Comput Syst Sci & Eng (2015) 5: 421–429

International Journal of
Computer Systems
Science & Engineering

Building the concept semantic space for large text database

Xiao Wei[1,2], Daniel Dajun Zeng[2], Wei Wu[1,*] and Yeming Dai[1]

[1] *Shanghai Institute of Technology, Shanghai, China*
[2] *State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. E-mail: shawnwei@outlook.com, dajun.zeng@ia.ac.cn, weiwu@sit.edu.cn, 975169205@qq.com*

To overcome such shortcomings of keyword-based systems on the large text database as low efficiency and recall of searching, this paper proposes a novel concept semantic space to describe the large scale of text database efficiently. The proposed concept semantic space describes the text database from multiple semantic granularities (i.e. keyword, concept, etc.) and multiple semantic dimensions (i.e. association relations, similar relations, etc.), which provides a macroscopic and dynamic view of the text database. With the support of concept semantic space, some novel systems can be constructed on the text database to provide novel, efficient and flexible services. In this paper, take association relation for example, the main steps of building such a concept semantic space on text database are discussed in detail. In the end, both the experimental results and a prototype system, named as Knowle, show that the proposed concept semantic space is efficient in organizing the text database.

Keywords: text database, concept semantic space, multiple semantic granularities, multiple semantic dimensions, semantic link network, concept extraction.

1. INTRODUCTION

Most of current web applications work on keywords or relations among keywords. For example, Google provides searching service based on keywords matching; some systems recommend objects based on the association rules among keywords. Much work has focused on mining keywords and their relations [4, 5, 14, 15]. The main shortcomings of keyword-based systems are as follows.

1) Low efficiency. Keyword matching-based systems need to hold entire keywords of each text to keep its high precision, which makes the system store and process a large scale of data.

2) Low recall ratio. Keyword belongs to the low semantic level and is used to provide exact matching. When keywords of synonym or hyponymy are used to search, the keyword-based system may not return the expected results to the searcher. Although semantic dictionary can be used to support searching by keywords of synonym or hyponymy, the shortcomings of semantic dictionary, such as slowly updating, limitation of keywords in dictionary, and so on, reduce the recall ratio.

*Corresponding author: Wei Wu E-mail: weiwu@sit.edu.cn

Compared with a keyword, a concept [12, 16] has a bigger semantic granularity and holds more semantic information, which is used in ontology construction, text semantic representation, semantic annotation, semantic search, etc., in order to improve the efficiency of text semantic processing. Concept-based system removes the above shortcomings of keyword-based system. At the same time, concept-based system also has its own shortcomings. For example, concept-based searching will return more results than keyword-based searching, which reduces its precision. Therefore, to a large text database such as webpages, a multi-layers semantic description which considers both keyword level and concept level could support its applications with high precision and recall.

How to build an efficient semantic description of a large text database is a key issue to be solved for all kinds of semantic services on the text database. To solve the problem, this paper proposes a concept semantic space to describe a large scale of text database to support efficient and flexible service on the database.

Concept space is the semantic organization model of concepts, which haven't had a uniform definition until now. Different definitions are proposed to fit the special research purposes and application scenarios [1, 9, 17]. In [9], concept space is formed by concepts and the semantic relation network of concepts. In [1], concept space is concepts and the semantic relations among concepts, among which concept is extracted and clustered from

keywords of text several rounds. In the above definitions, concept space is used as a container to save concepts and the semantic relations among them, which is a static space and cannot make an automatic evaluation.

In a large text database like the Web, the texts change frequently, which leads to the semantic activities of concepts, such as, the appearance of new concepts, the disappearance of old concepts, the semantic changing of current concept, etc. The above concept space cannot describe the above semantic changing of dynamic large text database efficiently. To overcome the shortcomings of the current concept space, the proposed concept space in this paper is not only the container of concepts and their semantic relations but also the space of all kinds of semantic activities.

The proposed concept sematic space of large text database should have the following features in order to meet the requirements as discussed above.

1) Multiple semantic granularities. The concept space should describe the text database from multiple semantic levels, such as keyword, concept, and so on, in order to support services at different semantic levels.

2) Multiple semantic dimensions. The concept space should hold different kinds of semantic relations, such as association relations, similar relations and so on, in order to support services of different types of semantics.

3) Dynamic. The concept space should change dynamically and automatically according to the changing of text database in order to support the service based on concept analysis.

4) Macroscopic view. The concept space should provide a global view of the text database in order to support the service based on global analysis of the database.

The main work of this paper is to propose such a concept space to fit the above features and the method to build the concept space on a large text database.

The rest of this paper is organized as follows. In section 2, some related work is discussed. In section 3, the textual concept semantic space is defined and the main steps of building a concept semantic space are discussed. Section 4 discusses how to build the semantic link network of keywords for concept semantic space. Section 5 discusses how to extract concepts from semantic link network of keywords. In section 6, some experiments are shown to evaluate the proposed method. Finally, the conclusion is reached in section 7.

2. RELATED WORK

2.1 Concept

The definitions of concept in philosophy, linguistics, logic, psychology, cognitive informatics, software engineering and knowledge engineering are not all the same [20] Philosophically, concept is the basic unit of thinking. In artificial intelligence, concept is used to model the knowledge of human. In linguistics, concept is a noun or noun phrase as the subject of to-be structure [13] In cognitive informatics, concept is an abstract structure with exact semantics of cognitive process, such as, thinking, learning, and reasoning [21]. In the above domains, concept is defined as the basic unit of thinking, learning and reasoning. A concept has intension and extension, namely its meaning (attribute words) and scope (example).

Facing the specific application area of the automatic semantic processing on large text database, it is impossible and useless to get all the intension and extension of a concept exactly, efficiently, and automatically. It is enough to get limited attribute keywords to describe a concept which could meet the practical requirement. Therefore in this paper a concept is a keyword of high semantic level, which can be described by several keywords or concepts at low semantic level.

2.2 Semantic link network

Semantic link network is an efficient organization model of web resources. In [6], a web resource space model is proposed based on semantic link network. [8, 18, 19] use semantic link network to organize web resources efficiently. [18] proposed the construction method of association semantic link network. [8] builds a multi-layer association link network of keywords.[19] proposed the method to build the similar semantic link network on a large scale of web resources.

In this paper, semantic link network is used as a basic tool to organize objects at different levels of the concept space.

3. TEXTUAL CONCEPT SEMANTIC SPACE

3.1 Definition of textual concept semantic space

Based on the discussion in section 2.1, a concept fits the requirement of processing a large scale of texts can be defined as following.

Definition 1. Concept (C): Concept is the abstract description of objects, which is formed by attributive keywords and the semantic relations among attributive keywords and denoted as

$$C = \langle P, R^P \rangle = \left\langle \begin{array}{l} P = \{t_k | t_k \in T, 0 \le k \le |T|\}, \\ R^P = \{(t_i, t_j, w) | t_i, t_j \in P, \\ \qquad 1 \le i, j \le |P|, 0 \le w \le 1\} \end{array} \right\rangle, \quad (1)$$

in which, T is the domain keywords list, P is the attribution set of concept C which is formed by the keywords t_k that belongs to T. $|T|$ is the length of T. R^P is the set of semantic relations in P, each semantic relation is described by a triad (t_i, t_j, w), in which w denotes the strength of the relation between t_i and t_j. |P| is the length of P.

A concept C is presented as a semantic link network, as the example shown in Figure 1(a). Concepts belong to different semantic levels. A concept may be the attributive words of the concept of higher semantic level. At the same time, it can consist of keywords or concepts of lower semantic level. This kind of

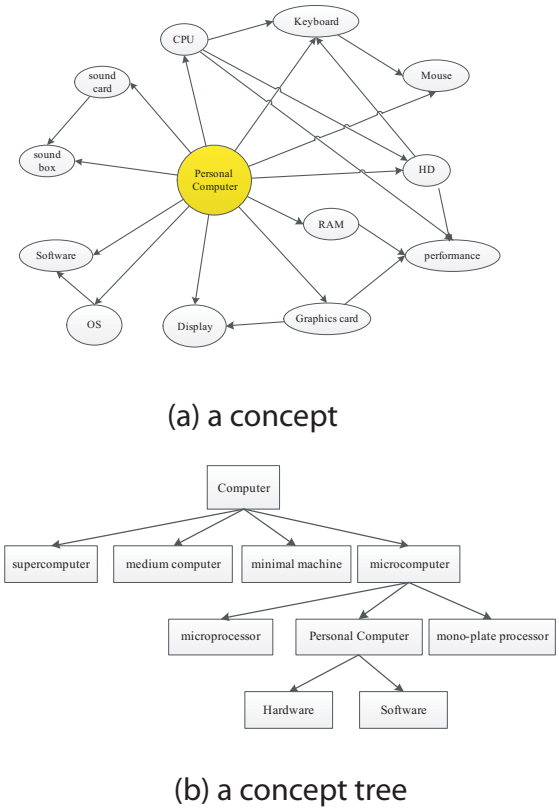


Figure 1 Examples of concept and concept tree.

inclusion relations among concepts are shown as concept tree, as the example shown in Figure 1(b).

Definition 2. Textual Concept Semantic Space (TCSS). TCSS is an open system composed of text set, semantic link network of keywords (keywords and the semantic relations among keywords), and semantic link network of concepts (concepts and the semantic relations among concepts), which is the space that holds all kinds of semantic activities of concepts. TCSS is denoted as

$$TCSS = \left\langle D, tSLN : (T, R^T), cSLN : (C, R^C) \right\rangle, \quad (2)$$

in which, D is the set of texts in the large scale text database, *tSLN* is the semantic link network of keywords/terms[1] which is composed of the set of all keywords of texts, denoted as T, and the set of semantic relations in T, denoted as R^T; *cSLN* is the semantic link network of concepts which is composed of the set of all concepts in the space, denoted as C, and the set of semantic relations in C, denoted as R^C. The structure of TCSS is shown in Figure 2.

Defintion 3. Semantic Link Network of keywords (*tSLN*). *tSLN* is a network composed of all keywords of texts and the semantic relations among keywords, denoted as

$$tSLN = \langle T, R^T \rangle = \left\langle \begin{array}{l} T, R^T = \{(t_i, t_j, [w_a, w_s]) | t_i, t_j \in T, \\ 1 \le i, j \le |T|, 0 \le w_a, w_s \le 1\} \end{array} \right\rangle, \quad (3)$$

[1] Keyword means not only a single word but also phrase, in this paper both single word and phrase are called as keywords.

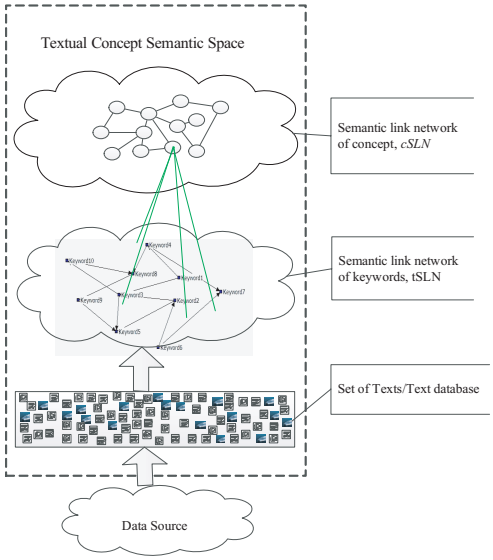


Figure 2 The structure of textual concept semantic space.

in which, T is the set of nodes in the network, each node denotes a keyword; R^T is the set of edges in the network, each edge is described by a triad $(t_i, t_j, [w_a, w_s])$[2], t_i and t_j are the two nodes of the edge, w_a denotes the weight of the association relation between t_i and t_j, w_s denotes the weight of the similarity relation between t_i and t_j.

Definition 4. Semantic Link Network of Concepts (*cSLN*). *cSLN* is a network composed of all concepts of texts and the semantic relations among concepts, denoted as

$$cSLN = \langle C, R^C \rangle = \left\langle \begin{array}{l} C, R^C = \{(c_i, c_j, [w_a, w_s]) | c_i, c_j \in C, \\ 1 \le i, j \le |C|, 0 \le w_a, w_s \le 1\} \end{array} \right\rangle, \quad (4)$$

in which, C is the set of nodes in the network, each node denotes a concept; R^C is the set of edges in the network, each edge is described by a triad $(c_i, c_j, [w_a, w_s])$, c_i and c_j are the two nodes of the edge, w_a denotes the weight of the association relation between c_i and c_j, w_s denotes the weight of the similarity between c_i and c_j.

3.2 Construct TCSS

The construction of TCSS should be totally automatic to fit the requirement of processing a large scale text database. According to the definitions of TCSS, the procedure to build the TCSS is as follows.

1) Build the semantic link network of keywords, which consists of extracting keywords from texts, mining the semantic relation among keywords, building the semantic link network of keywords, and detecting the community in tSLN. In this paper, we take the association relation for example to discuss how to build tSLN, which is discussed in detail in section 4.

[2] Although there may exist kinds of semantic relations among keywords, this paper only considers two kinds of relations, association and similarity relations.

2) Extract concepts from tSLN, which consists of detecting concept words, selecting attributive words for concept, describing a concept. The details of extracting concepts are discussed in section 5.

3) Build the semantic link network of concept, which is similar to the process of building semantic link network of keywords and isn't discussed in detail in this paper.

Based on the analysis of the definition and construction process of TCSS, the proposed TCSS has the feature expected in the introduction.

1) Multiple semantic granularities. TCSS provides three semantic granularities: text, keyword and concept.

2) Multiple semantic dimensions. TCSS provides two types of semantic dimensions: association and similarity.

3) Dynamic. TCSS is open and texts from data source, like the Web can enter the TCSS timely, which can bring about changes at keyword level or concept level with the support of automatic construction algorithms.

4) Macroscopic view. Both the semantic link network of concepts and the network of keywords describe the set of text from the globe view, which can be realized by some complex network analysis method.

4. BUILD THE SEMANTIC LINK NETWORK OF KEYWORDS

4.1 Generate the node set of tSLN

Given a set of texts, namely the text database, denoted as

$$D = \{d_1, d_2, ..., d_i, ..., d_n\}, \tag{5}$$

in which, d_i denotes the i^{th} text in D and is described by a set of keywords

$$d_i = \{tw_{i1}, tw_{i2}, \ldots, tw_{ij}, \ldots, tw_{is}\}, \tag{6}$$

in which tw_{ij} is the weight of the j^{th} keywords in d_i. Then all keywords of texts in D form the set of nodes, T, in the *tSLN*, which can be gotten by the joint operation of all d.

$$T = d_1 \cap d_2 \cap \cdots d_n \tag{7}$$

4.2 Generate the edge set of tSLN

The textual concept semantic space should hold all kinds of semantic relations. Each kind of semantic relation has its own features and mining methods. In this section, we take association relation for example to discuss how to construct the semantic link network of keywords.

When only association relation is considered, each edge in tSLN denotes the association relation between a pair of keywords. When each keyword is considered as an item and each text is considered as a transaction, the association rules among keywords can be mined by the current mining algorithms. Each pair of association rule connects a pair of keywords, which can be used as the initial edge of tSLN.

(1) The weight of Association Rule (WAR) Given a pair of keywords $< A, B >$, if there exists an association rule from A to B, denoted as AR_{AB}, then *WAR* is equal to the weight of AR_{AB}. Otherwise, if there is no association rule from A to B, then *WAR* is zero. *WAR* is defined as

$$WAR_{AB} = \begin{cases} weight_{AB}, & if\ AR_{AB}\ exists \\ 0, & else \end{cases}, \tag{8}$$

in which, AR_{AB} presents an association rule from A to B, $weight_{AB}$ is the weight of AR_{AB}.

(2) The contribution of association rule (CAR) Sometimes the *WAR* of an association rule cannot reflect its real importance to the text database. To association rules with the same WAR, some association rules are frequently used while others are seldom used. Generally, an association rule which is frequently used should be more important than those seldom be used. We use the contribution of association rule, denoted as *ARC*, to present the important degree of an association rule to the text dataset, which is calculated by

$$CAR_{AB} = p/q, \tag{9}$$

in which, p is the times that this association rule is used in the text dataset, q is the total times that all association rules are used in the text database.

(3) The association weight of a pair of keywords (kaw) The final weight of association relation from keywords A to B is influenced by the above two factors, which is calculated by

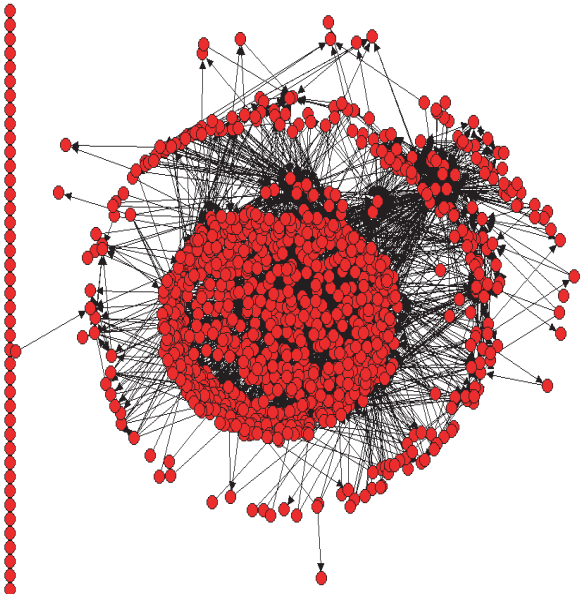
$$kaw_{AB} = \alpha * WAR_{AB} + \beta * CAR_{AB}, \tag{10}$$

in which, $\alpha + \beta = 1$. $\alpha > \beta$, which means C*AR* is more important than *WAR*. In our dataset, we set $\alpha = 0.75$ and $\beta = 0.25$ based on the experimental results.

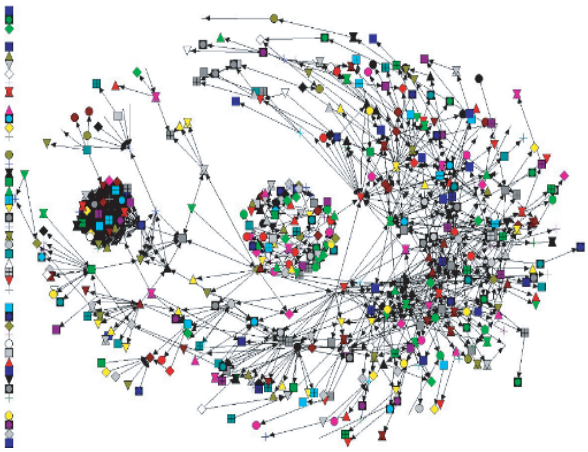
4.3 Construct tSLN

According to the above analysis, it is easy and efficient to generate the nodes (keyword set) of tSLN and calculate the *WARs* of edges. Because the text database is changing dynamically, the value of $p$in equation (9) can be gotten by traversing the whole text database. This step is quite complex. Supposing n texts, it needs n^2 steps to traverse each pair of texts in the text database. Supposing there are s keywords in each text, it needs further s^2 steps to check the association rules in a pair of texts. So it needs total $s^2 \times n^2$ steps to traverse all the keywords of all texts. For example, a small application has 1000 texts and each text has 20 keywords, the basic computing in the method is 4×10^8. So it is necessary to find a faster algorithm to construct *tSLN*.

From the above analysis, the effective method is to reduce the cycles of traversal. Here we give a small example of text database. Supposing the database consists of three texts $\{d1, d2, d3\}$ and the keywords of each text are $\{\{k1, k2, k3\}, \{k1, k3\}, \{k2, k3\}\}$. The association rules are $\{k1 \rightarrow k2, k2 \rightarrow k3, k1 \rightarrow k3\}$. Then the association rules are used in the text database as following.



(a)The initial tSLN (based on association relation)



(b)The tSLN after edges optimizing (based on association relation)

Figure 3 An example of tSLN.

$$L = \begin{bmatrix} & d1\{k1,k2,k3\} & d2\{k2,k3\} & d3\{k1,k3\} \\ d1\{k1,k2,k3\} & \begin{array}{l} k1 \to k2, \\ k1 \to k3, \\ k2 \to k3 \end{array} & \begin{array}{l} k1 \to k2, \\ k1 \to k3, \\ k2 \to k3 \end{array} & \begin{array}{l} k1 \to k3, \\ k2 \to k3 \end{array} \\ d2\{k2,k3\} & k2 \to k3 & k2 \to k3 & k2 \to k3 \\ d3\{k1,k3\} & \begin{array}{l} k1 \to k2, \\ k1 \to k3 \end{array} & \begin{array}{l} k1 \to k2, \\ k1 \to k3 \end{array} & k1 \to k3 \end{bmatrix}$$

From L, the number of each AR is counted and listed as

$$K = \begin{bmatrix} & k1:2 & k2:2 & k3:3 \\ k1:2 & 0 & 4 & 6 \\ k2:2 & 4 & 0 & 6 \\ k3:3 & 6 & 6 & 0 \end{bmatrix},$$

in which, the number after the keyword is Document Frequency (DF). For example, $k3:3$ means the keyword $k3$ is used three times in all texts. Based on the analysis on K, the used number of an association rule is just equal to the product of the two keywords' DFs. For example, $k1 \to k3$ is used six times, which is also equal to 2×3. 2 and 3 are the DFs of k1 and k3 respectively.

Based on the above example, *tSLN* construction method consists of two steps. The first step is calculating DF. The second step is to count the used time of each AR. The complexity of the new method is analyzed as follows. Supposing the number of association rules is q, the number of texts is still n, the number of keywords of each text is still s, the total number of keywords is w, and then the complexity of counting DF is $O(w)$, the complex of counting K is $O(w^2)$. Then the final complexity is $O(w^2)$. Compared with$s^2 \times n^2$, w^2 is much smaller because of $w \ll s \times n$.

Figure 3 shows an example of the tSLN of a sample text database.

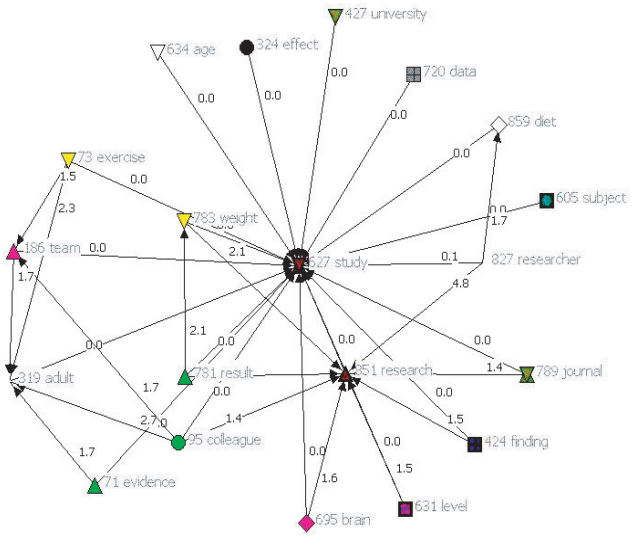


Figure 4 An example of keyword community in tSLN.

4.4 tSLN community

After analyzing the semantic link network of keywords, such as association relation network, there exist communities in the tSLN, which are the clustering of keywords based on association relations. That is to say, the tSLN community is the set of keywords which are related. An example tSLN community is shown in Figure 4. After the analysis on the dataset, we notice that in the community there exist one or more central words, which are used to select concept words in the next section.

There are already many researches on the community structure of complex network [2, 7, 10, 11]. Because tSLN is a typical complex network, we can select some appreciated ones from these methods to detect the communities in tSLN.

5. EXTRACT CONCEPTS FROM TSLN

5.1 Select concepts from tSLN

According to the distribution of communities in tSLN, each community has at least a central keyword. Compared with other keywords in the same community, the central keyword has stronger and more association relations with other keywords, which is more suitable to act as a concept to denote the community.

How to identify the central keyword from a community in

tSLN? After analyzing the graphic features of tSLN, it is more appropriate to select the nodes whose degrees are bigger than the average of all nodes' degrees. Another factor to be considered is the minimal degree of the concept node. Taking account of the summary and abstract ability of a concept, we select five as the minimal degree of a concept in our method. Based on the above idea, we propose an algorithm for selecting concept from tSLN, which is shown in algorithm 1.

Algorithm 1: Select concepts from tSLN

-Input: *tSLN* (stored as a sparse matrix), m(the number of keywords),miniDegree(the minimal degree to select a concept)
-Output: C (concept set)
-Description: Select concepts from *tSLN* according to the graphic features of *tSLN*.

```
AverageDegree = 0
Degrees[m]=[0,0,...,0]
foreachkaw_ij in tSLN
{
++ AverageDegree;
++ Degrees[i]; // out degree
++ Degrees[j]; // in degree
}
AverageDegree = AverageDegree /m;
if (AverageDegree < miniDegree)
AverageDegree = miniDegree
for k = m-1 to 0
if (Degrees[k]< AverageDegree);
remove Degrees[k]
C = Degrees.
end
```

The complex of algorithm1 is $O(h)$, where h is the number of edges in tSLN, which means it is an efficient algorithm.

5.2 Select candidate attributive keyword set for each concept

The next step is to select candidate attributive keywords to describe each concept. According to the graphic features of tSLN, the neighbor nodes (keywords) of the concept node have strong relations with the concept node than other nodes(keywords). Although these nodes have strong relations with the concept node, they may be not the most suitable ones to describe the concept. In this step, we do not consider if a keyword is the best attributive keyword to describe the concept, we just select the most relative keywords as the candidate attributive keywords for a concept. How to select the best ones from the candidate keywords will be discussed in the following subsection.

In tSLN, the strength of semantic relation decreases with the distance between a keyword and the concept node increase. Here we select all the direct neighbor nodes and the two-order neighbor nodes as the candidate attributive keyword set for the concept.

The candidate attributive keyword set is denoted as

$$ckw = \{\ kw_1,\quad kw_2,\quad kw_3,\quad ...,\quad kw_t\ \}, \tag{11}$$

in which , each kw is an attributive keyword of the concept.

Some examples of candidate attributive keyword set are as follows.

Virus={bird, chicken, Egypt, farm, flu, HIV, human, Indonesia, outbreak, pandemic, poultry, strain}

Study={research, university, data, det, researcher, journal, finding, brain, colleague, evidence, result, adult, team, exercise, age, effect}

5.3 Select best attributive keywords to describe a concept

Generally, a concept is easier to be understood with the growth of number of attributive keywords. However, more attributive keywords mean more cost to deal with a concept. In the candidate attributive keyword set of a concept, keywords are important to the concept with different degrees. Therefore, it needs to filter suitable ones from the candidate attributive keywords.

A concept can also be considered as an information system. Here we take advantage of methods of Entropy and Mutual Information to select the best keywords from candidate ones as the final attributive keywords to describe the concept . Compared with the terms of Information Theory, a concept C can be considered as an information source and attributive keywords can be considered as the signal sent by the information source. As the keywords are statistically independent, the concept C is a Zero-Memory Discrete Information Source. The probability presentation of the concept C is

$$\begin{bmatrix} C \\ p(c) \end{bmatrix} = \begin{bmatrix} kw_1 & kw_2 & kw_3 & ... & kw_t \\ p(kw_1) & p(kw_2) & p(kw_3) & ... & p(kw_t) \end{bmatrix}, \tag{12}$$

in which, $p(kw_i)$ is the probability that the keyword kw_i appears in a text, and $\sum_{i=1}^{t} p(kw_i) = 1$.

Definition 5: Probability of a single keyword, denoted as $p(kw_i)$. For a given text set, the probability of a single keyword kw_i is its text frequency in text set, calculated by

$$p(kw_i) = u/v \tag{13}$$

in which v is the total of texts in the set, u is the number of texts which include the keyword kw_i.

Definition 6: Probability of keyword set, denoted as $p(kw_1, ..., kw_i)$, can be calculated as

$$p(kw_1, kw_2, ..., kw_i) = u'/v \tag{14}$$

in which u' is the number of texts which include $kw_1, kw_2, ..., kw_i$. Meanwhile, v is the total number of texts in the text set.

Definition 7: Self-information of a keyword, denoted as $I(kw_i)$. Self-information of a keyword is a measure of the information content generated by the outcome of keyword kw_i, which can be calculated by

$$I(kw_i) = -\log p(kw_i) \tag{15}$$

Algorithm2: Generating the attributive keyword set for a concept

-Input: $ckw = \{ kw_1, kw_2, kw_3, ..., kw_t \}$, Tkw Tkw is the text set.
ckw is the candidate keyword set.
-Output: $C = \{ kw_1, kw_2, kw_3, ..., kw_{t'} \}$
C is the concept represented by attributive keywords.
- Description: This algorithm select the most appropriate attributive keywords from ckw to describe the concept C.
$C = \emptyset$
Calculate $H(kw_i) = p(kw_i), i = 1..t$
$C = \max(H(kw_i)), i = 1..t$
ckw = ckw - C
Calculate$I(C; kw_i)kw_i \in ckw$
$kw = \min(I(C; kw_i))kw_i \in ckw$
$C = C \cup kw$
$ckw = ckw - kw$
$if \Delta(H(C)) \to 0$ or $I(C; kw_i) = 0 kw_i \in ckw$
goto step12
else goto step 5
end

according to the definition of self-information in Information Theory.

Definition 8: Conditional self-information content of a keyword, denoted as$I(kw_{i+1}|kw_1, kw_2, ...kw_i)$. The conditional self-information content of keyword is a measure of the information content generated by the outcome of keyword kw_{i+1} on the condition of the outcome of kw_1 to kw_i, which can be calculated by

$$I(kw_{i+1}|kw_1, kw_2, ..., kw_i) = -\log p(kw_{i+1}|kw_1, kw_2, ..., kw_i) \tag{16}$$

Definition 9: Entropy of concept, denoted as $H(C)$. The mathematical expectation of self-information content kw_i refers to *mean information content of concept C*, which is also called as Entropy of concept, can be calculated by

$$H(C) = -\sum_{i=1}^{t} p(kw_i) \log p(kw_i) \tag{17}$$

Definition 10: Mutual information between keyword and keyword set, denoted as $I(kw_{i+1}; kw_1, kw_2, ..., kw_i)$. Suppose the keyword set has already owned i keywords, denoted as $\{ kw_1, kw_2, kw_3, ..., kw_i \}$, the mutual information between the following keyword kw_{i+1} and existing keywords can be calculated by

$$\begin{aligned} &I(kw_{i+1}; kw_1, kw_2, ..., kw_i) \\ &= I(kw_{i+1}) - I(kw_{i+1}|kw_1, kw_2, ..., kw_i) \end{aligned} \tag{18}$$

Based on the above equations (12)–(18), we propose an algorithm for generating the attributive keyword set for a concept based on its candidate attributive keyword set, as shown in Algorithm 2.

Algorithm 2 first selects the core attributive keyword from the candidate keyword set (step 1 to 3), then adds the other attributive keywords one by one by considering the mutual information between the keyword and the selected keyword set.

Table 1 Experimental dataset

	Dataset1	*Dataset2*
domain	Environment news of Reuters	Health news of Reuters
date of news	01/2009-12/2009	01/2009-12/2009
number of news	6445	2898
number of keywords	568	457

For an example of Algorithm2, in the testing dataset the candidate keyword set of concept 'study' is ckw_{Study} = *{research, university, data, det, researcher, journal, finding, brain, colleague, evidence, result, adult, team, exercise, age, effect}*. After the Algorithm2 runs, the output of Concept '*study*' is C_{study} = *{study, journal, result, researcher, university, colleague, team, exercise, finding}*.

6. EXPERIMENTS

In this section, we use experiments to evaluate the most important steps of constructing textual concept semantic space.

6.1 Dataset

The two datasets used in the experiments are shown in Table 1. Both are composed of news from Reuters during 01/2009-12/2009. Each news page is described by 10 keywords. Totally, dataset1 has 6445 news and 568 individual keywords; datase2 has 2898 news and 457 individual keywords.

6.2 Experimental process

The two datasets are processed by the proposed methods in this paper as the following steps and the results are shown in Table 2.

1. Construct tSLN;
2. Detect tSLN communities; the numbers of communities are shown in the first row of table 2.
3. Select concept words from tSLN; the numbers of concept are shown in the second row of table 2.
4. Select candidate attributive keywords for each concept; the average sizes of candidate attributive keyword sets are shown in the third row of table 2.
5. Select final attributive keyword set for each concept; the average sizes of the final attributive keyword sets are shown in the fourth row of table 2.
6. Evaluate the final concept by Wordnet[3] and the accuracy ratios are shown in the fifth row of table 2.
7. Evaluate the final concept by human, and the accuracy ratios are shown in the last row of table 2.

Table 2 Experimental results

	Dataset1	*Dataset2*
number of tSLN communities	35	25
number of concepts	94	65
average size of candidate attributive keyword set	14.4	15.4
average size of attributive keyword set	7.52	8.34
accuracy ratio (evaluated by WordNet)	42%	45%
accuracy ratio (evaluated by human)	65%	62%

6.3 Analysis

(1) Evaluation by WordNet As to each concept and its attributive keywords, the evaluation method is to search all the keywords in *WordNet*. If half of the attributive keywords are in the same sub-tree but lower than the concept. Thus, the concept just passes the evaluation. Otherwise, the concept fails in the evaluation.

As is shown in Table 2, the accuracy rates are 0.42 and 0.45 respectively, which are low. The main reason is that some keywords are not included in the *WordNet*. Another reason is some of the mined attributive keywords for a concept are cross. These reasons show that the real accuracy ratio may be higher than these ones.

(2) Evaluation by Human Considering the reason for the low rate of evaluation by *WordNet*, we evaluate the results by human beings by means of manual analysis. In fact, some concepts failing in the evaluation of *WordNet* are considered as passing the evaluation after the analysis. As a result, the accuracy ratio increases obviously about 20 percentage points.

The accuracy of the proposed method still has much room to increase. However, as automatic method, the proposed method can work together with semantic dictionary, i.e. WordNet, which can use advantages of both automatic method and human dictionary.

6.4 A prototype system based on the textual concept semantic space

The 'Knowle' system is a news retrieve system based on textual concept semantic space. In Knowle, the news webpages are organized based on concept semantic space from four semantic layers: webpages, keywords, concepts, topics. In each layer, the semantic link network is used to organize the objects: webpages, keywords, concepts and topics, which supports Knowle to provide the network based results. Knowle provides a concept-based retrieve and the interface of Knowle is shown in Figure 5.

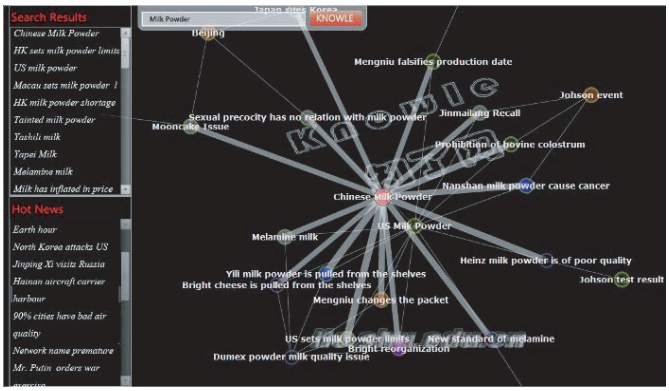


Figure 5 Knowle (A news retrieve system based on textual concept semantic space).

7. CONCLUSION

Keyword-based systems on the large text database have such shortcomings as low efficiency and recall of searching. Some novel and efficient description model are expected to overcome these shortcomings. This paper proposes a novel concept semantic space to describe the large scale of text database efficiently. The proposed concept semantic space describes the text database from multiple semantic granularities (i.e. keyword, concept, etc.) and multiple semantic dimensions (i.e. association relations, similar relations, etc.), which provides a macroscopic and dynamic view of the text database. With the support of concept semantic space, some novel systems can be constructed on the text database to provide novel, efficient and flexible services. Then this paper takes association relation for example to discuss the main steps of constructing such a concept semantic space on a text database. In the end, both the experimental results and a prototype system, named as Knowle, show that the proposed concept semantic space is efficient in organizing the text database.

The future work of this paper is to analyze the basic features and evolution rules of concept activities in the concept semantic space in order to support more useful services on the text database.

ACKNOWLEDGMENTS

Research work reported in this paper was supported by the National Science Foundation of China under grant no. 61471232.

REFERENCES

1. **C. C. van der Eijk, E. M. van Mulligen, J. A. Kors, et al**. Constructing an associative concept space for literature-based discovery. J*ournal of the American Society for Information Science and Technology*, 55(5): 436–444, 2004.
2. **F. Radicchi, C. Castellano, F. Cecconi et al.** Defining and identifying communities in networks. *PNAS*, 101(9): 2658–2663, 2004.
3. **G. A. Miller.** WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

4. **H. Zhuge.** Communities and Emerging Semantics in Semantic Link Network: Discovery and Learning. *IEEE Transactions on Knowledge and Data Engineering*, 21(6): 785–799, 2009.
5. **H. Zhuge.** Interactive Semantics. *Artificial Intelligence*, 174(2):190–204, 2010.
6. **H. Zhuge and Y. Xing.** Probabilistic Resource Space Model for Managing Resources in Cyber-Physical Society. *IEEE Transactions on Service Computing,* 5(3): 404–421, 2012.
7. **J. Tyler, D. Wilkinson and B. Huberman**. Email as spectroscopy: Automated discovery of community structure within organizations. *The Information Society*, 21(2): 143–153, 2005.
8. **J. Xuan, X. Luo, S. Zhang, et al.** Building Hierarchical Keyword Level Association Link Networks for Web Events Semantic Analysis. *In Proceedings of IEEE Ninth International Conference on Dependable Autonomic and Secure Computing,* pages 987–994, IEEE, 2011.
9. **K. Hori.** Concept space connected to knowledge processing for supporting creative design. Knowledge-Based Systems, 10(1): 29–35, 1997.
10. **M. E. Newman.** Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6): 066133, 2004.
11. **M. Girvan and M. E. Newman.** Community structure in social and biological networks. *PNAS*, 99(12): 7821–7826, 2001.
12. **P. Cimiano**. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of Artificial Intelligence Research,* 24: 305–339, 2005.
13. **P. J. Hurley.** *A concise introduction to logic (6th ed.).* Belmony, CA: Wadsworth, 1997.
14. **R. Cooley, B. Mobasher and J. Srivastava.** Web mining: Information and pattern discovery on the World Wide Web. *In Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence,* pages 558–567. IEEE, 1997.
15. **R. Agrawal and R. Srikant**. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. 1994.
16. **R. Wille**. Conceptual graphs and formal concept analysis. Conceptual Structures: Fulfilling Peirce's Dream. *Lecture Notes in Computer Science*, 1257:290–303, 1997.
17. **S. Bruce Schatz, et.al.** The Interspace Prototype: An Analysis Environment for Semantic Interoperabilty. http://www.canis.uiuc.edu/projects/interspace/ interspace-demo.pdf.
18. **X. Luo, Z. Xu, J. Yu, and X. Chen.** Building Association Link Network for Semantic Link on Web Resources. *IEEE Transactions on Automation Science and Engineering*, 8(3): 482–494, 2011.
19. **X. Luo, J. Ni, J. Zhang, et al.** Building Similar Link Network in Large-Scale Web Resources. *In Proceedings of IEEE 16th International Conference on Parallel and Distributed Systems*, pages 87–693, IEEE, 2010.
20. **Y. Wang.** On concept algebra: A denotational mathematical structure for knowledge and software modeling. *International Journal of Cognitive Informatics and Natural Intelligence*, 2(2): 1–19, 2008.
21. **Y. Wang.** The theoretical framework of cognitive informatics. *The International Journal of Cognitive Informatics and Natural Intelligence*, 1(1): 1–27, 2007.