main techniques for managing large document collections [6]. Several effective document clustering algorithms have been proposed including the k-means [16], Bisecting k-means [26], Hierarchical Agglomerative Clustering (HAC) [29], Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [19], etc. However, there still exist some challenges for the clustering quality [2,8,10], such as (1) have high-dimensional term features, (2) are not scalable for large document sets (like UPGMA), (3) require the user to specify the number of clusters as an input parameter, which is usually unknown in advance (like k-means), (4) do not provide a meaningful label (or description) for a cluster, and (5) do not embody any external knowledge to extract semantics from texts.

In reply to these challenges (1)–(4), a new research area, namely “frequent itemset-based clustering”, has been proposed. In [2], Beil et al. 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, e.g., Apriori [1]. They consider the only low-dimensional frequent itemsets as clusters. However, the experiments evaluated by Fung et al. [8] showed that HFTC is not scalable. For a scalable algorithm, Fung et al. proposed a novel approach, namely Frequent Itemset-based Hierarchical Clustering (FIHC), by using derived frequent itemsets to construct a hierarchical topic tree for clusters. They also proved that using frequent itemsets for document clustering can reduce the dimension of a vector space effectively. But, our experimental results showed that FIHC is not scalable for long average length of documents. Yu et al. [31] 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 dimensionality. TDC uses a complicated tree structure to build the hierarchy by linking each itemset of size k with all of its subsets at level $k - 1$. This approach may result in high accuracy, but would affect the overall clustering quality because of too much node duplication when terms in the document set are highly correlated. Moreover, HFTC, FIHC, and TDC only account for term frequency in the documents and all ignore the important semantic relationships between terms. Therefore, our approach aims to investigate whether WordNet semantic relationships can improve the clustering quality of frequent itemset-based clustering.

Recently, WordNet [20], which is one of the most widely used thesauruses for English, has been used to group documents with its semantic relations of terms [10,13]. Many existing document clustering algorithms mainly transform text documents into simplistic flat bags of document representation, e.g., term vectors or bag-of-words. Once terms are treated as individual items in such simplistic representation, their semantic relations will be lost. Thus, Dave et al. [13] employed synsets as features for document representation and subsequent clustering. However, synsets would decrease the clustering performance in all experiments without considering word sense disambiguation. Accordingly, Hotho et al. [10] used WordNet in document clustering for word sense disambiguation to improve the clustering performance. In order to consider the conceptual similarity of terms that do not co-occur actually, we employ WordNet in our document clustering approach and show where and how it can be fruitfully utilized.

As a result, a term that frequently occurs in a document does not imply its importance [25]. Since some important terms that express the topics of a document may be rarely appeared in the document collection. If we use association rule mining in our approach, then only the terms which frequently occur in the document collection can be obtained, which implies the important sparse terms be obscured in the process of document clustering. Moreover, association rule mining often suffers from producing too many itemsets, especially when items in the dataset are highly correlated [15]. Considering these two issues, we propose an approach which stems from prior studies [9,12,18], by integrating fuzzy set concepts [32] and association rule mining to provide significant dimensionality reduction over interesting frequent itemsets. To illustrate the usefulness of fuzzy data mining in document clustering, we use fuzzy set concept to model the term frequency describing the important degree of a term in a document. In contrast with using the crisp set concept, in which a term is either a member of a document or not, fuzzy set concept makes it possible that a term belongs to a document to a certain degree. 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*.

The frequent itemsets found in the document collections often reveal hidden relationships and correlations among terms. In this paper, we extend our previous study [3,4] and further propose an effective Fuzzy Frequent Itemset-based Document Clustering (F^{2}IDC) approach based on fuzzy association rule mining in conjunction with WordNet for clustering textual documents. In contrast to our previous study, this paper illustrates how to add these hypernyms as term features for the document representation, how to utilize the hypernyms of WordNet in the process of fuzzy association rule mining to obtain the conceptual labels from the derived clusters, and how we conducted experiments to evaluate more datasets.

In summary, our approach has the following advantages:

1. It presents a means of dynamically deriving a hierarchical organization of concepts from the WordNet thesaurus based on the content of each document without use of training data or standard clustering techniques;
2. Following prior studies [9,12,18], it extends a fuzzy data representation to text mining, especially with the use of is-a hierarchy of WordNet, for discovering generalized fuzzy frequent itemsets and providing conceptual labels for clusters;
3. By conducting experimental evaluations on the four datasets of Classic4, Re0, R8, and WebKB, the result presents better accuracy quality than that of FIHC, Bisecting k-means, and UPGMA methods.

The subsequent sections of this paper are organized as follows. In Sect. 2, we briefly review related work on the general process of document clustering. In Sect. 3, a detailed description of our approach with an example is presented. The experimental evaluation is described and the results are shown in Sect. 4. Finally, we conclude in Sect. 5.

2 A generic process of document clustering

The aim of document clustering is to group similar documents together based on the content of a set of documents. According to [28], we divide the general process of document clustering into three main stages, including *Document Pre-processing*, *Document Representation*, and *Document Clustering* (as shown in Fig. 1). These stages are described as follows:

1. *Document Pre-processing*. There are two steps in this stage, namely *Term Extraction* and *Term Selection*, for generating the term set from the document collection.

   (1) *Term Extraction*: The whole extraction process is as follows:
   - *Extract terms.* Divide the sentences into terms and extract terms as features.

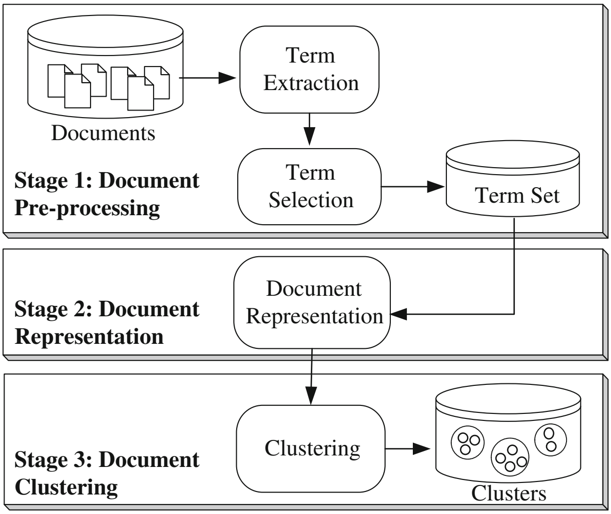

Documents
Term Extraction
Term Selection
Term Set
Stage 1: Document Pre-processing
Stage 2: Document Representation
Document Representation
Stage 3: Document Clustering
Clustering
Clusters

Fig. 1 General process of document clustering

- *Remove the stop words.* A pre-defined stop-word list[1] is applied to remove commonly used words that do not discriminate for topics.
- *Conduct word stemming.* Use the developed stemming algorithms, such as Porter [21], to convert a word to its stem or root form.

(2) *Term Selection*: After extracting terms, it is crucial to reduce the set of term features, a process referred to as term selection. Several methods, such as itemset pruning [2], feature clustering or co-clustering [17], feature selection technique [24], and matrix factorization [30], have been applied to reduce the dimensionality for high clustering accuracy.

2. *Document Representation*. Several document representation methods have been proposed, including binary (which shows the presence or absence of a term in a document) and term frequency (which shows the frequency of a term in a document).
3. *Document Clustering*. Common approaches for document clustering have been used, including the k-means [16], Bisecting k-means [26], Hierarchical Agglomerative Clustering (HAC) [29], Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [19], etc.

3 Fuzzy frequent itemset-based document clustering (F^{2}IDC)

In this section, we will illustrate the overall process and detail design of the proposed F^{2}IDC approach. As shown in Fig. 2, the process of F^{2}IDC is similar to the general process of document clustering (as depicted in Fig. 1), except for the gray-colored components (i.e., Document Enrichment, Fuzzy Frequent Itemset Mining, and Clustering, etc.) In the following, we explain these three stages in our framework:

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

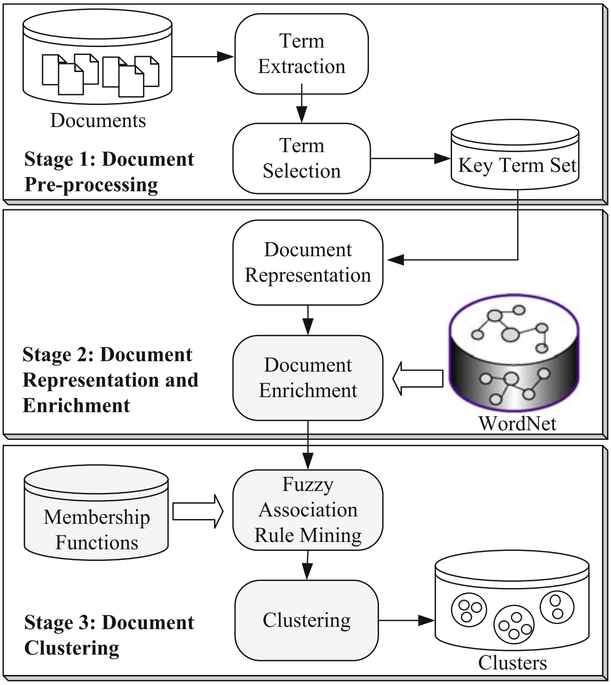


Fig. 2 The F^{2}IDC framework

1. *Document Pre-processing*. A set of terms from the document collection are first extracted. Then, we use a feature selection method to find the terms that are significant and important to represent the content of each document.
2. *Document Representation and Enrichment.* After the steps of document representation and enrichment, the designated document representation is prepared for the later mining algorithm.
3. *Document Clustering*. Starting from the designated document representation of all documents, we run fuzzy association rule mining algorithm to discover fuzzy frequent itemsets and then generate the candidate clusters. Furthermore, in order to represent the degree of importance of a document in a candidate cluster, an $n \times k$ Document-Cluster Matrix (DCM) will be constructed to calculate the similarity of terms in a document and a candidate cluster. Based on the obtained DCM, each document will be assigned into a target cluster.

3.1 Stage 1: Document pre-processing

As with document clustering techniques, the proposed approach starts with term extraction. For a document set $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$, a term set $T_D = \{t_1, t_2, \ldots, t_j, \ldots, t_s\}$, which is the set of terms appeared in D, can be obtained. The details of the term extraction are described in Sect. 2.

The feature description of a document is constituted by terms of the document set to form a term vector. A term vector with high dimensions is easy to make clustering inefficient and difficult in principle. Hence, in this paper, we employ tf-idf as the feature selection method to produce a low-dimensional term vector. A term will be discarded if its weight is less than a tf-idf threshold γ. Formula (1) is used for the measurement of $tfidf_{ij}$ for the importance of a term t_j within a document d_i. For preventing a bias for longer documents, the weighted frequency of each term is usually normalized by the maximum frequency of all terms in d_i and is defined as follows:

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

where f_{ij} is the frequency of t_j in d_i, and the denominator is the maximum frequency of all terms in d_i. $|D|$ is the total number of documents in the document set D, and $|\{d_i|t_j \in d_i, d_i \in D\}|$ is the number of documents containing t_j.

After the step of term selection, the *key term set* of D, denoted $K_D = \{t_1, t_2, \ldots, t_j, \ldots, t_p\}$ is obtained. K_D is a subset of T_D, including only meaningful key terms, and satisfying the pre-defined minimum tf-idf threshold γ.

3.2 Stage 2: Document representation and enrichment

In this stage, each document d_i in D is represented using those terms in K_D. Thus, each document $d_i \in D$, denoted $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \ldots, (t_j, f_{ij}), \ldots, (t_p, f_{ip})\}$, is represented by a set of pairs (term, frequency), where the frequency f_{ij} represents the occurrence of the key term t_j in d_i.

Accordingly, we enrich the document representation by using WordNet, a source repository of semantic meanings. WordNet, developed by Miller et al. [20], consists of so-called synsets, together with a hypernym/hyponym hierarchy.

The basic idea of document enrichment is to add the generality of terms by corresponding hypernyms of WordNet based on the key terms appeared in each document. Each key term is linked up to the top 5 levels of hypernyms. For a simple and effective combination, these added hypernyms form a new key term set, denoted $K_D = \{t_1, t_2, \ldots, t_p, h_1, \ldots, h_d\}$, where h_j is a hypernym. The enriched document d_i is represented by $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \ldots, (t_p, f_{ip}), (h_1, hf_{i1}), \ldots, (h_d, hf_{id})\}$, where a weight of 0 will be assigned to several terms appearing in some of the documents but not in d_i. The frequency f_{ij} of a key term t_j in d_i is mapped to its hypernyms $\{h_1, \ldots, h_j, \ldots, h_d\}$ to accumulate as the frequency hf_{ij} of h_j.

The reason of using hypernyms of WordNet is to reveal hidden similarities to identify related topics, which potentially leads to better clustering quality [23]. For example, a document about ‘sale’ may not be associated to a document about ‘trade’ by the clustering algorithm if there are only ‘sale’ and ‘trade’ in the key term set. But, if the more general term ‘commerce’ is added to both documents, their semantic relation is revealed. The suitable representation of each document for the later mining can be derived by Algorithm 1.

3.3 Stage 3: Document clustering

The final stage is to group the documents into clusters. In the following, we first define the membership functions and present our fuzzy association rule mining algorithm for texts. Subsequently, based on the mining results, we illustrate the details of the clustering process.

3.3.1 The membership functions

The membership functions are used to convert each term frequency into a fuzzy set. A *$t-f$ fuzzy set* of document d_i is a pair (F_{ij}, w^r_{ij}), where F_{ij} is a set and equals to

Algorithm 1 Obtain the designated representation of all documents

```
Input: A document set $D$; A well-defined stop word list; WordNet; The minimum tf-idf
        threshold $\gamma$.
Output: The formal representation of all documents in $D$.
1. Extract the term set $T_D = \{t_1, t_2, \ldots, t_j, \ldots, t_s\}$
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 (1)
         (2) Retain the term if $tfidf_{ij} \geq \gamma$
5. Form the key term set $K_D = \{t_1, t_2, \ldots, t_j, \ldots, t_p\}$
6. 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 WordNet
              (a) $hf_{ij} \rightarrow hf_{ij} + f_{ij}$
              (b) $K_D \rightarrow K_D \cup \{h_j\}$
7. 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 $tfidf_{ij} \geq \gamma$
8. Form the new key term set $K_D = \{t_1, t_2, \ldots, t_p, h_1, \ldots, h_d\}$   // $m(= p + d)$ is total
   number of key terms
9. 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 $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \ldots, (t_p, f_{ip}), (h_1, hf_{i1}), \ldots,$
   $(h_d, hf_{id})\}$
```

$\{w_{ij}^{Low}(f_{ij})/t_j.Low, w_{ij}^{Mid}(f_{ij})/t_j.Mid, w_{ij}^{High}(f_{ij})/t_j.High\}$, $w_{ij}^{r} : F \rightarrow [0, 2]$, and r can be *Low*, *Mid*, or *High*. The notation $t_j.r$ is called a fuzzy region of t_j. For each term pair (t_j, f_{ij}) of document d_i, $w_{ij}^{r}(f_{ij})$ is the grade of membership of t_j in d_i with *Low*, *Mid*, and *High* membership functions defined by Formulas (2), (3), and (4), respectively. The derived membership functions are shown in Fig. 3.

In Formulas (2), (3), and (4), $\min(f_{ij})$ is the minimum frequency of terms in D, $\max(f_{ij})$ is the maximum frequency of terms in D, and $\text{avg}(f_{ij}) = \frac{\lceil \sum_{i=1}^{n} f_{ij} \rceil}{|K|}$, 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* $C_D = \{c_1^1, c_2^1, \ldots, c_i^q, \ldots, c_f^q\}$ for a document set D, a *candidate cluster set* $\tilde{C}_D = \{\tilde{c}_1^1, \ldots, \tilde{c}_{l-1}^2, \tilde{c}_l^q, \ldots, \tilde{c}_k^q\}$, 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

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

$$w_{ij}^{mid}(f_{ij})=\begin{cases}0, & f_{ij}=0\\ 1, & f_{ij}<a\\ 1+{f_{ij}-a}\big/{b-a}, & a\le f_{ij}\le b\\ 2, & b<f_{ij}<c\\ 1+{f_{ij}-d}\big/{c-d}, & c\le f_{ij}\le d\\ 1, & f_{ij}>d\end{cases},\quad \begin{aligned}&a=min(f_{ij}),\\&b=\frac{\lceil avg(f_{ij})+min(f_{ij})\rceil}{2},\\&c=avg(f_{ij}),\\&d=avg(f_{ij})+\frac{\lceil max(f_{ij})-avg(f_{ij})\rceil}{4}\end{aligned}\tag{3}$$

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

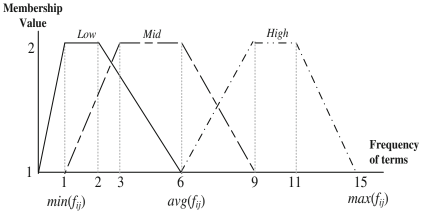


Fig. 3 The predefined membership functions

$\tau=\{t_1,t_2,\ldots,t_q\}\subseteq K_D$, $q\ge 1$, 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 $\tilde{c}^1_{(\text{trade})}=(\{d_2,d_3\},\{\text{trade}\})$ means the term "trade" appeared in documents d_2 and d_3.

In the mining process, it is considered that documents and key terms are transactions and purchased items, respectively. Algorithm 2 generates fuzzy frequent itemsets based on pre-defined 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. Moreover, 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, the strength of association among key terms in the document set will be estimated by using confidence values. Our algorithm computes two confidence values of a rule pair to check the strength of association among the key terms $(t_1,t_2,\ldots,t_q)$ of the fuzzy frequent q-itemsets. Consider the candidate cluster $\tilde{c}^2_{(\text{sale, trade})}$ 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, $\tilde{c}^2_{(\text{sale, trade})}$ is put in the candidate cluster set $\tilde{C}_D$. Finally, the candidate cluster set $\tilde{C}_D$ will be output. In this study, we set $\lambda=0.7$.

3.3.3 Clustering

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 $\tilde{c}^q_l$. The DCM is derived from the Document-Term Matrix (DTM) and the Term-Cluster (TCM). A formal illustration of DCM can be found in Fig. 4.

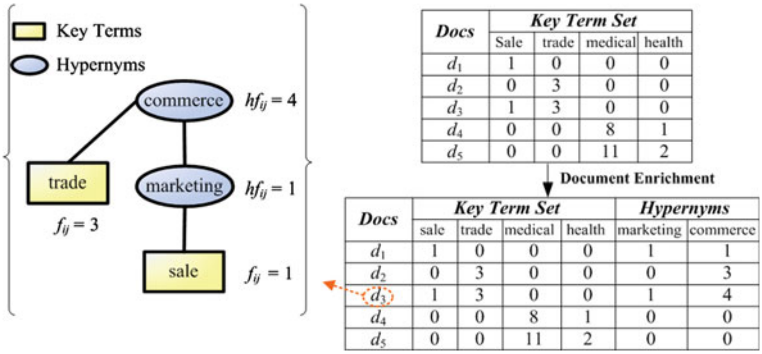
Based on DCM, c^q_i may or may not be assigned a subset of documents. For the documents in each c^q_i, the intra-cluster similarity is minimized and the inter-clusters similarity is maximized.

Algorithm 2 Obtain the fuzzy frequent itemsets as candidate clusters

```
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 Sect. 3.3.1); The minimum support value $\theta$; The minimum confidence value $\lambda$.
Output: A set of candidate cluster $\tilde{C}_D$.
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($L_{q-1}, \theta$)   // similar to the a priori algorithm [1]
      (2) For each candidate $q$-itemsets $\tau$ with key terms $(t_1, t_2, \ldots, t_q) \in C_q$ do
            (a) For each $d_i \in D$ do
            $w_{i\tau} = min\left\{w_{ij}^{max\text{-}R_j} | j = 1, 2, \ldots, q\right\}$   //$w_{ij}^{max\text{-}R_j}$ 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 $\tau$ 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
            $\tau_1 \wedge \cdots \wedge \tau_{k-1} \wedge \tau_{k+1} \wedge \cdots \wedge \tau_q \rightarrow \tau_k, k = 1$ to $q$
      (2) Calculate the confidence values of all possible association rules
            $confidence(\tau) = \frac{\sum_{i=1}^{n} w_{i\tau}}{\sum_{i=1}^{n} (w_{i1} \wedge \cdots \wedge w_{ik-1}, w_{ik+1} \wedge \cdots \wedge w_{iq})}$
      (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($L_{q-1}, \theta$)
1. For each itemset $l_1 \in L_{q-1}$ do
      For each itemset $l_2 \in L_{q-1}$ do
         (1) if ($l_1[1] = l_2[1] \wedge l_1[2] = l_2[2] \wedge \cdots \wedge l_1[k-2] = l_2[k-2] \wedge l_1[k-1] = l_2[k-1]$) then $C_q = \{c | c = l_1 \times l_2\}$
2. Return $C_q$
```

$$\underset{\textit{Document - Cluster Marix}}{\begin{array}{c|ccccc|} & \tilde{c}_1^1 & \cdots & \tilde{c}_{l-1}^2 & \tilde{c}_l^q & \cdots\ \tilde{c}_k^q \\ d_1 & v_{11}\cdots & & v_{1l-1} & v_{1l} & \cdots\ v_{1k} \\ d_2 & v_{21}\cdots & & v_{2l-1} & v_{2l} & \cdots\ v_{2k} \\ \vdots & \vdots\ \ddots & & \vdots & \vdots & \ddots\ \vdots \\ d_n & v_{n1}\cdots & & v_{nl-1} & v_{nl} & \cdots\ v_{nk} \end{array}}_{n\times k} = \underset{\textit{Document - Term Marix}}{\begin{array}{c|cccc|} & t_1 & t_2 & \cdots & t_m \\ d_1 & & & & \\ d_2 & w_{21}^{\max-R_j} & w_{22}^{\max-R_j} & \cdots & w_{2m}^{\max-R_j} \\ \vdots & & & & \\ d_n & & & & \end{array}}_{n\times m} \cdot \underset{\textit{Term - Cluster Marix}}{\begin{array}{c|cccc|} & \tilde{c}_1^1 & \tilde{c}_2^1 & \cdots & \tilde{c}_k^q \\ t_1 & & g_{12}^{\max-R_j} & & \\ t_2 & & g_{22}^{\max-R_j} & & \\ \vdots & & \vdots & & \\ t_m & & g_{m2}^{\max-R_j} & & \end{array}}_{m\times k}$$

Fig. 4 Document-Cluster Matrix



Docs	Key Term Set			
	Sale	trade	medical	health
d_1	1	0	0	0
d_2	0	3	0	0
d_3	1	3	0	0
d_4	0	0	8	1
d_5	0	0	11	2

Docs	Key Term Set				Hypernyms	
	sale	trade	medical	health	marketing	commerce
d_1	1	0	0	0	1	1
d_2	0	3	0	0	0	3
d_3	1	3	0	0	1	4
d_4	0	0	8	1	0	0
d_5	0	0	11	2	0	0

Fig. 5 The process of Algorithm 1 of this example

The objective of Algorithm 3 is to assign each document to the best fitting cluster c_i^q and finally obtain the target cluster set for output. For improving 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 clusters with the similar topic. The inter-cluster similarity measurement is defined as follows:

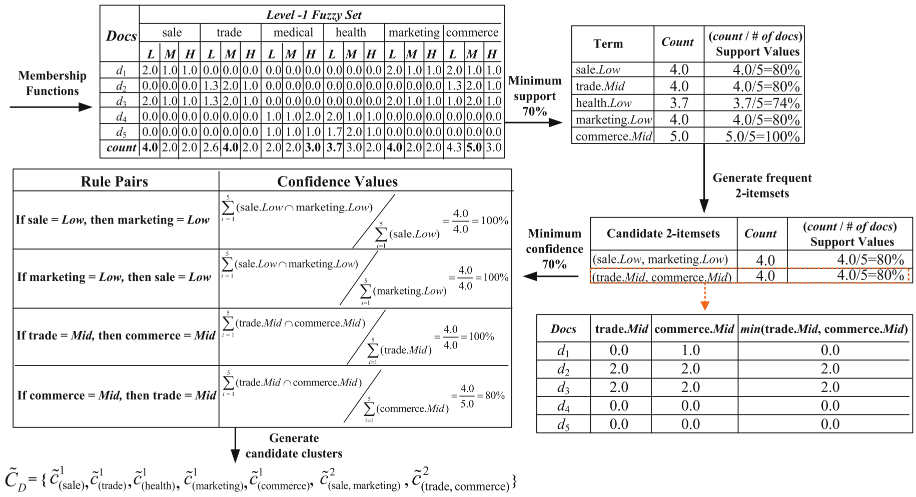
$$Inter\text{-}Sim(c_x^q, c_y^q) = \frac{\sum_{i=1,d_i\in c_x^q,c_y^q}^{n} v_{ix} \times v_{iy}}{\sqrt{\sum_{i=1,d_i\in c_x^q}^{n} (v_{ix})^2 \times \sum_{i=1,d_i\in c_y^q}^{n} (v_{iy})^2}} \tag{5}$$

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. The target cluster pair with the highest *Inter-Sim* value keeps merging until the *Inter-Sim* values of all target clusters are less than the minimum *Inter-Sim* threshold δ. In this study, we set $\delta = 0.5$.

3.4 An illustrative example of F^{2}IDC method

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}. Figure 5 illustrates the process of Algorithm 1 to obtain the representation of all documents.

Consider the representation of all documents generated by Algorithm 1 in Fig. 5, the membership functions defined in Fig. 3, the minimum support value 70%, and the minimum confidence value 70% as inputs. The fuzzy frequent itemsets discovery procedure is depicted in Fig. 6.



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	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	1.0	1.0	2.0	1.0	1.0
d_2	0.0	0.0	0.0	1.3	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.3	2.0	1.0
d_3	2.0	1.0	1.0	1.3	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	1.0	1.0	1.0	2.0	1.0
d_4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	2.0	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	0.0	0.0	0.0	1.0	1.0	1.0	1.7	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
count	**4.0**	2.0	2.0	2.6	**4.0**	2.0	2.0	2.0	**3.0**	**3.7**	3.0	2.0	**4.0**	2.0	2.0	4.3	**5.0**	3.0

Term	*Count*	(*count* / # *of docs*) Support Values
sale.*Low*	4.0	4.0/5=80%
trade.*Mid*	4.0	4.0/5=80%
health.*Low*	3.7	3.7/5=74%
marketing.*Low*	4.0	4.0/5=80%
commerce.*Mid*	5.0	5.0/5=100%

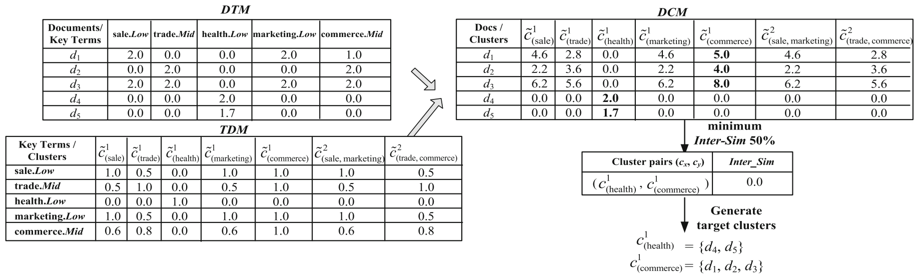
Candidate 2-itemsets	*Count*	(*count* / # *of docs*) Support Values
(sale.*Low*, marketing.*Low*)	4.0	4.0/5=80%
(trade.*Mid*, commerce.*Mid*)	4.0	4.0/5=80%

Docs	trade.*Mid*	commerce.*Mid*	*min*(trade.*Mid*, commerce.*Mid*)
d_1	0.0	1.0	0.0
d_2	2.0	2.0	2.0
d_3	2.0	2.0	2.0
d_4	0.0	0.0	0.0
d_5	0.0	0.0	0.0

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*	$\sum_{i=1}^{5}(\text{trade}.Mid \cap \text{commerce}.Mid) \Big/ \sum_{i=1}^{5}(\text{trade}.Mid) = \frac{4.0}{4.0} = 100\%$
If commerce = *Mid*, then trade = *Mid*	$\sum_{i=1}^{5}(\text{trade}.Mid \cap \text{commerce}.Mid) \Big/ \sum_{i=1}^{5}(\text{commerce}.Mid) = \frac{4.0}{5.0} = 80\%$

$$\tilde{C}_D = \{ \tilde{c}^1_{(\text{sale})}, \tilde{c}^1_{(\text{trade})}, \tilde{c}^1_{(\text{health})}, \tilde{c}^1_{(\text{marketing})}, \tilde{c}^1_{(\text{commerce})}, \tilde{c}^2_{(\text{sale, marketing})}, \tilde{c}^2_{(\text{trade, commerce})} \}$$

Fig. 6 The process of Algorithm 2 of this example



DTM

Documents/ Key Terms	sale.*Low*	trade.*Mid*	health.*Low*	marketing.*Low*	commerce.*Mid*
d_1	2.0	0.0	0.0	2.0	1.0
d_2	0.0	2.0	0.0	0.0	2.0
d_3	2.0	2.0	0.0	2.0	2.0
d_4	0.0	0.0	2.0	0.0	0.0
d_5	0.0	0.0	1.7	0.0	0.0

TDM

Key Terms / Clusters	$\tilde{c}^1_{(\text{sale})}$	$\tilde{c}^1_{(\text{trade})}$	$\tilde{c}^1_{(\text{health})}$	$\tilde{c}^1_{(\text{marketing})}$	$\tilde{c}^1_{(\text{commerce})}$	$\tilde{c}^2_{(\text{sale, marketing})}$	$\tilde{c}^2_{(\text{trade, commerce})}$
sale.*Low*	1.0	0.5	0.0	1.0	1.0	1.0	0.5
trade.*Mid*	0.5	1.0	0.0	0.5	1.0	0.5	1.0
health.*Low*	0.0	0.0	1.0	0.0	0.0	0.0	0.0
marketing.*Low*	1.0	0.5	0.0	1.0	1.0	1.0	0.5
commerce.*Mid*	0.6	0.8	0.0	0.6	1.0	0.6	0.8

DCM

Docs / Clusters	$\tilde{c}^1_{(\text{sale})}$	$\tilde{c}^1_{(\text{trade})}$	$\tilde{c}^1_{(\text{health})}$	$\tilde{c}^1_{(\text{marketing})}$	$\tilde{c}^1_{(\text{commerce})}$	$\tilde{c}^2_{(\text{sale, marketing})}$	$\tilde{c}^2_{(\text{trade, commerce})}$
d_1	4.6	2.8	0.0	4.6	**5.0**	4.6	2.8
d_2	2.2	3.6	0.0	2.2	**4.0**	2.2	3.6
d_3	6.2	5.6	0.0	6.2	**8.0**	6.2	5.6
d_4	0.0	0.0	**2.0**	0.0	0.0	0.0	0.0
d_5	0.0	0.0	**1.7**	0.0	0.0	0.0	0.0

Cluster pairs (c_x, c_y)	*Inter_Sim*
$(c^1_{(\text{health})}, c^1_{(\text{commerce})})$	0.0

$$c^1_{(\text{health})} = \{d_4, d_5\}$$
$$c^1_{(\text{commerce})} = \{d_1, d_2, d_3\}$$

Fig. 7 The process of Algorithm 3 of this example

Moreover, consider the candidate cluster set $\tilde{c}_D$ was already generated in Fig. 6. Now, suppose the minimum *Inter-Sim* value is 0.5. Figure 7 illustrates the process of Algorithm 3 and shows the final results.

4 Experiments

In this section, we experimentally evaluated the performance of the proposed algorithm by comparing with that of FIHC, 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. [26] compared the performance of some influential clustering algorithms, and the results indicated that UPGMA and Bisecting k-means are the most accurate clustering algorithms. Therefore, the CLUTO Clustering tool[3] is applied to generate the results of Bisecting k-means and UPGMA. 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

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

[3] http://glaros.dtc.umn.edu/gkhome/views/cluto/.

Algorithm 3 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 $\delta$;
Output: The target cluster set $C_D = \{c_1^1, c_2^1, \ldots, c_i^q, \ldots, c_f^q\}$
1. Build $n \times m$ document-term matrix $W = \left[ w_{ij}^{max\text{-}R_j} \right]$ //$w_{ij}^{max\text{-}R_j}$ is the weight (fuzzy value) of $t_j$ in $d_i$ and $t_j \in L_1$
2. Build $m \times k$ term-cluster matrix $G = \left[ g_{jl}^{max\text{-}R_j} \right]$ //$g_{jl}^{max\text{-}R_j} = \frac{score(\tilde{c}_l^q)}{\sum_{i=1}^{n} w_{ij}^{max\text{-}R_j}}$, $1 \le j \le m$, $1 \le l \le k$, and $score(\tilde{c}_l^q) = \sum_{d_i \in \tilde{c}_l^q, t_j \in \tau} w_{ij}^{max\text{-}R_j}$, where $w_{ij}^{max\text{-}R_j}$ is the weight (fuzzy value) of $t_j$ in $d_i \in \tilde{c}_l^q$ and $t_j \in L_1$.
3. Build $n \times k$ document-cluster matrix $V = W \cdot G = \left[ v_{il} \right] = \sum_{m=1}^{m} w_{im} g_{ml}$
4. Based on $V$, assign $d_i$ to a target cluster $c_l^q$
  (1) $c_l^q = \{d_i | v_{il} = max\{v_{i1}, v_{i2}, \ldots, v_{il}\} \in \tilde{c}_l^q$, where the number of $v_{il}$ is 1},
      otherwise (2)
  (2) $c_l^q = \{d_i | v_{il} = max\{v_{i1}, v_{i2}, \ldots, v_{il}\} \in \tilde{c}_l^q$, where the number of $v_{il} > 1$ and $\tilde{c}_l^q$
      with the highest fuzzy count value corresponding to its fuzzy frequent itemset}
5. 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 $I$
6. Output $C_D$
```

with 1 GB memory. The implementation 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: Classic4, Re0, R8, and WebKB, which are widely adopted as standard benchmarks for the text categorization task. They are heterogeneous in terms of document size, cluster size, number of classes, and document distribution. Moreover, these datasets are not specially designed to combine with WordNet for facilitating the clustering result.

Table 1 summarizes the statistics of these datasets. Each document is pre-classified into a single topic, i.e., a natural class. The class information is utilized in the evaluation method

Table 1 Statistics for our test datasets

Datasets	Documents total	Classes total	Class size			Document length
			Max	Average	Min	Average
Class4	7,094	4	3,203	1,774	1,033	39
Re0	1,504	13	608	116	11	69
R8	7,674	8	3,923	959	51	48
WebKB	4,199	4	1,641	1,050	504	124

for measuring the accuracy of the clustering result. The detailed information of these datasets is described as follows:

1. *Classic4*[4]: This document set is a combination of the four classes CACM, CISI, CRAN, and MED abstracts. Classic4 includes 3,204 CACM documents, 1,460 CISI documents from information retrieval papers, 1,398 CRANFIELD documents from aeronautical system papers, and 1,033 MEDLINE documents from medical journals.
2. *Re0*[5]: Re0 is a text document dataset, derived from Reuters-21578[6] text categorization test collection Distribution 1.0. Re0 includes 1,504 documents belonging to 13 different classes.
3. *R8*[7]: R8 is a subset of the Reuters-21578 text categorization collections. It considers only the documents associated with a single topic and the classes which still have at least one train and one test example. R8 includes 7,674 documents with 8 most frequent classes.
4. *WebKB*[8]: This dataset consists of web pages collected by the WebKB project of the CMU text learning group [5]. These pages are manually classified into seven categories. In our test, we select the four most popular entity-representing categories: course, faculty, project, and student.

4.2 Evaluation of cluster quality

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, a standard evaluation measures, namely Overall F-measure [8], is widely used to evaluate the generated clustering results. More important, this measure balances the cluster precision and cluster recall.

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.

[4] ftp://ftp.cs.cornell.edu/pub/smart/.

[5] The pre-processed datasets can be downloaded at http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download/.

[6] http://www.daviddlewis.com/resources/testcollections/.

[7] The pre-processed datasets can be downloaded at http://web.ist.utl.pt/~acardoso/datasets/.

[8] The pre-processed datasets can be downloaded at http://www.cs.technion.ac.il/~ronb/thesis.html.

Table 2 List of all parameters for our algorithms and the other three algorithms

Parameter name	F^{2}IDC	FIHC	UPGMA[a,b]	Bi. k-means[c]
Datasets			Classic4, Re0, R8, WebKB	
Stopword removal			Yes	
Stemming			Yes	
Length of the smallest term			three	
Weight of the term vector	tf	tf-idf	tf-idf	tf-idf
Levels of hypernyms			$H1$, $H2$, $H3$, $H4$, $H5$	
Cluster count k			3, 15, 30, 60	

$H1$ represents the addition of direct hypernyms; $H2$ stands for the addition of hypernyms of the first and second levels, and so on

[a] The command was vcluster -clmethod=agglo -crfun=upgma -sim=cos -rowmodel=maxtf -colmodel=idf -clabelfile=<X>.mat.clabel <X>.mat <K>

[b] <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

[c] The command was vcluster -clmethod=rbr -crfun=i2 -sim=cos –cstype=best -rowmodel=maxtf -colmodel=idf -clabelfile=<X>.mat.clabel <X>.mat <K>

Overall F-measure The F-measure is often employed to evaluate the accuracy of clustering results. Fung et al. [8] 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.

$$F(C) = \sum_{l_j \in L} \frac{|l_j|}{|D|} \max_{c_i \in C}\{F\}, \quad \text{where } F = \frac{2PR}{P+R}, \ P = \frac{|c_i \cap l_j|}{|c_i|} \quad \text{and} \quad R = \frac{|c_i \cap l_j|}{|l_j|} \tag{6}$$

In general, the higher the $F(C)$ values, the better the clustering solution is.

Improvement ratio The *Improvement ratio* (*IR*) is the ratio of improvements to the $F(C)$ value of our proposed approach, F^{2}IDC, when compared with the other compared algorithms. In the following, we define the *IR*:

$$IR = \frac{F(C)^{\mathrm{F^2IDC}} - F(C)^{<Y>}}{F(C)^{<Y>}}, \tag{7}$$

where $F(C)^{\mathrm{F^2IDC}}$ and $F(C)^{<Y>}$ represent the $F(C)$ values of F^{2}IDC and the other three algorithms (e.g., $<Y>$ can be FIHC, UPGMA, or Bi. k-means), respectively. A higher IR value indicates that the clustering quality of F^{2}IDC method is better than the clustering quality of the other algorithms.

4.3 Parameters selection

Table 2 summarizes the parameters for our proposed method and the other algorithms to compare the clustering performance.

Before applying F^{2}IDC, we first consider the feature selection strategy. In order to select the most representative features, we use Formula (1) to obtain the key terms with weights higher than the pre-defined thresholds γ. Table 3 shows the keyword statistics of our test datasets and the suggested threshold for each dataset. Documents were then represented as

Table 3 Keyword statistics of our test datasets

Dataset	# of terms	# of terms after pre-processing	# of terms after enriching	γ threshold	
				F^{2}IDC	WordNet_based F^{2}IDC
Classic4	40, 291	40,279	41,931	0.60	0.65
Re0	2, 886	2,678	3,507	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

tf (term frequency) vectors, and unimportant terms were discarded. This process implies a significant dimensionality reduction without loss of clustering performance.

The two algorithms, F^{2}IDC and FIHC, both have two main parameters for the adjustment of accuracy quality. The first parameter, denoted MinSup, is mandatory, which means the minimum support for frequent itemsets generation. The other parameter, denoted KCluster, is optional, which represents the number of clusters. As Bisecting k-means and UPGMA require a predefined number of clusters as their inputs, their KCluster parameters must be provided.

4.4 Experimental results and analysis

The experiments were conducted by the following steps. First, we evaluated our method, F^{2}IDC, on the four datasets mentioned earlier and compared its accuracy with that of FIHC, Bisecting k-means, and UPGMA. Moreover, we verified if the use of WordNet can generate conceptual labels for derived clusters. Second, the dataset, RVC1 (Reuters Corpus Volume 1) [14], was chosen to evaluate the efficiency and scalability of F^{2}IDC.

4.4.1 Accuracy comparison for F^{2}IDC algorithm

Table 4 presents the obtained overall F-Measure values for WordNet-based F^{2}IDC and the other WordNet-based algorithms by comparing four different numbers of clusters, namely 3, 15, 30, and 60, on four datasets, respectively. 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 minimum support in $\{25\%, 28\%, 30\%, 32\%, 35\%\}$ to run F^{2}IDC with WordNet for all datasets. Moreover, we set the minimum support values, ranging from 3 to 6%, to obtain the best results for FIHC.

It is apparent that the average accuracy of Bisecting k-means and FIHC are slightly better than that of F^{2}IDC in several cases. We argue that the exact number of clusters in a document set is usually unknown in real case, and F^{2}IDC is robust enough to produce stable, consistent and high-quality clusters for a wide range of number of clusters. This can be realized by observing the average overall F-measure values of all test cases. Notice that UPGMA is not available for large datasets because some experimental results cannot be generated for UPGMA, and we denoted them as NA. Since FIHC is not available for the documents of long average length, there is no experimental result generated on the WebKB dataset, and we also marked them as NA.

Table 4 Average overall F-measure comparison for four clustering algorithms on the four datasets

Dataset (# of natural classes)	# of clusters	F^2IDC (H)	FIHC (H)	UPGMA (H)	Bi. k-means (H)
Classic4 (4)	3	0.68 (3)*	0.51 (1)	NA	0.61 (5)
	15	0.70 (3)*	0.51 (1)	NA	0.59 (5)
	30	0.70 (3)*	0.52 (1)	NA	0.43 (5)
	60	0.69 (3)*	0.51 (1)	NA	0.28 (5)
	Average	0.69 (3)*	0.51 (1)	NA	0.48 (5)
Re0 (13)	3	0.56 (3)*	0.43 (1)	0.40 (3)	0.40 (3)
	15	0.53 (3)*	0.40 (1)	0.35 (3)	0.42 (3)
	30	0.52 (3)*	0.39 (1)	0.35 (3)	0.36 (3)
	60	0.52 (3)*	0.34 (1)	0.35 (3)	0.30 (3)
	Average	0.53 (3)*	0.39 (1)	0.36 (3)	0.37 (3)
R8 (8)	3	0.57 (3)	0.47 (1)	NA	0.59 (3)*
	15	0.44 (3)*	0.43 (1)	NA	0.42 (3)
	30	0.43 (3)*	0.43 (1)*	NA	0.36 (3)
	60	0.44 (3)*	0.43 (1)	NA	0.23 (3)
	Average	0.47 (3)*	0.44 (1)	NA	0.40 (3)
WebKB (4)	3	0.48 (1)*	NA	0.44 (1)	0.33 (3)
	15	0.49 (1)*	NA	0.43 (1)	0.19 (3)
	30	0.49 (1)*	NA	0.42 (1)	0.13 (3)
	60	0.49 (1)*	NA	0.36 (1)	0.07 (3)
	Average	0.49 (1)*	NA	0.42 (1)	0.18 (3)

NA means not available for the datasets
* The best competitor

Table 5 Improvement ratio for other three clustering algorithms on the four datasets

Dataset	Clustering algorithms				Improvement ratio		
	F^2IDC(H)	FIHC (H)	UPGMA (H)	Bi. k-means	FIHC	UPGMA	Bi. k-means
Classic4	0.69 (3)	0.51 (1)	NA	0.48 (5)	+0.35	NA	+0.43
Re0	0.54 (3)	0.39 (1)	0.36 (3)	0.37 (3)	+0.39	+0.50	+0.46
R8	0.47 (3)	0.44 (1)	NA	0.40 (3)	+0.07	NA	+0.18
WebKB	0.49 (1)	NA	0.42 (1)	0.18 (3)	NA	+0.17	+1.72

From the experimental result in Table 4, based on Formula (7), our proposed approach has gained $F(C)$ value improvement in average (as shown in Table 5) for the other three algorithms on four datasets. The percentage of improvement ratio ranges from 7 to 172% based on the increases of the $F(C)$ value.

4.4.2 The effect of enriching the document representation

As described in Sect. 3.2, when enriching the document representation, we utilize WordNet to exploit hypernymy for clustering. We now demonstrate the effect of adding hypernyms into the datasets as follows.

Table 6 The effect of enriching the document representation

Dataset	Classic4		Re0		R8		WebKB	
	F^{2}IDC	FIHC	F^{2}IDC	FIHC	F^{2}IDC	FIHC	F^{2}IDC	FIHC
Baseline	0.54	0.51	0.50	0.40	0.55	0.55	0.44	NA
$H1$	0.67	**0.51**	0.52	**0.39**	0.43	**0.44**	**0.49**	NA
$H2$	0.65	0.50	0.51	0.38	0.43	0.44	0.48	NA
$H3$	**0.69**	0.49	**0.53**	0.38	**0.47**	0.40	0.46	NA
$H4$	0.66	0.47	0.53	0.38	0.47	0.40	0.45	NA
$H5$	0.67	0.47	0.52	0.38	0.47	0.40	0.43	NA

Since FIHC obtained the best performance in terms of accuracy among the three comparing algorithms, we tested F^{2}IDC and FIHC by the baseline method and the addition of hypernyms of different levels. Table 6 shows the comparison of clustering results obtained by F^{2}IDC and FIHC, respectively. In Table 6, "Baseline" means that no hypernyms are added; "$H1$" corresponds to the addition of direct hypernyms; "$H2$" stands for the addition of hypernyms of first and second levels, and so on. We chose the minimum support, ranging from 4 to 8% to run the baseline result of F^{2}IDC for all datasets. The results in Table 6 show that FIHC decreases the clustering accuracy when increasing the levels of hypernyms. WordNet-based FIHC does not provide the improvement with respect to the baseline method. For the obtained results, the reasons could be

(1) Using hypernyms as additional features in the document enrichment process inevitably introduces a lot of noise into these datasets;
(2) Word sense disambiguation was not performed to determine the proper meaning of each polysemous term in documents [10].

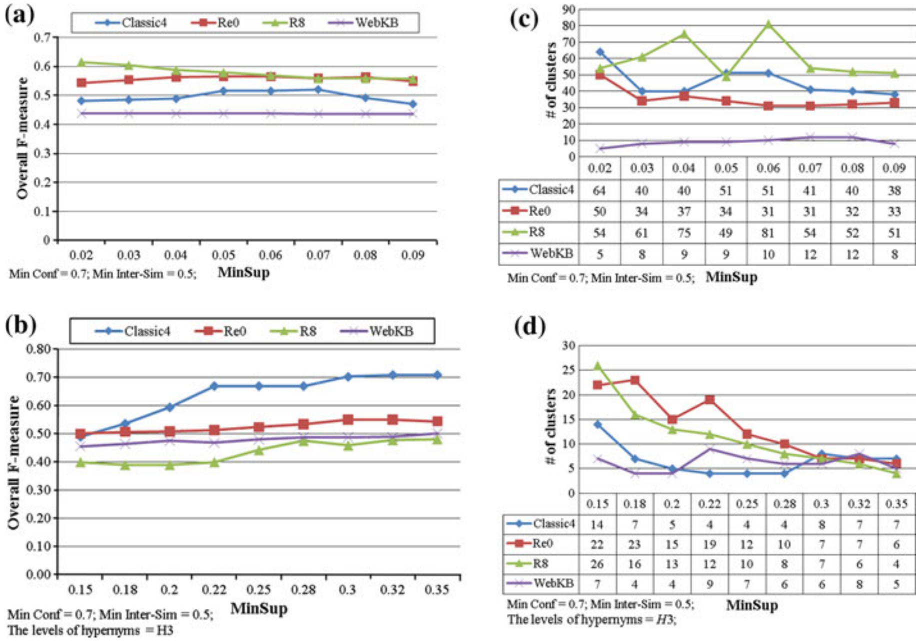
By Table 6, it is obvious that the average overall F-measure values of WordNet-based F^{2}IDC are superior to that of WordNet-based FIHC when adding hypernyms of the first, second, and third levels on almost all datasets, except for WebKB dataset. The performance of F^{2}IDC with the addition of direct hypernyms is better than that of F^{2}IDC with higher levels of hypernyms on WebKB dataset. Due to the longer average length of documents in WebKB dataset, higher levels of hypernyms may add more noise to the clustering process and decrease the clustering accuracy.

In contrast to WordNet-based FIHC, our approach can ameliorate the effect of adding hypernyms by filtering out noise for clustering. The use of WordNet for F^{2}IDC induces better clustering results on Classic4 dataset, while the improvements of the others are not particularly spectacular. In the case of the Reuters tasks, the limited improvement may not cause a particular worry. 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 [22].

To understand the reason why WordNet enhanced F^{2}IDC to perform better, a sample of the cluster labels generated by F^{2}IDC on Re0 dataset can be found in Table 7. Due to the rich semantic network representation provided by WordNet, F^{2}IDC with WordNet generates more general and meaningful labels for clusters. For example, the label 'commerce' produced by F^{2}IDC with WordNet is a more general concept than the labels 'sell' and 'trade' generated by F^{2}IDC without WordNet.

Table 7 Cluster Labels generated by F^{2}IDC algorithm on Re0 dataset

F^{2}IDC without WordNet	F^{2}IDC with WordNet
bank, dollar, currency, growth, industry, market, nation, rate, rise, rose, sell, trade	Activity, agent, assemblage, commerce, (commodity, good), currency, forecast, growth, merchant, nation, rate, record, (bush, rose, shrub)



	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
Classic4	64	40	40	51	51	41	40	38
Re0	50	34	37	34	31	31	32	33
R8	54	61	75	49	81	54	52	51
WebKB	5	8	9	9	10	12	12	8

	0.15	0.18	0.2	0.22	0.25	0.28	0.3	0.32	0.35
Classic4	14	7	5	4	4	4	8	7	7
Re0	22	23	15	19	12	10	7	7	6
R8	26	16	13	12	10	8	7	6	4
WebKB	7	4	4	9	7	6	6	8	5

Fig. 8 The accuracy test of F^{2}IDC for different MinSup values with the optimal cluster numbers determined by the clusters merging step algorithm

4.4.3 Sensitivity to various parameters

Figure 8a, b, respectively, depict the overall F-measure values of F^{2}IDC and WordNet-based F^{2}IDC when accepting different mandatory parameters, but ignoring the parameter values of the optional ones. We observed that high clustering accuracies are fairly consistent while MinSup are set between 2 and 9% for F^{2}IDC and set between 15 and 35% for WordNet-based F^{2}IDC. As KClusters is not specified in each test case, the clusters merging step in Algorithm 3 has to decide the most appropriate number of output clusters, which are shown in Fig. 8b, d for F^{2}IDC and WordNet-based F^{2}IDC, respectively.

Based on our test, we concluded a general observation that the best choice of MinSup can be set between 4 and 8% for F^{2}IDC, and set between 25 and 35% for WordNet-based F^{2}IDC. Nevertheless, it cannot be over emphasized that MinSup should not be regarded as the only parameter for finding the optimal accuracy.

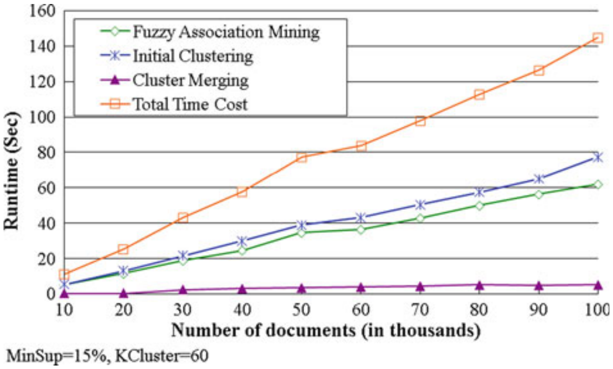


Fig. 9 Scalability of F^{2}IDC

4.4.4 Efficiency and scalability

To analyze the scalability of our algorithm, we got 100,000 documents from RVC1 (Reuters Corpus Volume 1) dataset [14], which contains news from Reuters Ltd. There are three category sets: Topics, Industries, and Regions. In our experiments, we consider the Topics category set, which includes 23,149 training and 781,265 testing documents. Before clustering this dataset, documents were parsed by converting all terms in documents into lower case, removing stop words, and applying the stemming algorithm.

Figure 9 shows the runtimes with respect to the different sizes of RVC1 dataset, ranging from 10K to 100 K documents, for different stages of our algorithm. The figure also shows that fuzzy association mining and initial clustering stages are the most two time-consuming stages in our algorithm. In the clustering process, most of the time is spent on constructing initial clusters and its runtime is almost linear with respect to the number of documents. As the efficiency of the fuzzy association rule mining is very sensitive to the input parameter MinSup, the runtime of F^{2}IDC is inversely related to MinSup. In other words, runtime increases as MinSup decreases.

5 Conclusion

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 improving the clustering quality. Particularly, most of the current documents clustering algorithms, including FIHC, do not consider the semantic relationships among the terms. In this paper, we derived an effective Fuzzy Frequent Itemset-based Document clustering (F^{2}IDC) approach that combines fuzzy association rule mining with the external knowledge, WordNet, for grouping documents. 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.

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

(1) Our approach facilitates the integration of the rich knowledge of WordNet into textual documents by effectively filtering out noise when adding hypernyms into documents and generating more conceptual labels for clusters.
(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 Bisecting *k*-means and UPGMA, are sensitive to the number of clusters.

In the future, we will explore some further issues. First, we will extend F^{2}IDC to generate overlapping clusters for providing multiple subjective perspectives onto the same document to enhance its practical applicability. Second, we intend to propose an efficient incremental clustering algorithm [7,11] for assigning a new document to the most similar existing cluster. Third, we will consider the abundant structural relation within Wikipedia, such as hyperlinks and hierarchical categories, to improve the performance of clustering [27].

Acknowledgments 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. Agrawal R, Imielinski T, Swami AN (1993) Mining association rules between sets of items in large databases. In: ACM SIGMOD international conference on management of data, pp 207–216
2. Beil F, Ester M, Xu X (2002) Frequent term-based text clustering. In: International conference on knowledge discovery and data mining (KDD'02), pp 436–442
3. Chen CL, Tseng FSC, Liang T (2008) Hierarchical document clustering using fuzzy association rule mining. In: The 3rd international conference of innovative computing information and control (ICICIC2008), pp 326–330
4. Chen CL, Tseng FSC, Liang T (2010) Mining fuzzy frequent itemsets for hierarchical document clustering. Inf Process Manag 46(2):193–211
5. Craven M, DiPasquo D, McCallum A, Mitchell T, Nigam K, Slattery S (1998) Learning to extract symbolic knowledge from the World Wide Web. In: AAAI-98
6. Cutting DR, Karger DR, Pederson JO, Tukey JW (1992) Scatter/gather: a cluster-based approach to browsing large document collections. In: The 15th international ACM SIGIR conference on research and development in information retrieval, pp 318–329
7. Exarchos TP, Tsipouras MG, Papaloukas C, Fotiadis DI (2009) An optimized sequential pattern matching methodology for sequence classification. Knowl Inf Syst 19(2):249–264
8. Fung B, Wang K, Ester M (2003) Hierarchical document clustering using frequent itemsets. In: SIAM international conference on data mining (SDM'03), pp 59–70
9. Hong TP, Lin KY, Wang SL (2003) Fuzzy data mining for interesting generalized association rules. Fuzzy Sets Syst 138(2):255–269
10. Hotho A, Staab S, Stumme G (2003) Wordnet improves text document clustering. In: SIGIR international conference on Semantic Web Workshop
11. Huang Z, Sun S, Wang W (2010) Efficient mining of skyline objects in subspaces over data streams. Knowl Inf Syst 22(2):159–183
12. Kaya M, Alhajj R (2006) Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rule mining. Appl Intell 24(1):7–15
13. Kushal Dave DMP, Lawrence S (2003) Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: The 12th international conference on World Wide Web (WWW)
14. Lewis DD, Yang Y, Rose TG, Li F (2004) RCV1: a new benchmark collection for text categorization research. J Mach Learn Res 5:361–397
15. Liu B, Hsu W, Ma Y (1999) Pruning and summarizing the discovered associations. In: The ACM SIGKDD conference on knowledge discovery and data mining, pp 125–134
16. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: The 5th Berkeley Symposium on Mathematical Statistics and Probability, pp 281–297
17. Mandhani B, Joshi S, Kummamuru K (2003) A matrix density based algorithm to hierarchically co-cluster documents and words. In: The 12th international conference on World Wide Web (WWW), pp 511–518

18. Martín-Bautista MJ, Sánchez D, Chamorro-Martínez J, Serrano JM, Vila MA (2004) Mining web documents to find additional query terms using fuzzy association rules. Fuzzy Sets Syst 148(1):85–104
19. Michenerand CD, Sokal RR (1957) A quantitative approach to a problem in classification. Evolution 11:130–162
20. Miller GA (1995) WordNet: a lexical database for English. J Commun ACM 38(11):39–41
21. Porter MF (1980) An algorithm for suffix stripping. Program 14(3):130–137
22. Scott S, Matwin S (1998) Text classification using WordNet hypernyms. In: Proceedings of Worksh Usage of WordNet in NLP Systems at COLING-98, pp 38–44
23. Sedding J, Kazakov D (2004) WordNet-based text document clustering. In: COLING-2004 workshop on robust methods in analysis of natural language data
24. Shihab K (2004) Improving clustering performance by using feature selection and extraction techniques. J Intell Syst 13(3):135–161
25. Singhal A, Salton G (1993) Automatic text browsing using vector space model. Technical Report, Department of Computer Science, Cornell University
26. Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. In: The 6th ACM SIGKDD international conference on knowledge discovery and data mining (KDD)
27. Wang P, Hu J, Zeng H-J, Chen Z (2009) Wikipedia knowledge to improve text classification. Knowl Inf Syst 19(3):265–281
28. Wei C, Hu P, Dong YX (2002) Managing document categories in e-commerce environments: an evolution-based approach. Eur J Inf Syst 11(3):208–222
29. Willett P (1988) Recent trends in hierarchic document clustering: a critical review. Inf Process Manag 24(5):577–597
30. Xu W, Gong Y (2004) Document clustering by concept factorization. In: The 27th ACM SIGIR conference on research and development in information retrieval, pp 202–209
31. Yu H, Searsmith D, Li X, Han J (2004) Scalable construction of topic directory with nonparametric closed termset mining. In: The IEEE international conference on data mining series (ICDM 2004), pp 563–566
32. Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353

Author Biographies

Chun-Ling Chen is a postdoctoral research fellow of the Institute of Statistical Science, Academia SINICA, Taiwan, ROC. She received her Ph.D. degree in computer science from National Chiao Tung University, Taiwan, ROC, in 2010. Her research interests include database, object-oriented conceptual modeling, information retrieval, text mining, and machine learning.

Frank S. C. Tseng, Ph.D. 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. From 1993 to 1995, he served the military obligation in the General Headquarters of ROC Air Force. Dr. Tseng is one of the winners of Acer Long Term Ph.D. dissertation prize in 1992. 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 and the chairperson 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, Special Interest Group on Management of Data.

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.