Appl Intell (2010) 33: 67–78
DOI 10.1007/s10489-010-0243-2

Constructing tree-based knowledge structures from text corpus

Sheng-Tun Li · Fu-Ching Tsai

Published online: 21 July 2010

Abstract A knowledge structure identifies how people think and displays a macro view of human perception. By discovering the hidden structural relations of knowledge, significant reasoning patterns are retrieved to enhance further knowledge sharing and distribution. However, the utilization of such approaches is apt to be limited due to the lack of hierarchical features and the problem of information overload, which make it difficult to enhance comprehension and provide effective navigation. To address these critical issues, we propose a new approach to construct a tree-based knowledge structure from corpus which can reveal the significant relations among knowledge objects and enhance user comprehension. The effectiveness of the proposed method is demonstrated with two representative public data sets. The evaluation results show that the method presented in this work achieves remarkable consistency with the domain-specific knowledge structure, and is capable of reflecting appropriate similarities among knowledge objects along with hierarchical implications in the document classification task.

The work of S.-T. Li is partly supported by National Science Council, Taiwan under contract NSC98-2410-H-006-007.

S.-T. Li (✉) · F.-C. Tsai
Institute of Information Management, National Cheng Kung University, Tainan, Taiwan, ROC
e-mail: stli@mail.ncku.edu.tw

F.-C. Tsai
e-mail: r7895104@mail.ncku.edu.tw

S.-T. Li
Department of Industrial and Information Management, National Cheng Kung University, Tainan, Taiwan, ROC

Keywords Knowledge structure · Hierarchical feature · Information retrieval · Knowledge management · Document classification

1 Introduction

In a fast-changing environment, knowledge is one of the most precious assets of a corporation and a key to enhancing organizational competitiveness. Numerous case studies of international companies have demonstrated that considerable benefits can be achieved by leveraging knowledge to meet evolving business needs. However, tacit knowledge can easily disappear when employees leave the firm, and, even if they stay, its usability is limited due to difficulties with regard to access, sharing and distribution. In order to preserve tacit knowledge and make it available to all members of an organization, knowledge engineering methods have been developed to transform it into more explicit forms. However, how to adequately organize the explicit knowledge thus obtained into a systematic model which can represent the whole picture of domain-specific knowledge and provide effective problem solving capabilities remains a considerable challenge.

Knowledge structures (KS) with the appropriate topology have been identified as an effective expression of explicit knowledge which can provide concrete comprehension and figurative navigation of the information they contain. KS provide the ability to discover the hidden relations among codified knowledge objects (KO), and thus produce a more holistic view of a subject domain. The information embedded in KS not only depicts granular KO, but also represents the coherent reasoning framework in terms of the relevant structural implications. In addition, various methods of organizing KO portray different reasoning patterns,

even though they are composed of identical objects [1]. This phenomenon has also been investigated in the educational field [2, 3]. For example, the distinct KS between students explains their different problem-solving skills after taking the same class. Undoubtedly, well organized KS lead to superior comprehension and more effective reasoning capabilities when facing decision making tasks.

Different KS formats are discussed in earlier research [4], and there are two major concerns for creating a knowledge structure: (1) how to formulate human cognition, and (2) how to make it in a form that users can visualize. With regard to the first question, people's cognition is believed to have generally hierarchical properties, and when individuals are asked to categorize an object in a neutral setting without further instructions, they are very likely to provide hierarchically-organized categories [5]. Accordingly, hierarchical KS are more appropriate to describe human cognition rather than non-hierarchical ones. Turning to the second question, human visualization capacity is limited with regard to information processing, and the more complex the structure is, the more difficult it is for humans to realize its morphology. Therefore, KS with large numbers of KO or links between objects can reduce the effectiveness of knowledge navigation and visualization [6–8]. To solve these two problems, we propose a new method to construct a tree-based KS (TKS) to provide both comprehensiveness through hierarchical features and concise KO relations to avoid information overload. The remainder of this article is organized as follows. In Sect. 2, we review various knowledge representation and hierarchical knowledge structure construction approaches. Section 3 introduces the fundamental definitions used in this research, while the TKS construction algorithm is illustrated in detail in Sect. 4. Section 5 discusses the experimental results from two data sets, and then Sect. 6 concludes this paper and gives some suggestions for future work.

2 Previous research

2.1 Knowledge representation

How to accurately represent experts' knowledge is the most fundamental challenge when starting to develop knowledge-based applications. The knowledge, or what experts call *intuition* or *natural ability* in solving difficult problems, is hard to describe and organize as principals for providing direct support to novices. Making knowledge visible is an instinctive method so that it can be better accessed, valued or generally managed by knowledge workers [1, 9]. Knowledge maps are often used as cognitive tools to provide a direct view of knowledge, and are composed of a set of nodes and links [10]. Nodes are widely recognized as intuitive KO, whereas links describe the correlations between them. Moreover, graphical knowledge maps also contain prognostic insights, principles, basic assumptions and relations [4, 11]. Previous studies of knowledge maps are divided into two main categories: non-hierarchical and hierarchical. Networks, which are cyclic weighted graphs, are the most typical formats of the non-hierarchical class [12]. Xu et al. [13] proposed a framework for criminal knowledge discovery through a network structure, while Scvaneveldt [14] and Bradley et al. [1] used PathFinder techniques to generate a reduced network containing a human cognitive model with the shortest paths. Although the nodes in this network are allowed to connect to each other without further restrictions, its complex linkages, i.e. cyclic redundancy and cycles, can hinder awareness of the knowledge structure. In addition, the absence of a root is another major difference between non-hierarchical and hierarchical structures. A root provides the ground level of hierarchy and the initial point for successive growth. Without a root, the general or specific features of KO cannot be distinguished according to their located level. These shortcomings with the non-hierarchical structure seriously impede knowledge representation because of the lack of concise and comprehensive subsumption features. As to the hierarchical structure, it is the most common design used to express multilayer perception, and is widely used in practice. However, several disadvantages, such as labeling and multiple inheritance, also exist in the state-of-the-art hierarchical techniques. Automatic labeling, naming according to the most frequent word or several frequent words from the composed instance, may cause ambiguities by giving the different nodes the same title [15]. Ontology construction is proposed as an important method to represent real-world knowledge, however, the complex structure of such systems also hinders human cognition in terms of multiple inheritance. Therefore, most of ontology studies focus on linguistic and semantic applications instead of visualization [16–19]. A more detailed analysis of these problems and suggested remedies are discussed in the next section.

2.2 Hierarchical knowledge structures

Hierarchies are the most straightforward mechanism for humans to utilize in knowledge modeling with respect to subordinate and superordinate associations. Lattices and trees are two representative structures that are based on a deep understanding of the philosophical background of knowledge engineering [20, 21]. The main application of lattice structures, i.e. formal concept analysis (FCA) [22], is to analyze data and provide well structured information for knowledge workers, such as investigating and interpreting implicit relations, deriving implication rules and facilitating knowledge acquisition [23–25]. Although FCA has demonstrated

its ability with regard to retrieval tasks and implicit relation discovery [26], it has two drawbacks that should be mentioned. First, many tangled nodes and linkages which are inapplicable for knowledge visualization are produced in lattices in order to map n-dimensional data to a two-dimensional space. Furthermore, in practice, complex lattice structures are too large to display visually in their entirety. Second, FCA cannot handle a very large amount of source data [21] due to its high complexity of $O(2^n)$ [22].

As to tree structures, they are defined as acyclic graphs which are composed of one root node, with several internal and leaf nodes and edges. In previous research, tree structures have been successfully applied to display massive and complex information with both comprehensible visualization and easy navigation. In the knowledge management field, such structures have been adopted to formulate a layered thematic knowledge map [4]. In tree structures, nodes and weighted edges represent knowledge objects and their correlations, respectively. Although many methodologies can generate a tree structure, some are not capable of indicating the upper and lower relations if they lack the important starting point, i.e., the root, such as the minimum spanning tree (MST) approach. Strictly speaking, MST is not a tree structure, but more like a condensed network which successfully holds the most potent links in a unique structure. With the hierarchical agglomerative clustering (HAC) approach [27], a bottom-up clustering algorithm treats each node as a singleton and then iteratively merges the most similar pairs until the number of clusters corresponds to a predefined criterion. The other top-down clustering method, bisecting K-means algorithm [28], starts with a single cluster and choose a cluster to split once at a time until the desired number of clusters is reached. Nevertheless, how to label the merged or split internal nodes, which is very important with regard to facilitating complete understanding of domain knowledge, remains a significant challenge in both top-down and bottom-up clustering approaches [29].

In a hierarchical structure, multiple inheritance causes ambiguity for nodes which contain cross-links to different superordinate nodes. For example, shoes for girls are not only daily necessities, but also often a kind of decorative accessory. To map the above example in a tree structure, there should be two respective links from shoes to its corresponding parent nodes, decorative accessories and daily necessities. However, we believe that it is appropriate to classify each element into only one category in a specific domain to avoid ambiguity. In addition, the complex linkages from multiple inheritances violate the fundamental notion of knowledge representation, and impede both recognition and comprehension. Although a number of previous studies have explored this issue, the problem of multiple inheritance has not yet been resolved [23, 24, 30–33]. An example of a proposed TKS in computer hardware domain, with one root node and single inheritance, is illustrated in Fig. 1.

3 Tree-based knowledge structures

In this section, two fundamental definitions of TKS are identified, and these provide the natural hierarchical features necessary to guide the TKS constructed in this work and avoid multiple inheritance problems. In addition, sibling independence, which is rarely discussed in previous research, is also considered to determine the appropriate frame of TKS.

3.1 Hierarchical features

The definition of hierarchy is considered the fundamental principle for locating nodes and growing edges. Partial order is often used to present superordinate and subordinate relations of a hierarchy [31]. Therefore, the similar notion is applied to define the tree structure, as shown in Definition 1.

Definition 1 (Tree structure) A Tree structure $:= (N, root, \leq)$ consisting of a set of nodes N, such that $\forall n \in N$ and $\forall n \leq root$, and where the root is located at the highest level of the tree structure and $root \in N$.

From Definition 1, a tree structure is composed of a set of nodes including a root which is located at the top level of a hierarchy. The superordinate and subordinate relations are formulated by partial order, $\leq$. Based on Definition 1, we further extend partial order relations and propose the definition of a hierarchical feature by way of comparing mutual similarities among nodes, as shown in Definition 2.

Definition 2 (Hierarchical feature) $\forall n \in N$, $sim(n, ancestor_i(n)) < sim(n, ancestor_j(n))$, if $n \leq ancestor_i(n)$, $n \leq ancestor_j(n)$ and $i < j$.

In Definition 2, $ancestor_i(n)$ represents the ancestor of node n which is located at level i and $sim(n, ancestor_i(n))$ stands for the similarity between node n and $ancestor_i(n)$. Following the definition of a tree structure, we define that the further a node is from the root, the larger its level number, which means that the level of the root equal to one is the smallest number in the entire tree structure. The main concept of the hierarchical feature is that for any two superior nodes, $ancestor_i(n)$ and $ancestor_j(n)$, of node n, their mutual similarities are disproportionate to their interval edge numbers. For example, we assume mammal, tiger and Siberian tiger are ancestor-descendant nodes in the animal taxonomy, as shown in Fig. 2.

Obviously, tiger inherits mammal and Siberian tiger inherits tiger. We can observe that the two nodes with one crossing link, Siberian tiger and tiger, are apparently more similar than nodes with two crossing links, Siberian tiger

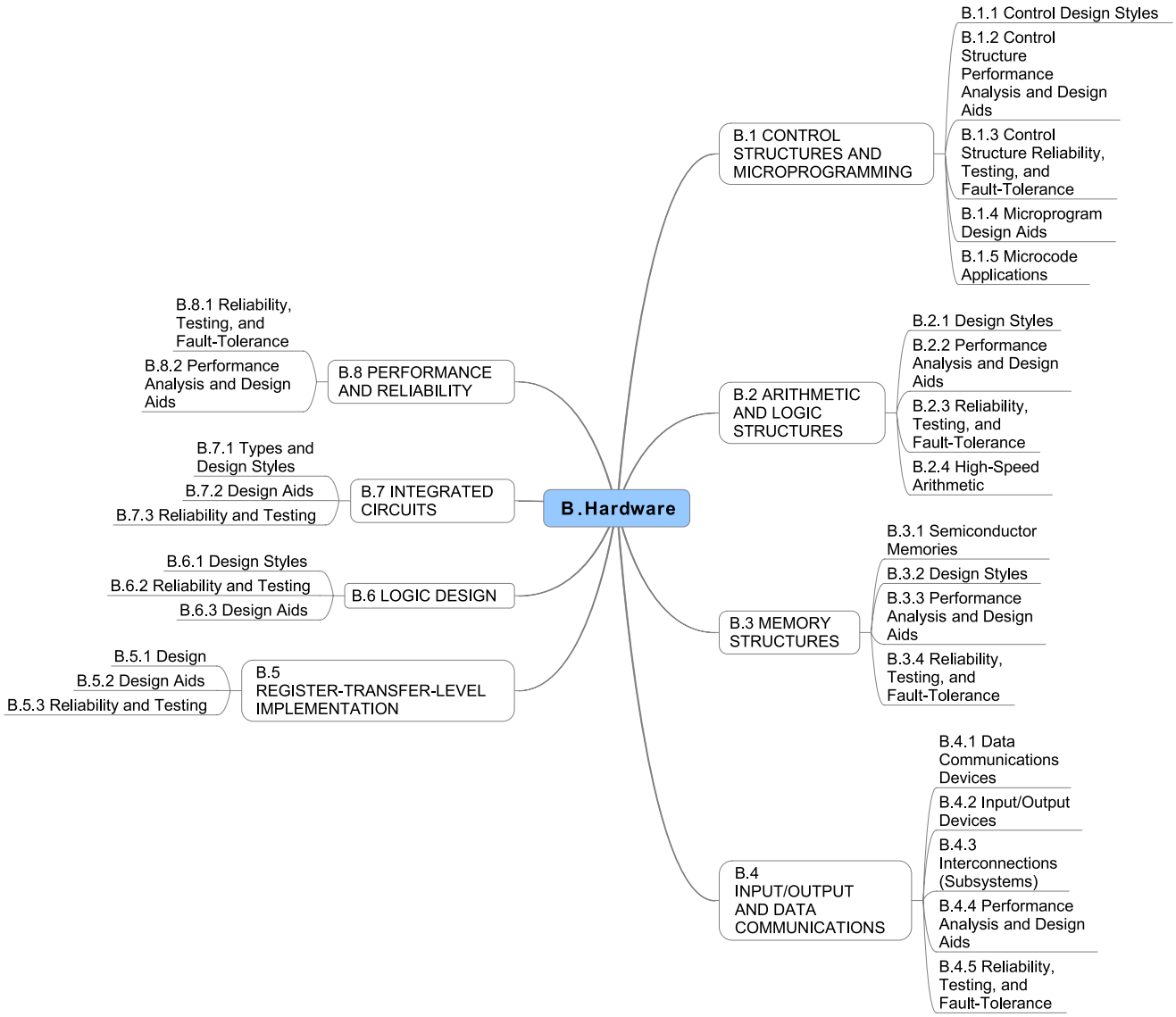

Fig. 1 An example of a tree structure

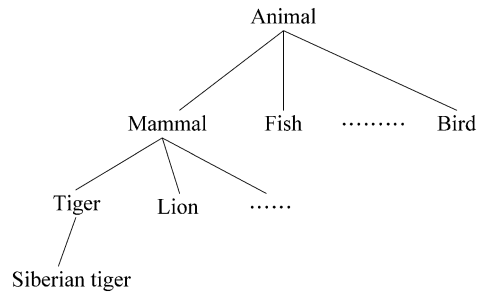

Fig. 2 An example of animal taxonomy

and mammal. According to Definition 2, the following equation holds:

$$sim(\text{Siberian tiger}, \text{Mammal}) < sim(\text{Siberian tiger}, \text{Tiger})$$

Moreover, if we take animal as the basis of similarity comparison toward other nodes, by analyzing their mutual similarities, we can examine the proper location for each node in TKS. For example, in Fig. 2, the degrees of similarity between animal and its descendants, mammal, tiger and Siberian tiger, are high, medium and low, respectively; hence, we should locate the most similar node, mammal, nearer to animal than other two nodes in TKS. It is worth noting that, by applying the basic nature of a hierarchical feature in Definition 2, we do not have to define and select features for each node in discriminating hierarchical relations, such as is required with the attribute selection in FCA. Instead, the general and specific features are clearly discriminated by comparing mutual similarities.

3.2 Sibling independence

Siblings in a tree structure are defined as nodes which are located in the same layer and share the same parent. In the tree construction process, sibling independence is a critical factor to determine the width and depth of a tree. However, only few studies focus on this important issue. If we apply stronger sibling independence criterion in the tree structure, the nodes which are supposed to be siblings would instead be viewed as child nodes, because they are no longer different enough to fit the stronger criterion. More overlap between nodes indicates they are more similar but have looser independence, and vice versa. In this research, we proposed a natural criterion of sibling independence to decide the location of each node, which means the sibling independence of node n_i is understood by examining how similar it is to its parent. For example, if the similarity between node n_i and its sibling is higher than with its parent, then node n_i should be located in the next level instead of the current one. The formal definition of sibling independence is shown as

$$\textit{sibling independence}(n_i) = \begin{cases} \textit{true}, & \forall(n_j, n_i) \in \textit{children}(n), \\ & \quad \textit{sim}(n_j, n_i) < \textit{sim}(n, n_i) \\ \textit{false}, & \text{otherwise} \end{cases} \quad (1)$$

where *children(n)* represents the child set of node n. n_j are the siblings of node n_i and the children of node n.

This criterion also fits the tree definition that the correlation between parent and child should be stronger than between siblings. Theoretically, siblings with stronger independence are more distinguishable and capable of avoiding the subsumption effect from one another. However, a straight vertical tree without siblings might be produced if we only consider raising sibling independence to the limit. In contrast, the hierarchical feature would be diminished with excessive loose sibling independence, and produce a flat, list-like structure. For the purpose of accurately codifying TKS from human cognition, we adopted different degrees of sibling independence as the parameter for KS construction. Further analysis of the impact of this is given later in this work.

4 Construction of tree-based knowledge structures

Figure 3 depicts the three main stages of the TKS construction framework, which are knowledge codification, similarity refinement and TKS construction. The detailed procedures are discussed in the following subsections.

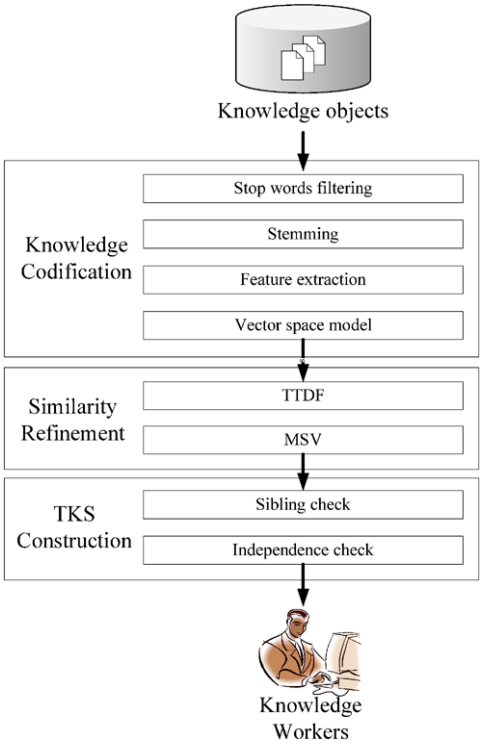

Fig. 3 Three stages of TKS construction

4.1 Knowledge codification from text corpus

The various perspectives of knowledge are widely embedded in and carried through multiple entities, but especially in documents, which are the most common source of explicit knowledge. Thus, we used documents to formulate KO with specific descriptions of the corresponding topics. To aid the explanation of how TKS are constructed, "document", "node" and "KO", which all represent the same entity, will be used interchangeably in the following discussion. Since the mutual similarities of KO are utilized for constructing TKS, traditional information retrieval (IR) techniques are applied in this stage to calculate the similarities of documents [34]. First, the data preprocessing stage includes tokenization, stop words filtering and stemming. Tokenization represents the division of text into words or terms. Stop words filtering omits words like prepositions and pronouns which are not representative of the documents. Stemming identifies the root form of words by removing suffixes and prefixes. Second, the weight of each word is measured by term frequency (TF) and inverse document frequency (IDF) in order to produce the vectors which represent document features. TFIDF is a statistical measure which is often used in IR and text mining to evaluate the importance of words in a corpus. TF provides the occurrence frequencies of terms that appear in a document, and IDF is used to distinguish the relevant terms in the corpus. Terms with high TFIDF are treated as important features with regard to representing the document.

Finally, the mutual similarities among documents are calculated using the vector space model (VSM), which is an algebraic model that transforms documents into vectors in a multi-dimensional space [34]. VSM is capable of facilitating the estimation of similarity among documents by calculating the inner product of vectors.

4.2 Similarity refinement

In most cases, mutual similarities among documents are derived by calculating the statistical distribution of words. However, synonyms, antonyms and writing styles usually affect the similarity measure in IR studies. Thus, we attempt to explore other information from the text to refine the analysis of similarities. The idea is that if one can identify upper level nodes in advance and connect them in the upper part of the structure, then the final result will be more reasonable. The special characteristic of upper level nodes is referred to as abstractness, which means the degree of generality of nodes within the repository, and this is composed of the following measures.

(1) Mutual Similarity Variance (MSV)

As mentioned above, higher level nodes are more general than lower level ones, and thus the similarity distribution of nodes at higher levels should be more diverse than those at lower ones. For this purpose, variance, a statistical measure to represent the degree of spread compared with the mean, is used to evaluate abstractness. Therefore, MSV is an index which stands for the degree of abstractness, which can revise the initial similarity measures and produce more accurate ones. The MSV for each node is calculated by

$$MSV(n_i) = \frac{\sum_{j=1}^{|N|-1}(sim(n_i, n_j) - \overline{n_i})^2}{|N|-1}, \quad \text{where } n_i, n_j \in N \text{ and } i \neq j \tag{2}$$

where $\overline{n_i}$ is the average of mutual similarities of node n_i to other nodes. We ignore the self similarity, which is always equal to 1, in calculating variance, and thus the denominator of MSV is $|N| - 1$.

(2) Top Term Document Frequency (TTDF)

TTDF is used to estimate the generality of document d_j with the cumulative document frequency of *topN* frequent terms k_i in the corpus. The *topN* frequent terms of a document can be treated as its typical features [35]. By summarizing the df of k_i in d_j, one can derive the generality of d_j. Sanderson et al. [33] pointed out that the generality and specificity of terms are determined according to their apparent frequency, and thus the TTDF index can successfully reflect the abstractness of a document. TTDF is obtained by using

$$TTDF(d_j) = \sum_{i=1}^{topN}\sum_{j=1}^{|N|-1} df_j(k_i), \quad 1 < j < |N| - 1 \text{ and } 1 \leq i \leq topN \tag{3}$$

Where *topN* is the number of top frequent terms in document d_j and $df_j(k_i)$ represents the document frequency, which is the number of documents in a corpus that contain term k_i in document d_j. In order to enrich the information gathered and simultaneously avoid noisy data, we use the top three most frequent terms, as suggested in Sanderson's work [33], to represent document features and calculate their abstractness.

It is worth noting that the effect of abstractness is not the same at every level of TKS. Since the locations of lower level nodes are determined by upper level ones, a node that possesses a high degree of abstractness should be moved from its current level to a higher one in order to have a greater influence on deciding the location of its subordinates. In contrast, a low degree of abstractness indicates that a node should be located at a lower level to decrease its influence. To emphasize the influence caused by high abstractness in TKS, we adopt an exponential operation to enlarge the influence of abstractness in a nonlinear fashion. The abstractness of node n_i is computed as

$$abstractness(n_i) = \frac{MSV'(n_i)^e + TTDF'(n_i)^e}{2}, \quad e > 1 \tag{4}$$

where $MSV'(n_i)$ and $TTDF'(n_i)$ ranging from 0 to 1 are the normalized MSV and TTDF, respectively, and e is their exponential parameter. Equation (5) depicts the weighted average operation of revised similarity, which is averaged with a weight parameter, α, to decide the degree of refinement by abstractness.

$$rev\_sim(n, n') = (1-\alpha) \times sim(n, n') + \alpha \times abstractness(n'), \quad 0 \leq \alpha \leq 1 \tag{5}$$

4.3 TKS construction algorithm

The TKS construction algorithm is essentially composed of five steps, as shown in Fig. 4. The algorithm starts from assigning the most likely root according to user's intention from a set of predefined nodes (concepts). Theoretically, each node has the same opportunity to become the root. However, the root in a hierarchy should contain the most universal attribute that is inherited by other nodes. In knowledge management, the root node should be the most abstract concept and one that is relevant to all other concepts. Therefore, the user chosen root is expected to be the most general concept according to his/her perspective toward a specific

domain. In Step 2, the remaining nodes which meet the sibling independence criterion will sequentially connect to the root according to their degree of similarity, from high to low. The purpose of prioritizing to connect the nodes with high similarities to their parent is to preserve the most significant links in TKS. The unlinked nodes are then moved down to the next level and treated as the descendants of current level ones. In order to decide which branch each unlinked node belongs to, the unlinked nodes are classified into different groups according to their similarities to the nodes located one level above the current one. For each subordinate group, Step 4 is recursively performed to link the qualified nodes which meet the sibling independence and hierarchical feature criteria simultaneously, and move the unlinked nodes to the next level to become the new subordinate groups. Ultimately, the construction process is completed when all nodes are connected to the TKS. However, the process is stopped if no qualified nodes in the remaining subordinate groups can be connected. In this study, we ignore incomplete TKS due to their inability to reflect appropriate hierarchical relations.

Input: N: a set of predefined nodes, each node represents a concept in the knowledge repository.
Output: A tree-based knowledge structure that represents well organized domain knowledge composed of predefined concepts.
Step 1 Select an appropriate node n_i from N as the root chosen by the user.
Step 2 Link the qualified child nodes which meet the sibling independence criterion to the root.
Step 3 Categorize all unlinked nodes into subordinate groups corresponding to the child nodes in Step 2.
Step 4 For each subordinate group

a. Link the nodes which meet the sibling independence and hierarchical feature criteria to TKS and move the remaining nodes down to the next level as a new subordinate group corresponding to the linked nodes in this step. Recursively perform Step 4 for the newly subordinate groups located at the next level.
b. If no node meets both the sibling independence and hierarchical feature criteria simultaneously, stop the algorithm.

Step 5 Return the TKS

Fig. 4 The five main steps of the TKS construction algorithm

To illustrate how to build a TKS more clearly, we give an example, as shown in Fig. 5, which corresponds to the steps of the TKS construction algorithm in Fig. 4. In Step 1, users choose the most likely node as the root, which is *node A* in our example. In Step 2, all remaining nodes which meet the sibling independence criteria link to *node A* sequentially according to their similarity to the parent. In Step 3, after *nodes B* and *C* are connected, the rest of the nodes are categorized into the subordinate groups of *nodes B* and *C* and treated as their descendants. In Step 4a, each subordinate group member which meets the sibling independence and hierarchical feature criteria is linked to its parent. Step 4a is recursively called until all nodes are connected to TKS, and the process then moves on to Step 5, which returns the completed TKS. However, in certain situations there are some nodes, like *node H*, that cannot be linked because of the violation of sibling independence or hierarchical feature crite-

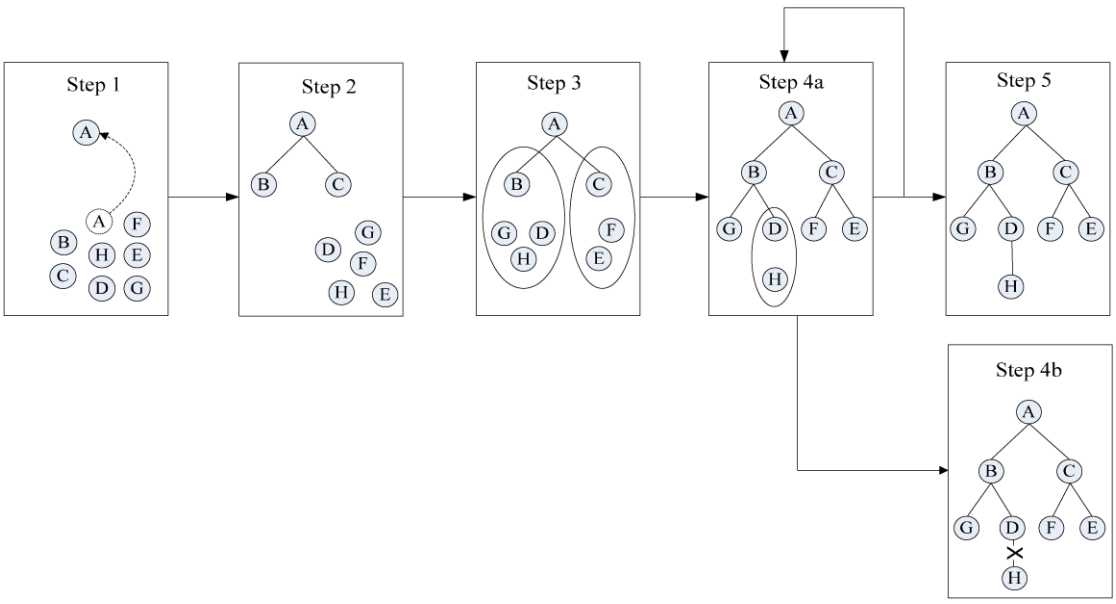

Fig. 5 An example of the TKS construction process

Table 1 The statistical information of the ACM gold standards

Gold standard	Nodes	Height	Documents	Terms
ACM CCS, category I	51	3	1,020 abstracts	154,148
ACM CCS, category B	37	3	740 abstracts	113,997

Table 2 The statistical information of the Reuters corpus

Category	Documents	Training set	Testing set
Earn	3809	3428	381
Acq	2212	1191	221
Crude	375	337	38
Trade	338	304	34
Interest	319	287	32
Money-fx	271	244	27
Ship	239	215	24
Grain	43	39	4

ria. When Step 4b occurs, the TKS construction process will be terminated.

5 Evaluation and discussion

We conduct two experiments to evaluate the effectiveness of the proposed TKS. In the first experiment, we choose a public tree structure from the Internet as the gold standard, and then the same nodes which belong to this standard are reconstructed using the TKS construction algorithm. We then assess the effectiveness by investigating the consistency between the gold standard and the constructed TKS. In the second experiment, we obtain the term-term similarities based on hierarchical information from TKS instead of traditional IR techniques. Two similarity measures, TKS and traditional IR, are applied to a document classification problem. We assess the utility of document classification using precision, recall and f-measure across categories in the evaluation dataset. In the following section, we first describe the two experimental datasets, and then analyze the evaluation results.

5.1 Data collection

In the first experiment, the Association for Computing Machinery (ACM) Computing Classification System (CCS) is applied as the gold standard (http://www.acm.org/about/class/). ACM CSS is an acyclic tree based categorization scheme without multiple inheritances, which fits our definition of a tree structure, and which accurately reflects the essential structure of the fields of computer science and information systems. Articles related to the same topic are be classified into the same category. Among various categories (from A to K), we choose B (hardware) and I (computing methodologies) in our research, as they have more subcategories compared to other items. Each subcategory is represented by a document which is composed of 20 abstracts randomly selected from it. However, several subcategories with less than 20 abstracts are eliminated to avoid the unreliable similarities caused by insufficient data. Table 1 depicts the statistics of the two categories in this experiment.

The second dataset for document classification is Reuters 21578, which is the collection of documents that appeared on Reuters news service in 1987. The Reuters 21578 dataset has been used in many document classification studies. Since this experiment only focuses on single class classification, we omit documents which are classified into multiple categories or not classified into any categories. Table 2 shows the statistics of the training and testing corpus of the Reuters dataset.

5.2 Evaluation of the gold standards

Since the TKS construction is affected by a number of parameters, i.e. TFIDF, degree of similarity refinement by abstractness (α), power of exponential operation and sibling independence, it is very time-consuming for the experimental process to determine the ideal ones. To better understand the adequacy and the effects of these, an experimental design process should be applied to decrease the experimental cost and determine the optimal parameters. The TFIDF criteria influence the number of features and form a basis for making relevant decisions [36]. Higher TFIDF successfully avoid the impact from noisy terms, although some information might be lost because fewer features are kept in document vectors. Different levels of power are designed to represent the various exponential curves for abstractness. α demonstrates the various degrees of influence on similarity adjustment produced by abstractness. In order to investigate

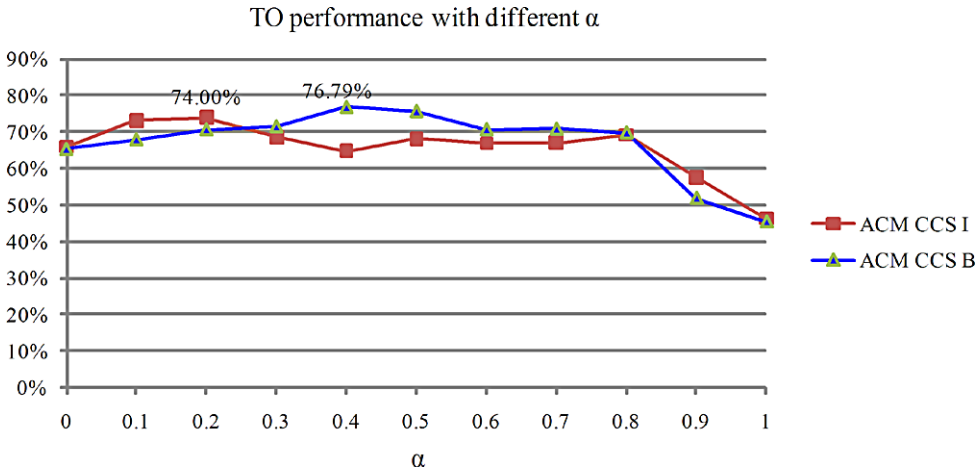

Fig. 6 The TO performance at parameter α between the range from 0 to 1

Table 3 The optimal set of parameters in the TO evaluation

Gold standard	TFIDF	Power	α	Sibling independence	TO
ACM CSS I	0.02	2	**0.2**	0.9	**74.00%**
			0		65.99%
ACM CSS B	0.02	2	**0.4**	0.9	**76.79%**
			0		65.35%

the structural impacts on TKS, different degrees of sibling independence are tested to determine the appropriate width and depth of the structure.

To validate the performance of this experiment, we adopt taxonomy overlap (TO) [31] to demonstrate the consistency between the gold standards and the TKS produced in this work. Although various measures have been proposed to evaluate consistency between two structures [37, 38], TO emphasizes the importance of upper level nodes and thus it is well-suited to the inherent characteristics of TKS. TO is computed as

$$TO(T_1, T_2) = \frac{1}{|C_1|} \sum_{c \in C_1} \frac{SC(c, T_1, T_2) \cap SC(c, T_2, T_1)}{SC(c, T_1, T_2) \cup SC(c, T_2, T_1)} \quad (6)$$

where

$$SC(c, T_1, T_2) := \left\{ c_j \in T_1 \cap T_2 | c_j \leq c \vee c \leq c_j \right\}$$

T_1 and T_2 are two different tree structures. The set of nodes in T_1 is denoted by C_1, and c is one of the nodes in C_1. $SC(c, T_1, T_2)$ represents the nodes, which both exist in T_1 and T_2, that are the ancestor or descendant of c. According to our preliminary experiment, TFIDF, power and sibling independence are all stable with regard to TO performance, but α is not. We thus conduct an experiment on α with a range from 0 to 1 using the data of ACM CCS I and ACM CCS B, as shown in Fig. 6.

Figure 6 demonstrates that the optimal α are 0.2 and 0.4 in the ACM CCS I and ACM CCS B, respectively. In addition, the results also show that the difference in TO performance is not obvious when α is under 0.8, although it falls significantly when $\alpha > 0.8$, which is reasonable, because the initial similarities are over influenced by large proportional abstractness. We follow the combination of optimal parameters determined empirically by a range test to investigate TO performance. Table 3 indicates that TO in ACM CCS I and B are 74.00% and 76.79%, respectively. The results represent the significant achievement of the proposed TKS with regard to the hierarchical feature within knowledge repositories, demonstrating that the reconstructed relations achieve high consistency compared to the gold standards. In order to evaluate the performance of abstractness, experiments without the similarity adjustment factor are also conducted. The results in Table 3 show that TO when $\alpha = 0$ in ACM CCS I and B are down to 65.99% and 65.35%, respectively. Hence, the higher TO when $\alpha = 0.2$ and $\alpha = 0.4$, comparing to $\alpha = 0$, shows the vital utility of abstractness in constructing more appropriate TKS. In addition, the results represent the analogous parameter settings for both datasets.

5.3 Evaluation of document classification

As mentioned above, the hierarchical relations in TKS are significant, as they make the data more comprehensible for users. Therefore, we assume that TKS also provides important information about the similarity of KO along with its formation. The appropriate similarities between terms are significant in representing documents as a set of vectors. The accurate weight of vectors provides substantial information

Table 4 The classification results for the Reuters corpus from TKS and WordNet

Category	Precision				Recall				F-measure			
	WordNet	TKS $\alpha=0$	TKS $\alpha=0.2$	TKS $\alpha=0.4$	WordNet	TKS $\alpha=0$	TKS $\alpha=0.2$	TKS $\alpha=0.4$	WordNet	TKS $\alpha=0$	TKS $\alpha=0.2$	TKS $\alpha=0.4$
Earn	**0.9997**	0.9990	0.9995	0.9995	**0.9997**	**0.9997**	**0.9997**	**0.9997**	**0.9997**	0.9993	0.9996	0.9996
Acq	**1**	**1**	**1**	**1**	0.8428	0.8632	0.8636	**0.8758**	0.8810	0.9033	0.9035	**0.9100**
Crude	0.8261	0.8417	0.8900	**0.9150**	0.4061	0.7330	**0.7488**	0.7461	0.4345	0.7079	0.7658	**0.7795**
Trade	0.7366	**0.8125**	0.7862	0.7849	**0.8022**	**0.8022**	**0.8022**	**0.8022**	0.7469	**0.8019**	0.7717	0.7699
Interest	0.5150	**0.5857**	0.5826	0.5826	**0.5156**	0.4625	0.4625	0.4625	**0.5151**	0.4930	0.4914	0.4914
Money-fx	**0.6925**	0.6050	0.6332	0.6312	**0.6890**	**0.6890**	**0.6890**	**0.6890**	**0.6859**	0.6328	0.6530	0.6518
Ship	0.6963	**0.7726**	0.7376	0.7376	**0.9958**	**0.9958**	**0.9958**	**0.9958**	0.7709	**0.8480**	0.8165	0.8165
Grain	0.8095	**0.8111**	**0.8111**	0.7825	0.8550	**0.8750**	**0.8750**	**0.8750**	0.7920	**0.8057**	**0.8057**	0.7890
Weighted avg	0.9377	0.9437	0.9449	**0.9459**	0.8837	0.9037	0.9046	**0.9080**	0.8862	0.9081	0.9095	**0.9118**

to enhance document classification performance. Hence, we transform the term-term similarities from TKS and compare them with those that are derived from semantic measurement using WordNet, a well known public lexical database for the English language [39]. The structure presented in this work achieves better performance, which indicates that the more accurate term-term similarities can be exhibited by its hierarchical relations. Moreover, the abstractness which is used to adjust TKS is also investigated to discover its effectiveness in document classification.

In a tree structure, the similarity between any two nodes can be intuitively derived according to their interval edge number. Specifically, two nodes are more similar when there are fewer edges between them, and vice versa. In this experiment, we adopt $e^{-\alpha l}$, a similarity measure representing a nonlinear transformation function of the shortest path length, to calculate term-term similarities from TKS [40]. The transformation function is in exponential form. Where l is the edge number of the shortest path between two nodes and α is a constant. The aforementioned data preprocessing techniques, namely tokenizing, stop word filtering and stemming, are applied to the Reuters dataset to reduce noise, overfitting and unnecessarily large feature vectors. For each category in the training set, the important term features are determined by calculating the TFIDF, and only terms with a high TFIDF are adopted in this experiment. We treat qualified terms as a set of KO in the Reuters dataset and used to construct the TKS. The similarity measure in TKS calculates term-term similarities according to the hierarchical structure, which was constructed using the parameter settings in Sect. 5.2. Moreover, we also test $\alpha = 0$, $\alpha = 0.2$ and $\alpha = 0.4$ to investigate the impact of similarity refinement. With regard to the traditional approach, the term-term similarities are checked from the WordNet thesaurus. It is worth noting that the initial term-term similarities which were used to construct TKS were also derived from WordNet.

The TKS and WordNet approaches can both show the term-term similarities for each category, C_j, using an $n * n$ matrix, $C_j = \begin{bmatrix} t^j_{1,1} \cdots t^j_{1,n} \\ \vdots \ddots \vdots \\ t^j_{n,1} \cdots t^j_{n,n} \end{bmatrix}$, where n is the number of features in the training set. Every document, d_i, in the testing set can also be represented by an n dimensional vector, $\bar{d}_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,n})$, which is composed of its TFIDF weights. Similar to the vector space model, the inner product of each category and a new testing document produce an n dimensional vector which is the n components of a testing document projected onto each category. Therefore, eight n dimensional vectors, $[w^j_{i,1}, w^j_{i,2}, \ldots, w^j_{i,n}]$, can be obtained for each new document, one for each category. In order to determine which category the new document belongs to, we add the n vector components up as an integrated index, $Category(d_i)$, to indicate which category should be assigned. The integrated index of document d_i is calculated by (7). The same procedures are applied to both TKS and WordNet.

$$Category(d_i) = \arg\max_j \left(M^j_i\right) \tag{7}$$

where

$$M^j_i = \sum_{k=1}^{n} w^j_{i,k}$$

and

$$C_j \bullet d_i = \begin{bmatrix} t^j_{1,1} & \cdots & t^j_{1,n} \\ \vdots & \ddots & \vdots \\ t^j_{n,1} & \cdots & t^j_{n,n} \end{bmatrix} \bullet \left[w_{i,1}, w_{i,2}, \ldots, w_{i,n}\right] = \left[w^j_{i,1}, w^j_{i,2}, \ldots, w^j_{i,n}\right]$$

The experiment performs 10-fold cross validation, and the results in Table 4 show that the TKS method achieves a

higher precision, recall and f-measure as compared to the WordNet approach. In addition, the results of $\alpha = 0.2$ and $\alpha = 0.4$ perform better than those of $\alpha = 0$ in the TKS method, which indicates that the similarity refinement is able to provide more precise relations of KOs, although the impact is slight in this experiment. The results in Table 4 also illustrate that the initial WordNet similarities can be adjusted to more appropriate ones through TKS transformation. It is worth noting that compared with other document classification methods, for example, VSM and Support Vector Machine (SVM), the proposed TKS can provide visualized domain knowledge to users by demonstrating the specific key term arrangement rather than by utilizing complex numerical computing. As a result, in KM applications, TKS can be used to enhance knowledge comprehension by discovering the appropriate hierarchical relations from a flat set of KOs.

6 Conclusions and future work

In the current era of information overload, knowledge representation techniques are important to aid comprehension of tacit knowledge. In previous research, the problem of multiple inheritance and the absence of hierarchical features are regarded as the main weaknesses in providing clear and comprehensive KS. In order to overcome these problems, we propose a methodology to construct a TKS which is single inherited and provides a layered view to enhance the navigation of knowledge. To demonstrate the effectiveness of the proposed methodology, two experiments are conducted to evaluate its consistency with gold standards and verify the similarity measure provided by TKS in the document classification task. The results of the experiment demonstrate the impressive consistency of TKS and the ACM CCS gold standards. Moreover, the results in document classification also show that the hierarchical information embedded in TKS can successfully aid the discovery of appropriate relations among KOs. The better performance when $\alpha > 0$ in both experiments also demonstrates that the similarity refinement method is able to provide effective similarity revision for TKS construction.

Future research should consider exploring more effective similarity refinement methods trying other similarity measures to obtain a mutual similarity matrix, and achieve more consistent performance with the gold standards. In addition, how to automatically construct TKS with the most appropriate root or with multiple appropriate roots remains a considerable challenge in this research. Although the parameters are very stable in our preliminary tests, experiments with a wider range values should be conducted and their impacts investigated in more detail. Moreover, experiments with more gold standards are also required to validate the robustness of the universal parameter setting. It is worth noting that, theoretically, the TCM construction algorithm can be applied to different language system. But the various text preprocessing techniques in knowledge codification stage should be modified along with the language systems.

References

1. Bradley JH, Paul R, Seeman E (2006) Analyzingthe structure of expert knowledge. Inf Manag 43:77–91
2. Chen NS, Kinshuk Wei CW, Chen HJ (2008) Mining e-learning domain concept map from academic articles. Comput Educ 50:1009–1021
3. Tseng SS, Sue PC, Su JM, Weng JF, Tsai WN (2007) A new approach for constructing the concept map. Comput Educ 49:691–707
4. Eppler MJ, RA Burkhard (2007) Visual representations in knowledge management: framework and cases. J Knowl Manag 11:112–122
5. Murphy GL, Lassaline ME (1997) Hierarchical structure in concepts and the basic level of categorization. In: Lamberts K, Shanks D (eds) Knowledge, concepts and categories. MIT Press, Cambridge
6. Stumme G, Taouil R, Bastide Y, Pasquier N, Lakhal L (2002) Computing iceberg concept lattices with titanic. Data Knowl Eng 42:189–222
7. Rajapakse RK, Denham M (2006) Text retrieval with more realistic concept matching and reinforcement learning. Inform Process Manag 42:1260–1275
8. Belohlavek R, Dvorak J, Outrata J (2007) Fast factorization by similarity in formal concept analysis of data with fuzzy attributes. J Comput Syst Sci 73:1012–1022
9. Sparrow J (1998) Knowledge in organizations: access to thinking at work. Sage, London
10. Novak JD (1993) How do we learn our lesson? Taking students through the process. Sci Teach 60:50–55
11. Ruiz-Primo MA, Schultz SE, Li M, Shavelson RJ (2001) Comparison of the reliability and validity of sores from two concept-mapping techniques. J Res Sci Teach 38:260–278
12. Wang J (2003) A knowledge network constructed by integrating classification, thesaurus, and metadata in digital library. Int Inf Libr Rev 35:383–397
13. Xu JJ, Chen H (2005) Crimenet explorer: a framework for criminal network knowledge discovery. ACM Trans Inform Syst 23:201–226
14. Schvaneveldt RW (1990) Pathfinder associative networks: studies in organization. Albex Publishing, Norwood
15. Chen RC, Liang JY, Pan RH (2008) Using recursive art network to construction domain ontology based on term frequency and inverse document frequency. Expert Syst Appl 34:488–501
16. Fenza G, Loia V, Senatore S (2008) A hybrid approach to semantic web services matchmaking. Int J Approx Reason 48:808–828
17. Lee CS, Kao YF, Kuo YH, Wang MH (2007) Automated ontology construction for unstructured text documents. Data Knowl Eng 60:547–566
18. Reformat M, Ly C (2009) Ontological approach to development of computing with words based systems. Int J Approx Reason 50:72–91
19. Lee CS, Jian ZW, Huang LK (2005) A fuzzy ontology and its application to news summarization. IEEE Trans Syst Man Cybern, Part B, Cybern 35:859–880
20. Stumme G (2003) Off to new shores: conceptual knowledge discovery and processing. Int J Human-Comput Stud 59:287–325

21. Priss U (2006) Formal concept analysis in information science. In: Cronin B (ed) Annual review of information science and technology (arist), vol 40. Information Today Medford, New Jersey, pp 521–543
22. Ganter B, Wille R (1999) Formal concept analysis: mathematical foundations. Springer, New York
23. Tho QT, Hui SC, Fong, Cao TH (2006) Automatic fuzzy ontology generation for semantic web. IEEE Trans Knowl Data Eng 18:842–856
24. Chi YL (2007) Elicitation synergy of extracting conceptual tags and hierarchies. Expert Syst Appl 32:349–357
25. Formica A, Missikoff M (2004) Inheritance processing and conflicts in structural generalization hierarchies. ACM Comput Surv 36:263–290
26. Carpineto C, Romano G (2004) Exploiting the potential of concept lattices for information retrieval with credo. J Univ Comput Sci 10:985–1013
27. Everitt B (1993) Cluster analysis. Edward Arnold, London
28. Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. Paper presented at the KDD Workshop on Text Mining, Boston, MA, USA
29. Treeratpituk P, Callan J (2006) Automatically labeling hierarchical clusters. Paper presented at the proceedings of the 6th national conference on digital government research, San Diego, USA
30. Chung W, Chen H, Nunamaker JF Jr (2005) A visual framework for knowledge discovery on the web: an empirical study of business intelligence exploration. J Manag Inf Syst 21:57–84
31. Cimiano P, Hotho A, Staab S (2005) Learning concept hierarchies from text corpora using formal concept analysis. J Artif Intell Res 24:305–339
32. Lammari N, Metais E (2004) Building and maintaining ontologies: a set of algorithms. Data Knowl Eng 48:155–176
33. Sanderson M, Lawrie D (2000) Build, testing and applying concept hierarchies. In: Advances in information retrieval: recent research from the center for intelligent information retrieval, vol 7. Springer, New York, pp 235–266
34. Yates RB, Neto BR (1999) Modern information retrieval. ACM Press, New York
35. Glover E Pennock DM, Lawrence S, Krovetz R (2002) Inferring hierarchical descriptions. In: Proceedings of the 20th international conference on information and knowledge management (CIKM), McLean, Virginia, pp 507–514
36. Wu HC, Luk RWP, Wong KF, Kwok KL (2008) Interpreting TF-IDF term weights as making relevance decisions. ACM Trans Inf Syst 26:1–37
37. Rodríguez MA, Egenhofer MJ (2003) Determining semantic similarity among entity classes from different ontologies. IEEE Trans Knowl Data Eng 15:442–456
38. Tang J, Li J, Liang B, Huang X, Li Y, Wang K (2006) Using bayesian decision for ontology mapping. Web Semantics: Science, Services and Agents on the World Wide Web 4:243–262
39. Fellbaum C (1998) Wordnet: an electronic lexical database. MIT Press, Cambridge
40. Li Y, Bandar ZA, McLean D (2003) An approach for measuring semantic similarity between words using multiple information source. IEEE Trans Knowl Data Eng 15:871–882