Knowledge map-based method for domain knowledge browsing

Jia Hao *,1, Yan Yan, Lin Gong, Guoxin Wang, Jianjun Lin

School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China
Department of Mechanical Engineering, University of Michigan, MI, USA

ARTICLE INFO

Article history:
Received 8 June 2013
Received in revised form 24 December 2013
Accepted 6 February 2014
Available online 15 February 2014

Keywords:
Information overload
Knowledge browsing
Knowledge map
Social network analysis

ABSTRACT

The exponential growth of available information and the deployment of knowledge management systems delivers excessive information to the end users that they cannot manage at once. This problem has led to an increased emphasis on solutions to information overload. Searching and browsing are two methods to locate information. Many studies have focused on solving the information overload problem in the searching process, but the methods to alleviate information overload in browsing process have not been adequately studied. Hence, a method that addresses information overload in the browsing process is presented in this paper. The aim is to reduce the information overload during browsing domain knowledge for new knowledge users who have little understanding of the information. In this method, a knowledge map and social network analysis are utilized to navigate the knowledge users. Technologies first construct a knowledge map from text mining and the important knowledge that includes more information about the domain is then identified via social network analysis. Based on this process, the knowledge user can browse the domain knowledge starting from the important knowledge and navigate via the knowledge map. We applied the method to assist new knowledge users in browsing the Computer Numerical Control (CNC) domain knowledge base to validate the method. The results indicate that the method can identify the important knowledge at a highly acceptable level, the constructed knowledge map can efficiently navigate the knowledge users, and the information overload can be significantly decreased.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Domain knowledge has become increasingly critical to develop competitive mechanical products [1]. In domain knowledge management systems, knowledge users usually employ one of two strategies to obtain domain knowledge: searching and browsing [2,3]. In the searching strategy, users have a topic or some keywords in mind, while they do not search for a specific thing in the browsing strategy, which is characterized by the absence of planning. Although the users can obtain information or domain knowledge via the two strategies, the information overload phenomenon is becoming increasingly frequent during the information acquisition process due to the exponentially increasing amount of information in the knowledge base. Information overload indicates that more information is provided than users can process in a period of time [4–7], which makes them spend more time on information acquisition. Therefore, new methods and tools need to be developed to assist the users in addressing information overload.

* Corresponding author at: 3635 Green Brier Blvd, Apt 120A, Ann Arbor, MI 48105–2011, USA. Tel.: +1 734 707 9263.
E-mail addresses: haojia632@gmail.com (J. Hao), yanyan331@bit.edu.cn (Y. Yan),
gonglin@bit.edu.cn (L. Gong), wangguoxin@bit.edu.cn (G. Wang), bitlin@sina.cn (J. Lin).
1 5 South Zhongguancun Street, Haidian District, Beijing 100081, China.

http://dx.doi.org/10.1016/j.dss.2014.02.001
0167-9236/© 2014 Elsevier B.V. All rights reserved.
2. Background

Several studies have been conducted to decrease the information overload that users are facing. Many studies have aimed to ease the information overload when obtaining information via the searching strategy. A search engine is the main tool to implement a searching strategy. Depending on the techniques employed, search engines can be classified into hierarchical search engines, agent-based search engines and Meta search engines [17]. To improve the search engine, several methods have incorporated the interest profile of subjects into the search engine [6], adopted a new result display method to replace the one-dimensional result list [6,7], provided a query-aided application to the subjects to help them compose important query sentences [8] and processed the information via social tags [9], such as Yahoo! Delicious.

However, this same problem has not been adequately researched for the browsing strategy. In our opinion, information overload in the browsing strategy is a more serious problem than in the searching strategy, because the search engine can remove a large amount of unrelated domain knowledge when searching knowledge, while this removal is not possible when browsing the knowledge base. Conversely, the interest information of knowledge users is helpful to refine the searching result. However, the knowledge users may lack a specific vocabulary to represent their interest when they browse information [3]. Therefore, knowledge users can easily become lost when browsing strategy. The site map and navigation page seem to be common methods to ease the information overload in the browsing strategy. However, the site maps generally cover only the main structure of information, which is insufficient to alleviate the information overload. To this end, the hierarchical subject category (like Yahoo) has been incorporated to guide the browsing process [2]. The recommendation system has also been adopted to alleviate the information overload; collaborative filtering is most likely the most widely implemented technique [10]. Other recommendations systems, including content-based recommenders [11], utility-based recommenders [12] and demographic recommenders [13], have also been widely studied. The obvious limitation of the above methods is the effort requirement of collecting extra information to implement the recommendation. Information visualization is another type of method that is used to alleviate the information overload; Chen [15] indicates that a Kohonen self-organizing map (SOM) -based method can categorize a large information space into manageable sub-spaces that the user can easily browse. Therefore, Yang [3] suggests that visual overload may occur when the size of visualization increases. The visual overload means knowledge users cannot easily locate the information from a large pool of information if the information is packed very densely in the visualization map. They proposed a fisheye view and fractal view to support information visualization. The evaluation result shows that the method increases the effectiveness of information visualization. See reference [14] for the information visualization method. The information visualization is indeed useful to alleviate the information overload. However, as stated by Yang [3] visual overload will emerge as the amount of information increases. Different from visualization methods, Wan [5] developed a new research tool called Citation-Sensitive In-Browser Summarizer (CSIBS) to help literature browsing tasks. This method alleviates the information overload by presenting information about a citation when the knowledge user encounters it, and this information can help the reader determine whether to invest the time to explore the citation. Kruk [16] presented the MultiBeeBrowser (MBB), which is a faceted navigation tool for content browsing. Faceted navigation means that the information space is partitioned using orthogonal conceptual dimensions of the data. These dimensions are called facets and represent important patterns of the information elements. Wang [17] presented a navigation graph-based recommendation system in which the browsing track information of previous knowledge users is utilized to provide the contents to new knowledge users.

In this paper, we propose a method to alleviate the information overload in design knowledge browsing. This method provides the knowledge users with a small number of important knowledge items at the beginning of browsing the knowledge base, and all other knowledge can be found very quickly starting from the important knowledge. The important knowledge will be defined and calculated in the following sections. The method incorporates a knowledge map [18] and a social network analysis (SNA) [19,20]. The knowledge map will be constructed via latent semantic analysis (LSA) [21], and the important knowledge will be defined and identified using SNA algorithms. The method includes two main stages. The first stage takes a domain knowledge base as an input and gives the corresponding domain knowledge map as the output, and the domain knowledge base under consideration is a natural (Chinese) language document database. The knowledge map is represented by a mathematical graph in which vertices indicate the domain knowledge and edges indicate the semantic similarity between domain knowledge. Technologies determine the semantic similarities from text-mining. The second stage identifies the important knowledge from the constructed knowledge map. Inspired by social network analysis (SNA), we used a graph-theory to define and calculate important knowledge. Our approach differs from others because the only required input is the content of the domain knowledge itself. This character is suitable for domain knowledge browsing, because it is unstructured and lacks connections.

3. Proposed method

Our method aims to alleviate the information overload when browsing domain knowledge base. Fig. 1 shows an overall illustration of the entire method.

As shown in Fig. 1, the method includes two steps: Knowledge Map Construction (KMC) and Important Knowledge Identification (IKI). The domain knowledge map can be determined from its corresponding domain knowledge base in the KMC step. Based on this map, the important knowledge can be identified from the constructed knowledge map in the IKI step. The proposed method contains three data objects: domain knowledge base, domain knowledge map and important knowledge, as shown in Fig. 1. Knowledge users can more easily browse the knowledge base with these three data objects. At the beginning of browsing the knowledge base, the important knowledge is presented to the knowledge users and the domain knowledge map is then used to navigate the knowledge users to the relevant domain knowledge. The knowledge base can provide the detail of the domain knowledge. The following two sections explain the two steps in detail.

4. Knowledge map construction

The goal of the knowledge map construction is to convert a knowledge base into a corresponding knowledge map. We first illustrate the meaning of the knowledge map in our work and then detail the three steps of constructing the domain knowledge map.

4.1. Domain knowledge map

The knowledge map is a knowledge representation technology that reveals the underlying relationships between knowledge sources [22]. It can be used to find sources of knowledge, implement knowledge creation and increase knowledge sharing [23]. Furthermore, the knowledge map is helpful to many fields, such as information visualization [15], information retrieval [6], strategic decision-making support [23] and business process re-engineering [25]. However, few studies have focused on the method to build a knowledge map [23,24], likely because the definitions and uses vary by situations. In this work, we treat the knowledge map as a mathematical graph and provide the construction method. In mathematics, a graph is a representation of a set of objects in which some pairs of the objects are connected by links. The interconnected objects are represented by mathematical abstractions called.
vertices, and the links that connect some pairs of vertices are called edges [26]. In a knowledge map, the vertices represent domain knowledge and the edges represent semantic similarities between domain knowledge. Generally, a graph is described in diagrammatic form as a set of dots for the vertices that are connected by lines or curves for the edges. Fig. 2 shows an example of a knowledge map.

The knowledge map shown in Fig. 2 includes ten domain knowledge items and some of them are semantically similar to each other as indicated by the lines. The key issue of constructing such a knowledge map is to determine the similarities between all domain knowledge and then determine which similarities should be used to build the knowledge map.

4.2. Domain knowledge representation

The knowledge base under consideration in this work is a natural language document database. In this step, the documents in the knowledge base will be converted into a computable form. The input of this step is a domain knowledge base and the output is the corresponding keyword-by-knowledge item matrix (co-occurrence matrix) whose rows represent keywords and columns represent knowledge items. We used a Vector Space Model (VSM) to represent the domain knowledge. The VSM is an algebraic model for representing text documents as vectors of identifiers. The definition of identifiers depends on the application; in this work, the identifiers are the keywords extracted from the domain knowledge base using the text-mining method. After this step, each knowledge item in the domain knowledge base is represented by Eq. (1).

\[ k = \{(t_1, w_1), (t_2, w_2), \ldots, (t_M, w_M)\} \]  

where \( t_i \) denotes the \( i \)th keyword extracted from the domain knowledge base; \( w_i \) denotes the weight of \( t_i \) and the value is calculated by the term frequency–inverse document frequency (TF–IDF) as shown in Eq. (2); \( M \) is the total number of keywords extracted from the domain knowledge base.

\[ w = \frac{f(t,k)}{\max \{f(t,k) : t \in k\}} \times \log \frac{|K|}{|\{k \in K : t \in k\}|} \]  

where \( f(t,k) \) denotes the co-occurrence times of the keyword \( t \) in the domain knowledge item \( k \); \( K \) denotes the domain knowledge base. The TF–IDF is a numerical statistic that reflects how important a keyword is to a knowledge item in the knowledge base. The TF–IDF value increases proportionally to the number of times a keyword appears in the knowledge item, but is offset by the frequency of the keyword in the knowledge base, which helps to control for the fact that some keywords are generally more common than others.

4.3. Semantic similarity calculation

The semantic similarities between domain knowledge items are determined in this step based on the co-occurrence matrix calculated in the previous step. At the end of this step, a similarity matrix is determined. The similarity matrix is a knowledge item-by-knowledge item matrix whose entries are symmetric with respect to the main diagonal. The elements of the similarity matrix indicate the content relevance between the domain knowledge and both the columns and rows of the similarity matrix represent the domain knowledge items.

In this work, we used Latent Semantic Analysis (LSA) to calculate the similarity between domain knowledge. LSA was proposed in 1988 in a patent and is a technique in vectorial semantics that analyzes the relationships between a set of documents and the keywords they contain by producing a set of concepts related to the documents and keywords. We adopted LSA because (1) the original co-occurrence matrix was presumed too large for the computing resources and (2) the keywords in the original co-occurrence matrix are not independent from each other in the real world. LSA can mitigate the problems by finding a low-rank approximation to the original co-occurrence matrix.

LSA uses singular value decomposition (SVD) to obtain a low-rank approximation of the original co-occurrence matrix. According to SVD, the original co-occurrence matrix can be decomposed by Eq. (3).

\[ S = U \Sigma V^T \]  

where \( S \) represents the original co-occurrence matrix, \( U \) and \( V \) are orthogonal matrices and \( \Sigma \) is a diagonal matrix. The values in \( \Sigma \) are called singular values. The column vectors of \( U \) are the left singular vectors while the column vectors of \( V \) are the right singular vectors. When the \( k \) largest
singular values and their corresponding singular vectors are selected, we will obtain the rank k approximation to S with the smallest error from Eq. (4).

\[ S_k = U_k \sum_{i=1}^{k} \sigma_i V_i^T \]  

(4)

If we define \( \sigma_1 > \sigma_2 > \cdots > \sigma_N \) as the singular values, then Eq. (5) is used to determine the value of k. \( \tau \) is the information retention rate, which controls the amount of information from the original co-occurrence matrix that is retained in the low-rank approximation.

\[ \sum_{i=1}^{k} \sigma_i / \sum_{i=1}^{N} \sigma_i > \tau \]  

(5)

After the calculation, we can treat the k-rank approximation matrix \( S_k \) as a "semantic space". Based on the semantic space, we calculate the semantic similarity between the domain knowledge simply by the cosine of the angle between the vectors of domain knowledge.

4.4. Knowledge map building

The input of this step is the similarity matrix that was determined in the previous step, and the output will be a determined knowledge map. The main task consists of determining the similarities that remain and those that are deleted. Intuitively, we think that all domain knowledge in the specific domain is gathered into different sub-areas and some connections exist between them. Fig. 3 gives an example of the expected domain knowledge map. The knowledge map includes fourteen domain knowledge items that are gathered into three sub-areas.

We first analyzed all the similarities to determine the ones that should be deleted. As shown in Fig. 4, we divided all similarities into four groups represented by S, M, L1, and L2. The + sign indicates that the corresponding group of similarity should be retained in the final knowledge map while the − sign indicates that the corresponding group of similarity should be removed. The S group includes the smallest similarities that represent some pairs of domain knowledge that are completely unrelated. This group of similarity should be removed, similar to the similarity between \( K_4 \) and \( K_5 \) in Fig. 3. The M group includes the medium similarities that are regarded as the common similarities between domain knowledge. The knowledge map contains common similarities, because knowledge in the same domain necessarily shares a batch of keywords in its content. However, these common similarities in a domain knowledge map will conceal the connection characteristics of the domain knowledge map. Therefore, this group should also be deleted. The L1 group constitutes the biggest group of similarities that are presumed to represent some pairs of domain knowledge that are close related, especially the domain knowledge within a sub-area. This group of similarities should be retained in the final knowledge map, similar to the connection between \( K_4 \) and \( K_9 \) in Fig. 3. The L2 group represents the connections between the sub-areas, similar to the connection between \( K_7 \) and \( K_4 \) in Fig. 3. This type of similarity should be important for the connectedness, because it acts as the bridge in the final knowledge map. Therefore, the L2 group of similarities should also be retained in the final knowledge map.

Based on the previous analysis of the similarities, we selected a simple policy to build the final knowledge map. The policy consists of adding the biggest similarity continuously to the knowledge map until the final knowledge map connects. In a connected graph, any other vertices can be accessed from one vertex. This simple policy can build the connections within the sub areas and also bridge different sub-areas by retaining the relatively bigger similarities. Therefore, we presume that this policy will help us to form a domain knowledge map in accordance with our intuitive understanding.

5. Important knowledge identification

The domain knowledge map that was determined in the previous section can navigate knowledge users to browse the domain knowledge base. However, the knowledge users still do not know where to begin browsing the knowledge base. The goal of this step is to determine the relatively important domain knowledge in the knowledge map, so that the knowledge users can start from the important knowledge. The input is the constructed domain knowledge map and the output will be a set of important domain knowledge items.

5.1. Definition of important knowledge

In this section, the domain knowledge map is treated as mathematical graph. The degree of a vertex in a mathematical graph is the number of edges incident to the vertex. This metric provides the possibility to define the important knowledge. As shown in Fig. 2, the degree of \( K_4 \) is 4, and the degree of \( K_9 \) is 3. The degrees of vertices \( K_4 \) and \( K_9 \) are larger than those of the other vertices. Therefore, these two vertices are much more important than the others. If the degree of a vertex is high in a knowledge map, we hold that its corresponding domain knowledge carries more information about the domain than the domain knowledge items corresponding to low degree vertices. Therefore, we can define the domain knowledge that corresponds to a higher degree as the important knowledge of the domain knowledge map.

5.2. Existence of important knowledge

This step aims to identify the presence of important knowledge in the constructed knowledge map. This step is necessary because the knowledge maps do not necessarily or clearly contain important knowledge. For example, we consider that a knowledge map in which most of the vertices hold the same or similar degree lacks important knowledge. The aim is to determine whether the domain knowledge map shares the same structure with the example knowledge map shown in Fig. 3. Therefore, we require a method to examine the internal structure of the knowledge map.
The small-world network and scale-free network are helpful to investigate the internal structure of the domain knowledge map. If a domain knowledge map is conjectured to be a small-world network, we can know that the domain knowledge is gathered into different groups. If a domain knowledge map is inferred as a scale-free network, we can know that it contains a small number of high degree vertices. If a domain knowledge map is small-world network as well as a scale-free network, we can infer that the knowledge map shares its structure with the example knowledge map shown in Fig. 3.

A small-world network [27] is a type of mathematical graph in which most nodes are not neighbors of one another, but most nodes can be reached from every other node via a small number of steps. The small-world network is characterized by a high clustering coefficient and small mean-shortest path. The clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. The value can be determined with by Eq. (6). The mean-shortest path length is the average distance of all vertices in a graph. The distance between two vertices is the number of edges in a shortest path connecting them. This value can be determined with Eq. (8).

\[ C = \frac{1}{N} \sum_{i=1}^{N} c_i \]  

(6)

where C denotes the clustering coefficient of the entire graph; N is the number of vertices in the graph; \( c_i \) is the clustering coefficient of vertex \( i \), which can be calculated with Eq. (7).

\[ c_i = \frac{2 \cdot e_i}{d_i(d_i - 1)} \]  

(7)

where \( e_i \) is the number of connections among the neighbors of vertex \( i \); \( d_i \) denotes the degree of vertex \( i \).

\[ L = \frac{2}{N(N-1)} \sum_{i,j=1, i \neq j} l_{ij} \]  

(8)

where \( l_{ij} \) denotes the shortest distance between vertex \( i \) and vertex \( j \).

We can compare the C and L of the domain knowledge map with its corresponding regular graph and random graph to confirm whether the domain knowledge map is a small-world graph [28]. The corresponding regular graph, random graph and the knowledge map contain the same number of vertices and edges. The number of neighbors of each vertex is identical in a regular graph, which is characterized by a high clustering coefficient and high mean-shortest path. Random graphs are generated by a random process and characterized by a low clustering coefficient and low mean-shortest path length. Eq. (9) [28] shows the equations used to calculate the C and L of a regular graph and random graph.

\[ C_{\text{random}} \sim \frac{D}{N} \ln N \]  

\[ C_{\text{regular}} \sim \frac{3(D-2)}{4(D-1)} \frac{N}{2D} \]  

\[ L_{\text{average}} \sim \ln D \]  

\[ L_{\text{random}} \sim \ln N \frac{N}{2D} \]  

\[ L_{\text{regular}} \sim \frac{N}{2D} \]  

(9)

where D represents the average degree of all vertices and N represents the number of vertices in the network.

A scale-free network [27] is a network whose degree distribution follows a power law, i.e., the fraction of vertices in the network having \( k \) connections to other vertices \( P(k) \) satisfies \( P(k) \sim k^{-\beta} \), where \( \beta \) is a parameter. This type of graph contains certain high degree vertices that greatly exceed the average. These vertices are called hubs. To determine whether the knowledge map is scale-free, we can generate the degree distribution of the knowledge map and verify it using the curve-fitting method.

After this step, we can proceed to calculate the important knowledge if the knowledge map is a small-world network and scale-free network. Otherwise, the knowledge map should be rebuilt by a different policy.

5.3. Calculation of important knowledge

If the knowledge map is presumed to have important knowledge, we can identify a set of domain knowledge items in this section. Based on the definition of important knowledge, we can simply select vertices with the highest degree as the important knowledge. However, this approach is not sufficient for the end users, because the systems are not aware of how many important knowledge items should be provided. Therefore, we should determine the total number of the important knowledge items (\( N_b \)). This value is required to develop the assistant tools for end users. The value of \( N_b \) should remain small because our goal is to alleviate the information overload. Therefore, this section aims to determine the total number of the important knowledge items. To this end, we adopted a new metric called graph efficiency [21], which is defined by Eq. (10).

\[ E = \frac{1}{N(N-1)} \sum_{i,j} \frac{1}{d_{ij}} \]  

(10)

where \( E \) denotes the efficiency of the graph and \( d_{ij} \) is the shortest distance between vertex \( i \) and vertex \( j \). Specially, if vertices \( i \) and \( j \) lack a connection, the efficiency between the two vertices is zero. In a domain knowledge map, this metrics indicates the average number of steps required to navigate the knowledge users from one domain knowledge item to others. A low graph efficiency indicates more steps are required to reach a knowledge item. To calculate the number of important knowledge items, we can sort all vertices in the graph by their degree and then select the top n vertices as the important knowledge. The removal of the top n vertices from the graph decreases \( E \) to a small value, \( E^* \), and we define the efficiency reduction threshold (ER) as \( (E - E^*)/E \). Therefore, we can determine the number of important knowledge items and extract a set of important knowledge items by selecting a proper ER.

6. Case study

6.1. The dataset

Public datasets for domain knowledge map construction are currently unavailable. Therefore, we obtain 844 documents related to computer numeric control (CNC) from the CNC laboratory in our college. These documents include published papers, patents and expert experiences that were selected from different sub-areas including CNC programming, CNC fault, CNC machine, CNC technology, CNC testing, CNC system and servo motor. Table 1 shows the number of documents of each sub-area. The published papers and patents were retrieved and downloaded from the China National Knowledge Infrastructure (CNKI: http://www.cnki.net/), and the expert experiences were collected from industry practice. The CNC laboratory in our college provides these documents to newcomers to allow them to access and understand the published research as soon as possible. However, users cannot easily browse the content without an assistance tool. Therefore, we processed the collected knowledge using the method in this paper.

6.2. Knowledge map construction

We first built a database that includes the title, sub-area and content of all collected knowledge. We then used IKAnalyzer [29], which is a text segmentation tool written in Java, to analyze the database, and develop an 844 by 153 co-occurrence matrix. We first extracted a set of
keywords using the text segmentation method to obtain the 153 keywords. We then found that too many keywords are generated, some of which were meaningless with respect to representing the content of the domain knowledge. Therefore, we asked a group of experts who are familiar with the domain under consideration to filter the keywords. Finally, a total of 153 keywords were obtained. Based on the co-occurrence matrix, we calculated the similarity matrix via LSA and $\tau$ was set to 85%. The impact of different $\tau$ will be analyzed later. According to the similarity removing policy introduced in Section 4.4, we obtained a graph that included 844 vertices and 5762 edges.

### 6.3. Important knowledge identification

The internal structure of the graph was analyzed with the method described in the previous section. Fig. 5 shows the small world graph analysis result. The left figure shows that $C$ of the knowledge map and regular graph were larger (0.73 and 0.72) than that of the random graph (0.03). The right figure indicates that the $L$ of the knowledge map and random graph were smaller (1.99 and 2.04) than that of regular graph (15.45). The analysis proves that the constructed knowledge map is a small-world graph. The result implies that domain knowledge is gathered into different groups, which agrees with the fact that domain knowledge is selected from different sub areas.

Fig. 6 shows the scale-free network analysis result. The blue circles represent the degree distribution of the knowledge map. The result indicates a clear power law scaling decay as $P(k) \sim k^{-\beta}$. The value of $\beta$ in Fig. 6 is 0.79, and the value of $R$ is 0.97, which is a statistic that provides information about the goodness of curve fitting. We found that the value of $\beta$ is very low compared to the real world scale-free networks, such as the World Wide Web ($\beta = 2.1$) [30], the Internet ($\beta = 2.5$) [31], metabolic networks ($\beta = 2.2$) [32], and the movie actor network ($\beta = 2.3$) [33]. However, this value is much closer to the knowledge map constructed by Jun Liu [27], in which the value of $\beta$ ranges from 0.79 to 1.69. This result suggests that a small number of knowledge items in the knowledge map have more access to other knowledge items.

The analysis clearly indicates that the internal structure of the knowledge map and the example knowledge map shown in Fig. 3 are similar. Therefore, we can infer that the knowledge map contains important knowledge. Fig. 7 shows that the graph efficiency changes when vertices are removed from the graph according to their degree. The x-axis indicates the number of removed vertices, and the y-axis indicates the efficiency of the graph. The red stars denote the efficiency when randomly removing vertices, while the blue dots denote the efficiency when removing vertices by degree. We can clearly see that the efficiency reduces rapidly as the high degree vertices are removed.

Table 2 shows the fraction of efficiency reduction when the vertices are removed. We selected the number of important knowledge items when the efficiency reduction exceeded 90%. In this case, the total number of the important knowledge items is 10. Table 3 lists the title and their sub-area of the important knowledge; the content was originally in Chinese and we translated it into English.

#### 7. Discussion

##### 7.1. Result validation

The result was validated based on three aspects, including the important knowledge, the knowledge map and the utility of the proposed approach. We used four metrics to measure the result: the acceptable level (AL), comprehensiveness (S), efficiency (Ef) and information overload reduction (IOR). The first two were used to measure the important knowledge, while the third was used to measure the knowledge map and the last could be used to measure the utility of the proposed approach with respect to decrease the information overload.

The AL can measure the acceptability of the important knowledge, and a group of domain experts were invited to calculate this metric. S measures how many sub-areas of the domain are covered by the important knowledge. A high value of $S$ implies that the important knowledge covers many sub-areas of the domain, which implies that the result is good. Ef can be calculated using Eq. (10); it measures how many steps are needed on average to move from one knowledge item to any other in the knowledge map. We adopted this metric because it reveals whether a knowledge user can be navigated conveniently in the knowledge map. The IOR measures the utility of the proposed approach. Because our goal is to decrease the information overload, this metric can measure how much information overload is eliminated by the proposed approach.

---

**Table 1**

Number of knowledge items in each sub area.

<table>
<thead>
<tr>
<th>Sub area</th>
<th>CNC programming</th>
<th>CNC fault</th>
<th>CNC machine</th>
<th>CNC technology</th>
<th>CNC testing</th>
<th>CNC system</th>
<th>Servo motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>114</td>
<td>106</td>
<td>211</td>
<td>135</td>
<td>97</td>
<td>79</td>
<td>102</td>
</tr>
</tbody>
</table>

---

[Fig. 5. Internal structure analysis (small-world).]
To obtain the AL of the important knowledge, we invited four groups of domain experts. Each group included two masters and one doctor. Their research areas closely relate to CNC, and they understand the content in the knowledge base because they were asked to read these documents when they came to the laboratory. They were asked to evaluate whether the calculated important knowledge can serve as the important knowledge of the domain. After the evaluation process, we obtained the data shown in Table 4.

In Table 4, the M represents the master and the D represents the doctor. The values are the scores that the experts gave to each domain knowledge item. The values ranged from 1 to 5, and the high values indicate the knowledge was more acceptable to be selected as the important knowledge. Based on the table, we can compute the acceptable level using Eq. (11).

\[
AL = \frac{\sum_{i=1}^{G} M_{ik} + M_{ik} + 2D_{ik}}{T_m + 2T_d} / 5N \cdot G
\]  

where AL represents the acceptable level; G indicates the number of groups to evaluate the result; Tm stands for the total number of masters in each group, and Td represents the total number of doctors in each group; N indicates the total number of important knowledge items. In this case, the acceptable level was 76.9%.

The value of S is much easier to calculate, and it also ranges from 0 to 1. This parameter is the ratio of the number of sub-areas covered by important knowledge to the total number of sub-areas. In this case, the comprehensiveness value is 5/7 = 71.4%. Ef can be calculated with Eq. (10), and its value also ranges from 0 to 1. In this case, the value is 0.51, which indicates only two steps are needed on average to move from one knowledge item to another.

In this work, we regard the total number of knowledge items that are provided to the knowledge user at any one time as the information overload. Although this view ignores the different characteristics of different knowledge users, measuring the reduction of information overload is meaningful. Based on this idea, we calculated the IOR with Eq. (12).

\[
IOR = \frac{N - N_{\text{Processed}}}{N}
\]  

where N indicates the total number of domain knowledge, which is regarded as the initial information overload, Nprocessed stand for the total number of domain knowledge items that the knowledge user may be provided using the proposed approach. We assumed that the value is the sum of the total number of important knowledge and the average number of domain knowledge items which can be directly navigated to from the important knowledge. Therefore, Eq. (12) can be changed to Eq. (13). In this case the value of IOR was 88.93%, which indicates the information overload significantly decreased.

\[
IOR = \frac{N - \left( \frac{N_{ik} + N - N_{ik}}{N_{ik}} \right)}{N} = 1 - \frac{1}{N_{ik}} \frac{1}{N} \frac{N - N_{ik}}{N_{ik}}
\]

Table 5 shows the value of the four above-mentioned metrics. The validation result provides some information about the goodness of the proposed approach, which yields the following conclusions about the data in Table 5.

First, the AL indicates that most of the domain experts tend to accept the important knowledge; if we remove the domain knowledge that most experts give a low score, such as for domain knowledge #227 shown in Table 4, the AL value was significantly improved. S indicates that most of the sub-areas are covered by the important knowledge, which makes the important knowledge more useful to navigate the knowledge users because the increased amount of comprehensive important knowledge increases the possibility that the knowledge users access knowledge in all sub-areas. The value of AL and S suggest that the calculated important knowledge can be used to navigate the knowledge users.

Second, Ef indicates that the domain knowledge items in the constructed knowledge map are closely connected, and an average of approximately 2 steps is required to move from one domain knowledge item to another. The high Ef value allows the knowledge users to jump from one domain knowledge item to another within limited steps.

Finally, the IOR implies that a large portion of information overload is eliminated by the proposed approach. During the knowledge browsing process, only a small number of domain knowledge items are provided...
to the knowledge users, and this provided knowledge is relatively important to the domain or sub-areas. The IOR only considers the information overload reduction from the perspective of the total number of domain knowledge provided to the knowledge users. In fact, the proposed approach also reduces the information overload from the perspective of the domain knowledge content, because the important knowledge is more connected to the other domain knowledge, which implies that the content is more general and relatively easy to understand.

The above standpoints suggest that the important knowledge contributes to the reduction of the information overload: the constructed knowledge map can conveniently navigate the knowledge users, and the proposed approach can significantly reduce the information overload.

7.2. Further analysis

In this study, one experiment was conducted to investigate the influence of the information retention rate ($\tau$). The experiment focuses on the sensitivity of the information retention rate.

We constructed the domain knowledge maps and validated the results for four different information retention rates ($\tau$) takes four different values (50%, 70%, 85% and 100%). When the information retention rate is 100%, all information from the original similarity matrix will be retained, which indicates that the LSA method is not executed. During the experiment, the efficiency reduction threshold was set to 90%. The experiment result is shown in Table 6.

Table 6 clearly shows that the efficiencies (Ef) of knowledge maps do not change as $\tau$ increases from 50% to 100%. This lack of change may result from the highly clustered internal structure of the final knowledge map. The highly clustered internal structure makes easily results in a high efficiency with a small amount of connections. The small amount of connections tends to be unaffected by value of $\tau$, because they are the largest group of similarity in the similarity matrix. The value of the acceptable level (AL) increases when $\tau$ increases from 50% to 85%. However, if $\tau$ increases from 85% to 100%, AL decreases. Therefore, we can conclude that both the very low value (50%) and the highest value (100%) of $\tau$ are not optimal. We believe that essential semantic information is lost when $\tau$ is 50%, which results in the low AL (64.3%). On the contrary, we believe too much noise information is retained when $\tau$ is 100%, which also results in a low AL (60.1%). Thus, we can conclude that the selection of $\tau$ is critical for improving the acceptable level of the important knowledge. The changes in comprehensiveness ($S$) in response to increases in $\tau$ increases from 50% to 100% are difficult to explain because the total number of important knowledge changes when $\tau$ changes. If we use $S/N_k$ to normalize the comprehensiveness values, the result becomes clear. Similar to the acceptable level, the $S/N_k$ increases when $\tau$ increases from 50% to 85% while it decreases when $\tau$ increases from 85% to 100%. Thus, we can conclude that an optimal $\tau$ lies between 50% and 100%. The value of IOR does not directly relate to $\tau$. According to Eq. (13), it affects by the number of important knowledge ($N_k$). The $N_k$ remains very small compared to the total number of domain knowledge items while $\tau$ increases from 50% to 100%. Therefore, IOR remains large, which suggests that a large portion of information overload can be eliminated by the proposed approach.

In summary, we can conclude that the information retention rate ($\tau$) does not affect the efficiency of the knowledge map and the utility of the proposed approach. However, the acceptable level and $S/N_k$ change with $\tau$, and the proposed best when $\tau$ is 85%.

8. Conclusion and future work

In an effort to alleviate information overload, this paper proposes a domain knowledge browsing-aided method by incorporating a knowledge map and social network analysis. The method requires only a set of domain knowledge items. The corresponding knowledge map can then be constructed according to the content, and the important knowledge in the knowledge map can be identified via the social network analysis method. In this paper, the knowledge map was constructed using methods from text-mining, and basic algorithms from complex networks are used to identify the important knowledge. Hence, the method can be easily implemented from a technology perspective. We also demonstrated the method using a set of domain knowledge from the CNC area, and the result shows that the method can identify the

### Table 3
The important knowledge in the knowledge map.

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>681</td>
<td>Design of numerical control system of conical wheel gear grinding machine based on PMAC</td>
</tr>
<tr>
<td>135</td>
<td>Research on the application of tool nose round radius compensation based on HNC-21 T NC System</td>
</tr>
<tr>
<td>655</td>
<td>Development of CNC system under windows</td>
</tr>
<tr>
<td>678</td>
<td>Study on the numerical control systems for open micro-milling machine based on PC</td>
</tr>
<tr>
<td>268</td>
<td>Failure analysis of CNC system based on PCA</td>
</tr>
<tr>
<td>33</td>
<td>Review of the numerical control machine tool servo system research</td>
</tr>
<tr>
<td>724</td>
<td>Construction of numerical control system for aspect-oriented orthogonal structure</td>
</tr>
<tr>
<td>819</td>
<td>Simulation and automatic programming platform for NC electrochemical machining of integral impeller</td>
</tr>
<tr>
<td>211</td>
<td>CNC turning tool nose radius compensation and processing errors</td>
</tr>
<tr>
<td>227</td>
<td>Grinding of high precise ball bear through NC interpolation on NC round grinding machine</td>
</tr>
</tbody>
</table>

| Sub area | CNC system |

### Table 4
Evaluation of the acceptable level of the important knowledge.

<table>
<thead>
<tr>
<th>Group1</th>
<th>Group2</th>
<th>Group3</th>
<th>Group4</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
</tr>
<tr>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>D1</td>
<td>D1</td>
<td>D1</td>
<td>D1</td>
</tr>
<tr>
<td>D2</td>
<td>D2</td>
<td>D2</td>
<td>D2</td>
</tr>
</tbody>
</table>

| 681 | 5 | 5 | 5 | 4 | 5 | 4 | 5 | 5 | 5 | 4 |
| 135 | 4 | 5 | 5 | 4 | 4 | 4 | 5 | 5 | 4 | 3 |
| 635 | 3 | 5 | 4 | 4 | 3 | 3 | 2 | 3 | 4 | 3 |
| 678 | 5 | 5 | 5 | 4 | 4 | 5 | 5 | 4 | 3 | 4 |
| 268 | 4 | 3 | 4 | 4 | 5 | 3 | 4 | 3 | 3 | 5 |
| 33  | 3 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 3 | 4 |
| 724 | 5 | 4 | 5 | 5 | 5 | 5 | 4 | 5 | 4 | 5 |
| 819 | 4 | 5 | 4 | 3 | 3 | 4 | 3 | 3 | 2 | 3 |
| 211 | 4 | 3 | 3 | 5 | 4 | 4 | 3 | 3 | 4 | 2 |
| 227 | 2 | 3 | 2 | 3 | 3 | 4 | 3 | 3 | 4 | 3 |

### Table 5
The validation result.

<table>
<thead>
<tr>
<th>$\tau$</th>
<th>ER</th>
<th>AL</th>
<th>S</th>
<th>Ef</th>
<th>IOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>85%</td>
<td>90%</td>
<td>76.9%</td>
<td>71.4%</td>
<td>0.51</td>
<td>88.93%</td>
</tr>
</tbody>
</table>

### Table 6
The result of experiment one.

<table>
<thead>
<tr>
<th>$\tau$</th>
<th>AL</th>
<th>S</th>
<th>N_k</th>
<th>S/N_k</th>
<th>Ef</th>
<th>N_k</th>
<th>IOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>50%</td>
<td>64.3%</td>
<td>85.7%</td>
<td>35</td>
<td>2.45%</td>
<td>0.52</td>
<td>11,686</td>
<td>93.11%</td>
</tr>
<tr>
<td>70%</td>
<td>71.4%</td>
<td>57.1%</td>
<td>10</td>
<td>5.71%</td>
<td>0.51</td>
<td>5734</td>
<td>88.93%</td>
</tr>
<tr>
<td>85%</td>
<td>76.9%</td>
<td>71.4%</td>
<td>10</td>
<td>7.14%</td>
<td>0.51</td>
<td>5762</td>
<td>88.93%</td>
</tr>
<tr>
<td>100%</td>
<td>60.1%</td>
<td>71.4%</td>
<td>17</td>
<td>4.20%</td>
<td>0.51</td>
<td>8893</td>
<td>92.22%</td>
</tr>
</tbody>
</table>
important knowledge in the domain and that the constructed knowledge map can efficiently navigate the knowledge user. Despite the progress of the method, some issues need to be addressed in future work.

1) First, we adopted a simple policy in Section 4.4 to decide the similarities that should be retained by analyzing the functions played by different groups of similarities. This policy is intuitively reasonable. Further experiments with different datasets are needed to validate the policy, and different policies should be compared.

2) Second, the connections in the knowledge map are represented by a Boolean value that ignores the degree of similarity between the domain knowledge. Representing the connection with Boolean value is reasonable in this work, because the knowledge construction policy retained only a small amount of high similarities in the final knowledge map, which indicates that the similarities are close to each other. However, the value of similarity should be taken into consideration if we compare different policies.

3) Finally, the definition of important knowledge requires further study. In this paper, we defined the important knowledge as the high degree vertices in the graph. This definition reflects the fact that the domain knowledge that contains more information will have more connections with other domain knowledge. Different definitions of important knowledge should be provided to satisfy different situations. One possible method is to define the important knowledge by the betweenness centrality of the vertices in a graph.

The method introduced in this paper can be used to develop a domain knowledge browsing system. The system can provide the important knowledge items in the domain and a knowledge map can navigate the knowledge users to find the relevant domain knowledge. The domain knowledge base provides knowledge users with detailed information. This system is applicable to navigate new knowledge users who attempt to learn a new domain.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and thank the strong support provided by the National Defense Basic Scientific Research of China (A222011001) and the National Natural Science Foundation of China (51375049).

References


Jia Hao, born in 1984, is currently a Ph.D. candidate at the Department of Industrial Engineering, School of Mechanical Engineering, Beijing Institute of Technology, China. He received his bachelor degree from Beijing Institute of Technology, China, in 2008. And he received his master degree from Beijing Institute of Technology, China, in 2010. His research interests include engineering knowledge management and product design. He is now at the University of Michigan, Ann Arbor, USA as a visiting student. Tel: 1-734-707-9263; E-mail: haojia632@gmail.com.

Yan Yan, born in 1967, is currently a professor at Beijing Institute of Technology, China. She received her doctoral degree in Mechanical Engineering in Beijing Institute of Technology, China, in 2001.

Lin Gong, born in 1979, is currently a lecturer at Beijing Institute of Technology, China. He received his doctoral degree in Mechanical Engineering in Beijing Institute of Technology, China, in 2006.

Guoxin Wang, born in 1977, is currently an associate professor at Beijing Institute of Technology, China. He received his doctoral degree in Mechanical Engineering in Beijing Institute of Technology, China, in 2007.

Jianjun Lin, born in 1980, is currently an engineer at China North Vehicle Research Institute, China. He received his master degree in Mechanical Engineering in Beijing Institute of Technology, China, in 2006.