Neural Comput & Applic (2014) 24:779–798
DOI 10.1007/s00521-012-1272-z

ORIGINAL ARTICLE

Domain ontology graph model and its application in Chinese text classification

James N. K. Liu · Yu-lin He · Edward H. Y. Lim · Xi-zhao Wang

Received: 9 October 2011 / Accepted: 10 November 2012 / Published online: 19 December 2012

Abstract This paper proposes an ontology learning method which is used to generate a graphical ontology structure called ontology graph. The ontology graph defines the ontology and knowledge conceptualization model, and the ontology learning process defines the method of semiautomatic learning and generates ontology graphs from Chinese texts of different domains, the so-called domain ontology graph (DOG). Meanwhile, we also define two other ontological operations—document ontology graph generation and ontology graph-based text classification, which can be carried out with the generated DOG. This research focuses on Chinese text data, and furthermore, we conduct two experiments: the DOG generation and ontology graph-based text classification, with Chinese texts as the experimental data. The first experiment generates ten DOGs as the ontology graph instances to represent ten different domains of knowledge. The generated DOGs are then further used for the second experiment to provide performance evaluation. The ontology graph-based approach is able to achieve high text classification accuracy (with 92.3 % in *f*-measure) over other text classification approaches (such as 86.8 % in *f*-measure for tf–idf approach). The better performance in the comparative experiments reveals that the proposed ontology graph knowledge model, the ontology learning and generation process, and the ontological operations are feasible and effective.

Keywords Domain ontology graph · Knowledge representation · Text classification · Ontology

J. N. K. Liu (✉) · E. H. Y. Lim
Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
e-mail: csnkliu@comp.polyu.edu.hk

E. H. Y. Lim
e-mail: edward@iatopia.com

Y. He (✉) · X. Wang
College of Mathematics and Computer Science, Hebei University, Baoding 071002, China
e-mail: csylhe@gmail.com

X. Wang
e-mail: xizhaowang@ieee.org

1 Introduction

Ontology is a fundamental representation form for the knowledge in real world. From the perspective of computer science and information science, ontology [3, 14] defines a set of representational primitives with which the domain of knowledge or discourse can be modeled. A well-constructed ontology can be effectively used to develop a knowledge-based information search and management system in many fields, for example, artificial intelligence [23], semantic web [21], software engineering [28], biomedical informatics [31], etc. However, due to the lack of essential knowledge as the core components, most of these existing systems are ineffective in terms of low accuracy in searching and managing information (especially for Chinese text data). Therefore, ontology that is acknowledged as a knowledge representation language is becoming a very important research area for developing the knowledge-based information systems.

The major challenge for ontology learning is to create and maintain an effective and efficient ontology with the minimal human intervention. In order to achieve this objective, many representative ontology learning methodologies [2, 17, 24, 30, 34, 40, 44] are designed to develop the related theories, methods, or software tools which can

be used to obtain the required ontology. However, in the process of ontology construction and learning, these existing ontology learning methodologies always need much more direct human interventions. These interventions may be required at the early stage [24, 44], that is, applying the analytic method for matching strategy selection before the execution of ontology extraction, or at the end of the domain ontology editing [34, 40], that is, correcting or reusing the learned conceptualizations. Meanwhile, some colleagues [2, 17, 30] also argue that human intervention could be placed at the middle phase with an iterative and dynamic method of learning the concepts or properties. It is well acknowledged [3, 14, 21] that the manual creating or maintaining the ontology is time-consuming and inefficient; thus, the simplified ontology learning method with little or minimal human intervention is more practical and feasible to handle and represent the information inherent in the real applications based on the semantic webs.

In addition, the ontology learning from text data is a very interesting and useful method in formalizing ontology, because the text data are a kind of important and rich source of human knowledge. Many methodologies [7, 13, 15, 16, 23] on ontology learning from text data have been widely developed in recent years. Most of those researchers use artificial intelligence approaches such as machine learning or statistical analysis to develop the methodologies, and they try to extract the domain ontology knowledge from the text data semiautomatically. These existing ontology learning technologies are relatively handy and effective when conducting the learning tasks on English text data. However, due to the language dependence, the algorithms applied to English text data are found not working well and efficiently in Chinese text. This is based upon the structure of Chinese characters which are more complex and multivariate compared with an English word. And, as far as we know, there is not a successful Chinese text-based knowledge and information system developed and applied to the real applications. So, designing and constructing an effective and time-saving ontology learning system based on Chinese text data has the theoretical and practical values in commercial market.

Motivated by the demand of a high-performance ontology learning strategy to deal with Chinese text data and minimize the human intervention during the generation of domain ontology simultaneously, in this paper, we develop a comprehensive and innovative ontology extraction system called KnowledgeSeeker that represents the domain ontology with a graphical ontology structure—ontology graph. Focusing on the specific application domain for Chinese web text, KnowledgeSeeker can automatically generate the domain ontology graph (DOG) that is a knowledge conceptualization model depicted by our proposed ontology graph. Meanwhile, we also define another type of ontology graph named document ontology graph (DocOG) that is used to represent the content of a single text document. The main objective of extracting DocOG is to carry out the ontology graph-based text classification through matching the single document ontology (i.e., DocOG) with the domain ontology (i.e., DOG). Based on the collected data sets of web documents (totally contains 2,814 Chinese documents with an average of 965 Chinese characters in each document) from ten different domains, that is, 文藝 (Arts and Entertainments), 政治 (Politics), 交通 (Traffic), 教育 (Education), 環境 (Environment), 經濟 (Economics), 軍事 (Military), 醫療 (Health and Medical), 電腦 (Computer and Information Technology), and (Sports), thirteen different types of DOGs with different term sizes 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, and 300, respectively, are extracted and generated from every topic to represent the domain knowledge. Then, the generated DOGs are further used for the knowledge bases of ontology graph-based text classification. From Chinese News Web site (人民網, http://www.people.com.cn), we have collected a very large number of documents (57,218 documents) with an average of 2,349 Chinese characters in each news document that belong to ten distinct topic classes described above. We compare the performances of ontology graph-based text classification with two classical text classification methods—tf–idf [18] and term dependency [19]—in terms of precision [35] (measures the accuracy of the retrieval model), recall [10] (measures the ability of the retrieval model to retrieve correct documents from the whole data set), and *f*-measure [12] (measures the harmonic average of precision and recall). The experimental results show that the ontology graph-based text classification approach can remarkably outperform each of tf–idf and term dependency methods regarding the 3 above-mentioned indexes. The better performances of ontology graph-based text classification method reveal that the ontology graph learning method can successfully generate a set of smaller sized DOGs (30–80 term sizes) that are capable of representing the domain knowledge. And, the high performance given in the experimental result also demonstrates that KnowledgeSeeker can effectively handle the knowledge representation and management [37] with the presented ontological operations based on ontology graph theory.

The rest of this paper is organized as follows: In Sect. 2, we summarize the research background on which our methods are based upon. Section 3 discusses the detailed methodology of generating the proposed domain ontology graph model and applying this new model to Chinese-based information search and management system. Experimental simulations are carried out and the corresponding analyses of these empirical observations are presented in Sect. 4 Finally, in Sect. 5, we conclude this paper and outline the main directions for future research.

2 Background

2.1 Ontology

"Ontology" originates from the philosophy discipline, and it has been becoming a popular research topic in computer science [21, 45] and information system [11, 27, 43] fields. From the perspective of information science, ontology defines a set of representational primitives with which we can model the knowledge of a specific domain or discourse [3, 14, 21]. The representational primitives of an ontology contains the classes, attributes (properties), and relationships between classes. They are used to model knowledge of particular application domains. An ontological structure is to define how those components gather and construct together to represent a valid ontology. A 5-tuple based structure [3, 14] is a commonly used formal description to summarize the concepts and their relationships in a domain. The 5-tuple core ontology structure is defined as:

$$S = (C, R, H, \mathrm{rel}, A), \tag{1}$$

where C is the set of concepts describing objects, R a set of relation types, H a set of taxonomy relationship of C, $\mathrm{rel} \subseteq C \times C$ a set of relationship of C with relation type R, and A a set of description logic sentences.

The term rel is defined as a set of 3-tuple relations: $\mathrm{rel} = (s, r, o)$, standing for the relationship of subject-relation-object, where s is the subject element from C, r is the relation element from R, and o is the object element from C. In this 5-tuple ontological structure, knowledge is mainly represented by the logic sentences A, and the most important component is rel where it defines 3-tuple based concept relationship. A graphical representation of the 3-tuple structure is shown in Fig. 1, in which the subject s is defined as a node of source n_1, the object o defined as a node of target n_2, and the relation r defined as the association link between n_1 and n_2.

2.2 Ontology learning

Textual data are the most direct resource of human knowledge. Human beings write texts about what they perceive and think about the world, so it is a descriptive data that enable humans to share and exchange their knowledge. Although analyzing the textual data by computer is not a simple task, many methodologies on ontology learning from text [1, 6, 25] have been widely developed in recent years. Most of them use artificial intelligence approaches to develop the methodologies, and the semi-automatic text learning process is the goal of these researches. They use many artificial intelligence approaches such as information retrieval [23], machine learning [36], natural language processing [13], and statistical mathematics [15], to build the ontology learning system. However, the ontology learning outcome is sometimes not satisfactory to represent human knowledge [5, 8, 20, 26]. This is because the computational ontology is defined explicitly, but the knowledge in textual data is vague and implicit. There are difficulties to convert the implicit knowledge from text to a formalized ontology representation, in terms of both its quantity and quality. Quantity refers to that the ontology learning outcome is not comprehensive enough to express the whole knowledge domain, and it should have missed out some useful knowledge from the text. Quality refers to that the ontology learning outcome cannot express the relevant knowledge. In other words, from the standpoint of human understanding, the formalized knowledge from existing semi-automatic learning processes could be partly irrelevant or wrongly generated due to insufficient and inappropriate knowledge representation.

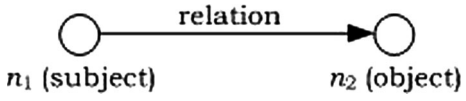


Fig. 1 Subject-relation-object representation

However, due to the fact that manual creating or maintaining the ontology is more time-consuming and inefficient, the semiautomatic ontology learning from text becomes a more practical and feasible scheme for formalizing ontology knowledge amid the rapid development of the ontology engineering tools. With the use of semi-automatic ontology learning method, the extracted ontology can serve for two main purposes: First, the ontology outcome can improve the performance of traditional information system by increasing the intelligent ability with embedded basic ontology knowledge. Although the embedded ontology is incomplete for the entire knowledge domain, it is still relevant to enhance the performance by artificial intelligence technology. Secondly, the ontology outcome can serve as an intermediate ontology or a base ontology for human to further develop and revise it. The incomplete or unsatisfied ontology can facilitate human beings to further develop a desired ontology for the knowledge domain, so that it is not required to build the entire ontology from scratch.

2.3 Text classification

With rapid growth of Internet technologies, a lot of web information is now available online. Information retrieval [29] such as text classification [4, 22, 32] on web data is so becoming a very important research area, as most web documents are created in the form of unstructured or semistructured text. A text classification system refers to

constructing a classifier in such instances, given a set of classes $C = \{c_1, c_2, \ldots, c_m\}$ and a document d, and determines the relevancy of class c_i in order to find out where document d belongs to. The classifier is a function $f_i(d) \rightarrow \{0, 1\}$ that expresses the relevancy value of the document d for the class c_i. A classical text classification model consists of documents as the input, processes with natural language processing, feature extraction, feature weighting, feature reduction, classification engine, and output to relevant classes or categories.

The traditional keyword-based text classification systems [22, 32] are mostly without many intelligent features, which always provide the inaccurate results in many practical applications. Intelligent text classification system applies the computational knowledge model, such as ontology, to enhance the classification algorithms. Ontology-based text classification approach improves the performance, in terms of its accuracy, over traditional approaches to gain effectiveness amid current information environment.

3 Methodology

In this section, we will describe how to generate the domain ontology graph (DOG) and conduct the text classification based on document ontology graph (DocOG).

3.1 Ontology learning framework—KnowledgeSeeker

KnowledgeSeeker is a comprehensive system framework which defines and implements four components; they are as follows: (1) ontology graph modeling (the ontology graph structure); (2) ontology learning (the learning algorithm); (3) ontology generation (the generation process); and 4) ontology querying (the operations for information retrieval system). The KnowledgeSeeker system can be used to develop various ontology-based intelligent applications by using the four defined ontological components. These intelligent applications include knowledge-based information retrieval system, knowledge mining system, personalization system, and intelligent agent system. Therefore, the entire KnowledgeSeeker system framework breaks down into four modules for handling different kinds of ontological process, as shown in Fig. 2.

3.2 Ontology graph modeling

The ontology graph is a novel approach used in KnowledgeSeeker system to model the ontology of knowledge in a domain (i.e., domain ontology graph, DOG) or in a single text document (i.e., document ontology graph, DocOG). The ontology graph consists of different levels of conceptual units, in which they are associated together by different kinds of relations. It is basically a lexicon system (terms) that links up each other to represent a group (a cluster), to formulate concepts and denote meanings. The conceptual structure of an ontology graph consists of many terms with some relationships between them, so that the different conceptual units are formed like a network model, as shown in Fig. 3.

Figure 4 further illustrates the conceptual view of the ontology graph model which is created based on the structure of term nodes and relations. The ontology graph

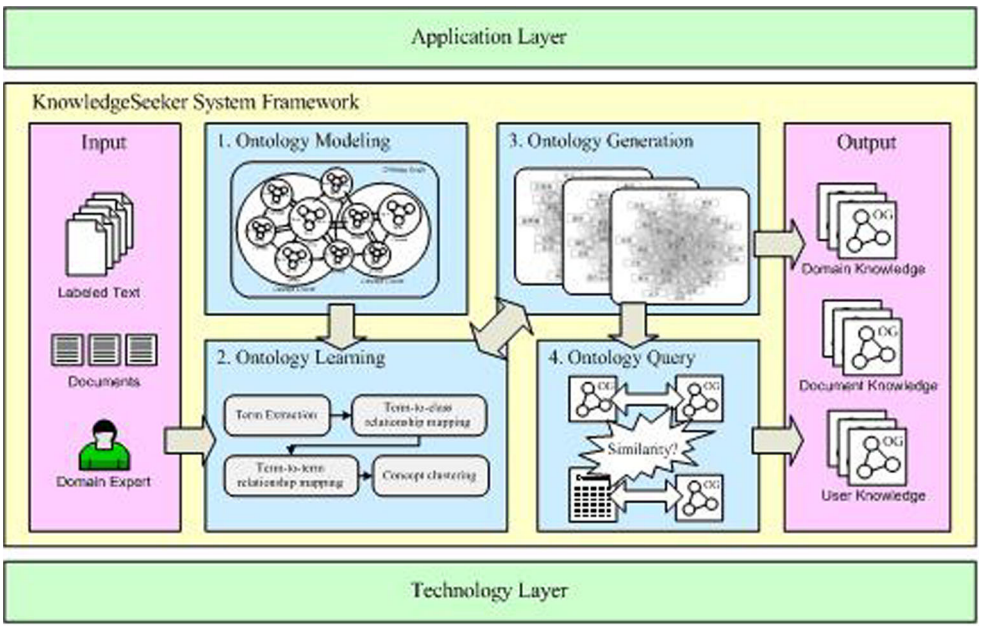


Fig. 2 The basic framework of KnowledgeSeeker system

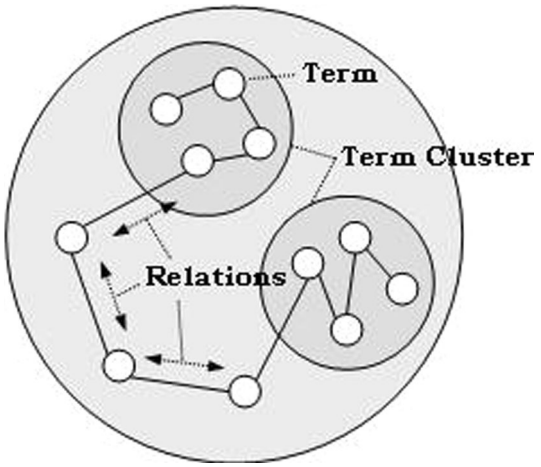


Fig. 3 Conceptual units

consists of four types of conceptual units (CU) according to their level of complexity exhibiting in knowledge. We define four CUs—any objects (nodes) in the ontology graph that gives semantics expression. All of these CUs are linked up and associated by conceptual relation (CR) within each other, to comprise the entire conceptual structure of ontology graph and to model an area (a domain) of knowledge.

The definitions of four CUs in ontology graph, their natures, and the levels of knowledge according to their complexity are described as follows:

1. Term. The smallest conceptual unit that extracted in the form of a meaningful word (a sequence of Chinese characters), those consist of "meaning" from the human perspective.
2. Concept. A number of terms grouped together with CR between each other form a concept. It is the basic conceptual unit in the concept graph.
3. Concept cluster. A number of concepts related to each other form a concept cluster. It groups similar meaning of concepts in a tight cluster representing a hierarchy of knowledge.
4. Ontology graph. The largest and entire conceptual unit grouped by concept clustering. It represents a comprehensive knowledge of a certain domain.

3.3 Ontology learning method in Chinese text

The ontology learning is the process to learn and create a domain of knowledge (a particular area of interest such as art, science, entertainment, sport, etc.) in the form of ontology graph, which is an ontology representation model described in Sect. 3.2. The ontology graph creation is considered as a knowledge extraction process. As described in Sect. 3.2, we defined different levels of knowledge objects, in the form of CU and CR, which are required for extraction in the learning process. We define a bottom-up learning approach to extract CU and CR and create the ontology graph. The approach identifies and generates CU from the lowest level, that is, term, to the highest level, that is, DOG.

We focused on the ontology learning in Chinese text, because the relationships between Chinese words are more difficult to be analyzed simply by grammar and word pattern (such as by regular expression) than English text. Therefore, we use Chinese text as the data source for learning and creating ontology graph which can reveal the

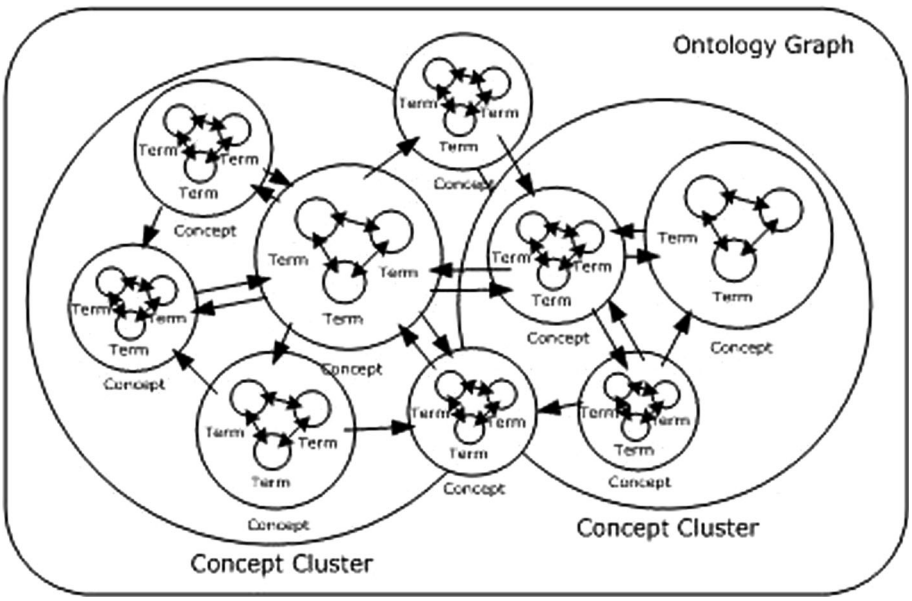


Fig. 4 The conceptual structure of ontology graph in KnowledgeSeeker

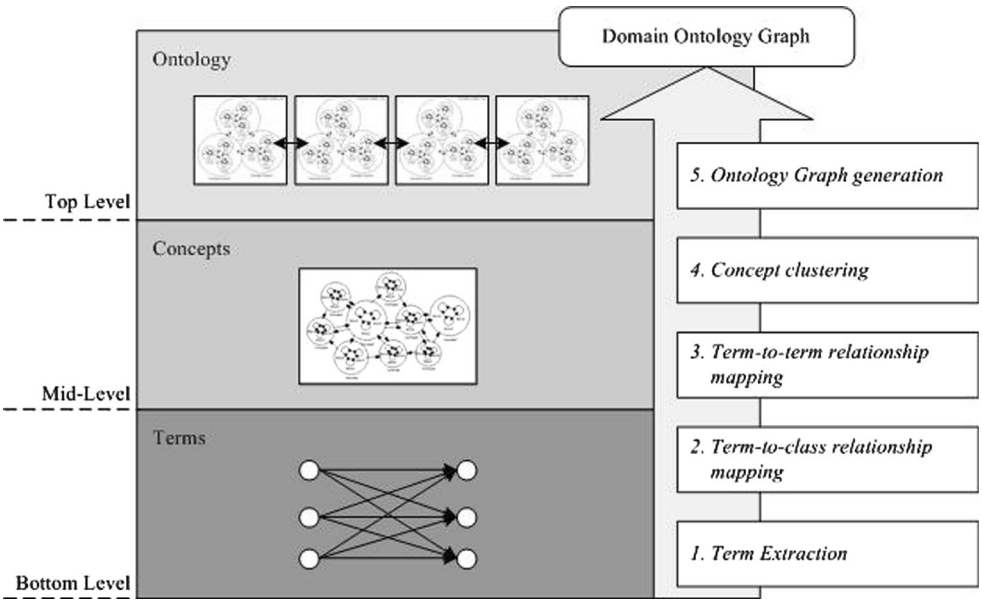


Fig. 5 The learning process of domain ontology graph

feasibility and effectiveness of learning ontology based on term relations, through the proposed learning approach.

The five learning subprocesses in the bottom-up learning approach are as follows:

1. *Term extraction* the most basic process that recognizes all meaningful Chinese terms in text documents.
2. *Term-to-class relationship mapping* the second process that finds out the relations between terms and classes.
3. *Term-to-term relationship mapping* the third process that finds out the relations between all Chinese terms within a class.
4. *Concept clustering* the fourth process that further groups (clusters) the Chinese terms within a class (domain) based on their similarity.
5. *Ontology graph generation* the final process that generates an ontology graph to represent the domain knowledge for specific application.

Figure 5 shows the detailed subprocesses in the bottom-up approach of DOG learning method. All of these subprocesses correspond to various CUs at different levels. Thus, the knowledge is learned from the smallest CU (i.e., term) toward the largest CU (i.e., DOG).

3.3.1 Term extraction

Our approach focuses on learning the domain ontology graph from Chinese text. Since there is no space among the characters in Chinese writing, a useful means of word disambiguation is not available in Chinese that is available in English. For this reason, Chinese term extraction typically relies on dictionaries. An existing electronic dictionary is available such as HowNet [9]. It contains over 50,000 distinct Chinese words, and it is useful for identifying the meaningful words inside a text, and it can constitute an initial term list for carrying out the term extraction process. By applying a maximal matching algorithm to the word list and a corpus of a set of Chinese texts, we can extract a candidate term list (a list of terms that are potentially associated with a relevant concept and thus to be extracted from the learning process), while filtered out other unnecessary terms that do not appear in the text corpus. n numbers of candidate terms $T = \{t_1, t_2, \ldots, t_n\}$ are thus extracted, where every term $t_i (i = 1, 2, \ldots, n)$ in the term list T appears at least once in the text corpus.

Besides the existing terms in the dictionary, an additional input of Chinese terms into the term extraction process is also required. These additional words, such as named person/organization, brand/building names, and new technologies, usually are not maintained in the dictionary since the dictionary is not keeping updates all the time. Therefore, adding new terms into the initial word list by human effort is also required.

3.3.2 Term-to-class relationship mapping

The candidate term list T extracted from the Chinese text corpus does not have any meaning and relationship to any conceptual units in the ontology graph model; thus, the second process that applied to the candidate term list is the term-to-class relationship mapping. This mapping process acts like feature selection that selects and separates every term in the term list from its most related domain class. First of all, we need to prepare a corpus of a set of labeled texts (a set of text documents which are classified into

different domains), and then, we can measure how the terms are related to each class. Finally, a sublist of terms is selected from the candidate term list for each class. The mapping process means that we give every term in the sublist associated with a class with a weighted and directed relation between a term and a class, as shown in Fig. 6a.

The term-to-class relationship mapping applies a χ^2 statistical term-to-class independency measurement to measure the degree of interdependency between a term and a class. The measurement is carried out by first calculating the co-occurrence frequencies between every term t and class c which is expressed as the observed frequency $O_{i,j}$ where $i \in \{t, \neg t\}$ and $j \in \{c, \neg c\}$. Thus, $O_{t,c}$ is the observed frequency (number) of documents in class c which contains the term t; $O_{t,\neg c}$ is the observed frequency of documents that are not in class c and contain the term t; $O_{\neg t,c}$ is the observed frequency of documents that are in class c and do not contain the term t; and $O_{\neg t,\neg c}$ is the observed frequency of documents that are neither in class c nor contain the term t. The observed frequency is compared to the expected frequency $E_{i,j}$ where $i \in \{t, \neg t\}$ and $j \in \{c, \neg c\}, E_{i,j}$ is defined as

$$E_{i,j} = \frac{\sum_{a\in\{t,\neg t\}} O_{a,j} \sum_{b\in\{c,\neg c\}} O_{i,b}}{N}, \tag{2}$$

where $N = O_{t,c} + O_{t,\neg c} + O_{\neg t,c} + O_{\neg t,\neg c}$ denotes the size of a classified document corpus.

Chi-square statistics independency measurement for term t and class c is defined as

$$\chi^2_{t,c} = \sum_{i\in\{t,\neg t\}} \sum_{j\in\{c,\neg c\}} \frac{\left(O_{i,j} - E_{i,j}\right)^2}{E_{i,j}}. \tag{3}$$

An alternative chi-square statistical-based word-class dependency measure defines $R_{t,c}$ as

$$R_{t,c} = \frac{O_{t,c}}{E_{t,c}}, \tag{4}$$

where $R_{t,c}$ is the ratio between $O_{t,c}$ and $E_{t,c}$. The term t is measured as positive dependency on class c if $R_{t,c} > 1$, or the term t is measured as negative dependency on class c if $R_{t,c} < 1$. $R_{t,c} = 1$ means that there is no dependency between t and c. In summary, $\chi^2_{t,c}$ measures the dependency between a term and a class in a data set of documents, while $R_{t,c}$ measures whether the dependency is positive or negative:

$$t \text{ is} \begin{cases} \text{positive dependency on } c & \text{if} \quad R_{t,c} > 1 \\ \text{negative dependency on } c & \text{if} \quad R_{t,c} < 1 \end{cases} \tag{5}$$

Algorithm 1 Calculate term-to-term dependency matrix

```
for every term vector v_i corresponding to class c_i do
  for every term t_a in v_i do
    for every term t_b in v_i do
      Calculate χ²_{t_a,t_b} according to Eq. (6);
      Calculate R_{t_a,t_b} according to Eq. (7);
      Normalize nχ²_{t_a,t_b} = χ²_{t_a,t_b} / χ²_{t_a,c_i};
    end for
  end for
end for
```

3.3.3 Term-to-term relationship mapping

The term-to-term relationship mapping is a further learning process that aims to calculate the inter-relationships between every term in the term list of a class (the term list of a class that has been created in the term-to-class relationship mapping process). In the term-to-class relationship mapping process, we find out the weighted relationship between a term and a class, but we do not know how those terms are related to each other inside the class. Therefore, the term-to-term relationship mapping further finds out and calculates this weighted relationship between those terms. We calculate and create a directed relation between the two terms, as shown in Fig. 6b.

To measure term-to-term relationship, we first select a certain number of terms in each class. In a real case, we determine a threshold k for the maximum number of highest ranked positive terms inside a term dependency vector of each class to represent the term group of the corresponding class for calculation:

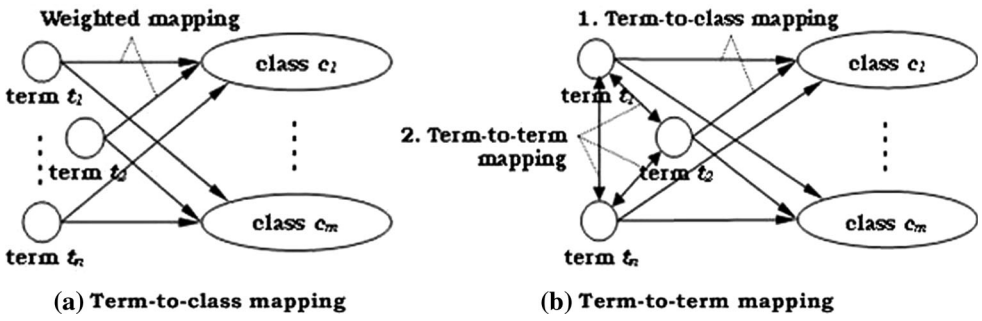


Fig. 6 The terms mapping

1. k-number of highest ranked positive terms in each class for there are m number of classes: $V = \{v_1, v_2, \ldots, v_m\}$ for each $v_i = \{(t_1, \chi^2_{t_1,c_i}, R_{t_1,c_i}), (t_2, \chi^2_{t_2,c_i}, R_{t_2,c_i}), \ldots, (t_k, \chi^2_{t_k,c_i}, R_{t_k,c_i})\}$ where $R_{t_j,c_i} = \frac{O_{t_j,c_i}}{E_{t_j,c_i}} > 1$ denotes the dependence of a term $t_j (t_j \in T, T = \{t_1, t_2, \ldots, t_k\})$ on a class $c_i (c_i \in C, C = \{c_1, c_2, \ldots, c_m\})$ is positive.
2. If the number of positive terms ($R_{t_j,c_i} > 1$) in a class is smaller than the threshold k, then we select all positive terms inside the class as the term group.

In the term-to-term relationship mapping process, we similarly apply chi-square statistical measurements of all the terms in the term group v_i of each class $c_i (c_i \in C, C = \{c_1, c_2, \ldots, c_m\})$. The equation for χ^2 statistics is modified to measure the independency between the two terms, instead of between a term and a class in the previous term-to-class mapping process. The co-occurrence frequency between the two terms t_a and t_b is the observed frequency $O_{i,j}$ where $i \in \{t_a, \neg t_b\}$ and $j \in \{t_b, \neg t_b\}$. Thus, $O_{ta,tb}$ is the observed frequency (number) of documents that contain term t_a as well as term t_b; $O_{t_a,\neg t_b}$ is the observed frequency of documents which contains term t_a but does not contain term t_b; $O_{\neg t_a,t_b}$ is the observed frequency of documents which does not contain term t_a but contains the term t_b; and $O_{\neg t_a,\neg t_b}$ is the observed frequency of documents which does not contain both terms t_a and t_b.

Hence, the Eqs. (3) and (4) given in Sect. 3.3.2 can be changed to Eqs. (6) and (7), respectively, to measure the dependency between t_a and t_b:

$$\chi^2_{t_a,t_b} = \sum_{i \in \{t_a, \neg t_a\}} \sum_{j \in \{t_b, \neg t_b\}} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}, \tag{6}$$

$$R_{t_a,t_b} = \frac{O_{t_a,t_b}}{E_{t_a,t_b}}. \tag{7}$$

$\chi^2_{ta,tb}$ measures the dependency between terms t_a and t_b in a data set of documents, while $R_{ta,tb}$ measures whether this dependency is positive or negative. Algorithm 1 presents the detailed procedure of calculating $\chi^2_{ta,tb}$ of a class $c_i (c_i \in C, C = \{c_1, c_2, \ldots, c_m\})$. Therefore, after the term-to-term relationship mapping and normalization process, we can obtain two $k \times k$ term-to-term dependency matrices containing the values of $n\chi^2_{t\_a,t\_b}$ and R_{t_a,t_b} which represent the relationship of every term-to-term pair within a class $c_i (c_i \in C, C = \{c_1, c_2, \ldots, c_m\})$.

The term independence representations can be converted into a directed graph $G = (V, A)$, where V is the set of selected terms, $V = \{t_1, t_2, \ldots, t_{k-1}, t_k\}$; and A is the set of directed and weighted edge, $A = \{(t_i, t_j, n\chi^2_{t_i,t_j}) | i \in \{1, 2, \ldots, k\}, j \in \{1, 2, \ldots, k\}\}$. Whether the edge between the two terms t_i and t_j should be represented in the directed graph G depends on the value of R_{t_i,t_j}. When $R_{t_i,t_j} > 1, i \in \{1, 2, \ldots, k\}, j \in \{1, 2, \ldots, k\}$ which denotes the dependency between the two terms t_i and t_j is positive, in this case, there is a visual link created between these two terms in graph G; otherwise, when $R_{t_i,t_j} < 1, i \in \{1, 2, \ldots, k\}$, $j \in \{1, 2, \ldots, k\}$, which denotes the dependency between the two terms t_i and t_j is negative, then the associated link between these two terms will not be represented in the directed graph G. For example, if $k = 4$ and all $R > 1$ which means the dependency relation between any two terms is positive, the directed graph is created as in Fig. 7. The vertices in this graph are the top $k = 4$ terms in the class c_1, and the edges in this graph are the directed and weighted links between the two terms if their dependency relation is positive. In Fig. 7, the term-to-class relationship vector of class c_1 is $v_{c\_1} = \{(t_1, 0.476, 1.667), (t_2, 10.3.333), (t_3, 0.476, 1.667), (t_4, 0.476, 1.667)\}$.

3.3.4 Concept clustering

The concept clustering is the process of grouping semantically similar terms into a tight semantic group. The graph created in the previous step is the base input for the concept clustering process. The idea is to group terms with high weighted relations into a subgraph while separating out other terms to create a new subgraph of low weighted relations. Clusters are automatically created without explicitly defining the number of clusters which need to be created. The highest weighted edge e_x with two vertices t_a and t_b is firstly grouped together to form an initial cluster. We then select the next highest weighted edge e_y with the next two vertices t_c and t_d. If the next selected vertices are linked by any vertices from the existing cluster, the vertices are put into that cluster. Otherwise, a new cluster is formed with the inclusion of the selected vertices. Algorithm 2 gives the detailed term-to-concept clustering procedure.

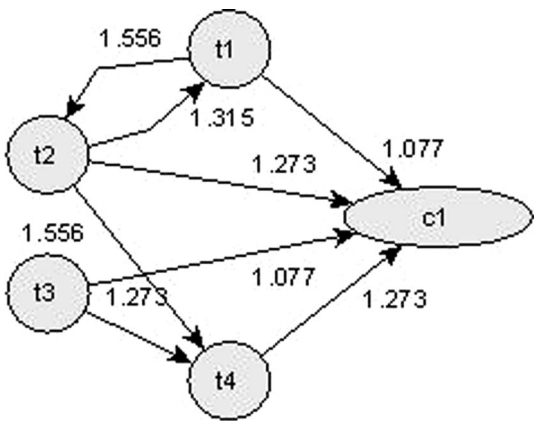


Fig. 7 The ontology graph containing 4 terms in class c_1

Algorithm 2 Terms-to-concept clustering

for every DOG **do**
  Select the highest weighted edge $e_x = \{(t_a, t_b, R_{t_a,t_b})\}$ in vector V;
  Create the first concept cluster containing t_a and t_b;
  for every edge e in the edge vector V **do**
    Select the next highest weighted edge $e_x = \{(t_c, t_c, R_{t_c,t_d})\}$;
    if t_c or t_d in the existing cluster **then**
      Put both t_c and t_d into that existing cluster;
    else
      Create a new cluster containing t_c or t_d which does not appear in existing clusters;
    end if
  end for
end for

The concept clustering, therefore, creates a second taxonomical relationship. The first layer of concept hierarchy is created in term-to-class relationship mapping, where all the terms related to that class are now further clustered and form a second layer of hierarchy. That is, every cluster creates a relation to its related class as a parent, relations to all their contained terms as children, and finally a three-level taxonomical hierarchy of concepts inside the DOG.

Algorithm 3 DOG generation

Obtain the term list $T = \{t_1, t_2, \cdots, t_k\}$ from domain d, where k is a predefined parameter which denotes the maximum number of terms in the generated DOG;
for every term t in T **do**
  Generate a node n_i in DOG;
  Assign the node label according to the term name;
  Assign the node weight according to the term-to-class dependency $\chi^2_{t,d}$;
end for
Obtain the term dependency values of term-list T and set the value of θ which denote the minimum dependency value in the generated DOG;
for every term-term (e.g., t_a-t_b) dependency mapping **do**
  if the dependency value between t_a and t_b is grater than or equal to θ **then**
    Generate an edge e_i between t_a and t_b in DOG;
    Associate with the two ends of e_i to the nodes t_a and t_b;
    Set the edge weight to the term-to-term dependency value $\chi^2_{t_a,t_b}$;
  end if
end for
Remove all unlinked nodes in DOG;

3.3.5 Domain ontology graph generation

In KnowledgeSeeker, we define ontology graph to model a set of concepts. Concepts are created by the set of terms and relations between them. The relations of terms are enhanced by weights, which are generated automatically by the statistical learning method as presented in Sect. 3.3.2. In the following, we formalize the definition of DOG.

Definition 3.1 The DOG in KnowledgeSeeker system is an ontology graph associated with the specific domain. It can be defined as:

$$OG_d = (T, F, H, R, C, A), \tag{8}$$

where d defines the domain in which the ontology graph is associated with; T is a set of terms t_i of OG_d; F a set of word functions of terms $t_i \in T$; H a set of taxonomy relationship of T; R a set of relations between t_i and t_j, where $t_i, t_j \in T$; C a set of clusters of $t_1, \ldots, t_n$, where $t_1, \ldots, t_n \in T$; and A a set of axioms that characterizes each relation of R.

The DOG is created from a large classified Chinese corpus in the ontology learning process. The generation is a semiautomatic process. The manual processes include defining the initial term list (can be obtained from existing dictionary), defining and mapping the types of word function (also may be obtained from that dictionary), and labeling the concept clusters. The semiautomatic processes include the domain terms extraction, terms relationship extraction, and concept cluster extraction. Algorithm 3 shows the steps of DOG generation.

Algorithm 4 DocOG generation

Input: A text document and a generated DOG-OG_d for domain d;
Output: A DocOG (OG_{doc}) representing the input document *doc*;
Obtain the document content represented by a collection of terms $T = \{t_1, t_2, \cdots, t_k\}$;
Obtain the term set T_d and relation set R_d of a DOG $OG_d = (T_d, F_d, H_d, R_d, C_d, A_d)$;
Transform the document to weighted term vector: $V_{doc} = \{(t_1, w_{t_1}), (t_2, w_{t_2}), ..., (t_k, w_{t_k})\}$, where $w_{t_i}, t_i \in T$ denotes the weight of term t_i in document *doc*. (The weight w_{t_i} is defined as: $w_{t_i} = \frac{tf_i}{dl}$, where tf_i is the frequency of term t_i appearing in the document *doc* and dl is the length of document *doc*, i.e. the size of the term list of document *doc*.)
Ontology Graph matching for creating the term set T_{doc} and relation set R_{doc} of DocOG (OG_{doc});
for every term t_i in V_{doc} **do**
  if term t_i exits in T_d of OG_d **then**
    for every related term t_j in R_d of OG_d **do**
      Add both t_i and t_j to the term set T_{doc} of OG_{doc};
      Add the relation to the relation set R_{doc} of OG_{doc} for the terms t_i and t_j;
      Calculate the weight w_{t_i,t_j} between terms t_i and t_j according to $w_{t_i,t_j} = w_i \times w_j$;
    end for
  end if
end for
Ontology Graph Generation for the document with the created T_{doc} and R_{doc} of OG_{doc};

3.4 Document ontology graph generation

DocOG is another type of ontology graph which is used to represent the content of a single text document. Traditional

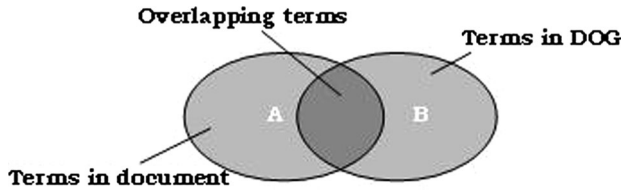


Fig. 8 Mapping overlapping terms in document and DOG

information retrieval system usually represents documents by term vectors. In KnowledgeSeeker system, we proposed to use the ontology graph model to represent the content of a text document. In addition, the DocOG can also describe more information about the document, such as the related knowledge of a certain domain. This can be done by matching the text document to a DOG to acquire more knowledge about the related domain.

The matching of a text document to a DOG aims at extracting more knowledge about the domain inside the document. Text document is often represented by a list of terms (a weighted term vector). We match the term 'vector' to a DOG to create mappings between them if they have the intersection of same terms (Fig. 8). This process can relate a document to a particular domain. The process can be used to generate DocOG for the original document and also measure the similarity of the document to the matched DOG. Algorithm 4 describes the detailed process of DocOG generation.

3.5 Ontology graph-based text classification

Ontology graph-based text classification is done by measuring the similarity between a DocOG (representing a document) and a DOG (representing a domain). This is to find out how the document is related to the domain for comparison, which is useful in text classification process. When a DocOG is compared to more than one DOG, the highest scored DOG in the result is the domain that the document is mostly related to. Figure 9 depicts the components of DOG and DocOG comparison process. And, Algorithm 5 summarizes the detailed process of ontology graph-based text classification.

Algorithm 5 Ontology graph based text classification

1: **Input:** A DocOG representing the input document and a DOC representing the domain knowledge;
2: **Output:** A similarity score representing the comparison result;
3: Generate corresponding DOG for every domain. If there are m number of classes of domain to be classified, m number of DOG are created;
4: Generate m number of DocOG correspondingly to each DOG for a single text document;
5: Obtain the score of each DocOG by summing up all the weights of relations, excluding that of self-relations (the weight of the term itself);
6: Select the highest scored domain as the classified result.

3.6 Example of comparison between DocOG and DOG

In this section, we will give an example to show the matching process of DOG and DocOG. For simplification, the tabular form and graphical representation of DOG in domain d (i.e., DOG_d containing 5 terms-A, B, C, D, E) are presented in Table 1 and Fig. 10, respectively. The information of two DocOGs, that is, DocOG_{d_1} and DocOG_{d_2} corresponding to two different documents d_1 and d_2 are also summarized in Table 2 and Fig. 11 and Table 3 and Fig. 12.

We firstly compute the document score of each DocOG by summing up all the relations as following expressions (excluding the weight of self-relations):

$$\mathrm{score}(\mathrm{DocOG}_{d_1}, \mathrm{DOG}_d) = 1.425;$$
$$\mathrm{score}(\mathrm{DocOG}_{d_2}, \mathrm{DOG}_d) = 2.275.$$

Then, the similarity between DocOG and DOG can be obtained by finalizing the document scores:

$$\mathrm{sim}(\mathrm{DocOG}_{d_1}, \mathrm{DOG}_d) = \frac{\mathrm{score}(\mathrm{DocOG}_{\mathrm{d}_1}, \mathrm{DOG}_{\mathrm{d}})}{\mathrm{score}(\mathrm{DOG}_d)} = \frac{1.425}{5} = 0.285;$$
$$\mathrm{sim}(\mathrm{DocOG}_{d_2}, \mathrm{DOG}_d) = \frac{\mathrm{score}(\mathrm{DocOG}_{\mathrm{d}_2}, \mathrm{DOG}_{\mathrm{d}})}{\mathrm{score}(\mathrm{DOG}_d)} = \frac{2.275}{5} = 0.455.$$

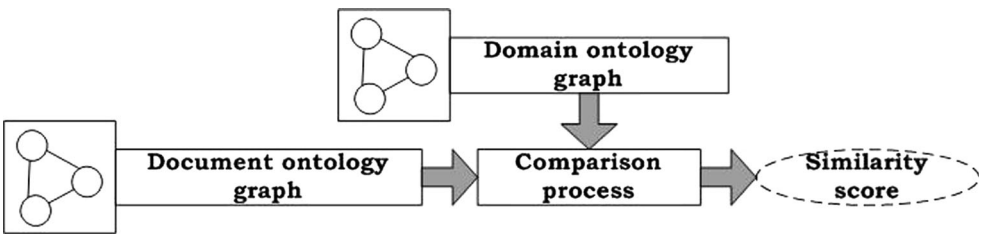


Fig. 9 Components of DOG and DocOG comparison process

Table 1 Relations in DOG_d

	A	*B*	*C*	*D*	*E*
A	1	0.1	0.2	0.3	0.4
B	0.1	1	0.2	0.3	0.4
C	0.1	0.2	1	0.3	0.4
D	0.1	0.2	0.3	1	0.4
E	0.1	0.2	0.3	0.4	1

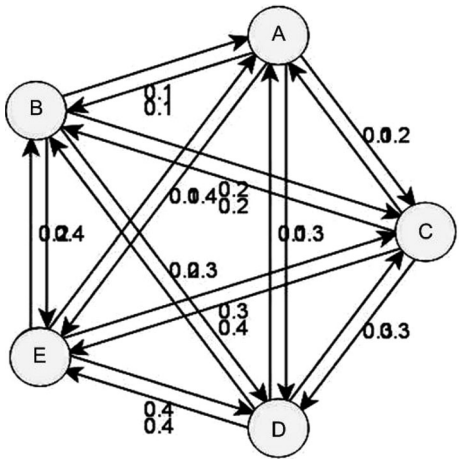


Fig. 10 DOG_d

Table 2 Relations in $DocOG_{d_1}$

	A	*B*	*C*	*D*	*E*
A	0.5	0.05	0.1	0.15	0.2
B	0.05	0.25	0.05	0.075	0.1
C	0.05	0.05	–	0.075	–
D	0.05	0.05	0.075	0.25	0.1
E	0.05	0.05	–	0.1	–

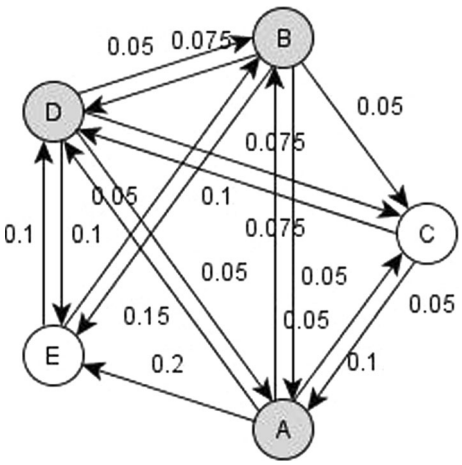


Fig. 11 $DocOG_{d_1}$

Table 3 Relations in $DocOG_{d_2}$

	A	*B*	*C*	*D*	*E*
A	–	–	–	0.225	0.1
B	–	–	–	0.225	0.1
C	–	–	–	0.225	0.1
D	0.075	0.15	0.225	0.75	0.3
E	0.025	0.05	0.075	0.3	0.25

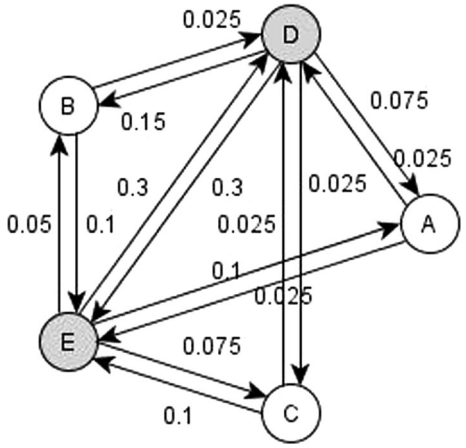


Fig. 12 $DocOG_{d_2}$

4 Experimental results

4.1 Experimental data set

The source of knowledge serves for the data prepared for domain ontology graphs generation. In the entire experiment, from the learning source to testing document, all of the involved documents are written in Chinese. First of all, we need a classified Chinese document corpus for both learning and testing purposes.

The document corpus (D_1) contains 2,814 Chinese documents with an average of 965 Chinese characters in each document. The articles belong to ten different topic classes, which are 文藝 (Arts and Entertainments), 政治 (Politics), 交通 (Traffic), 教育 (Education), 環境 (Environment), 經濟 (Economics), 軍事 (Military), 醫療 (Health and Medical), 電腦 (Computer and Information Technology), and 體育 (Sports). These ten topics are labeled as the classes for the domain ontology learning processes. The documents of the corpus in every class are further divided to allow 70 % of them for the learning set (D_1-Learn) and 30 % for the testing set (D_1-Test). The document distributions (including total document count, the size of training set, and the size of testing set) of the ten classes are shown in Table 4. We use only the 70 % classified documents (1,972 documents) for the process of the term extraction and term-to-class relationship mapping.

Table 4 The detailed document distribution of corpus D_1

Class	Number of documents	D_1-*Learn* (70 %)	D_1-*Test* (30 %)	Number of positive terms	Number of negative terms
文藝	248	174	74	867	29281
政治	505	354	151	966	37481
交通	214	150	64	769	34691
教育	220	154	66	904	30604
環境	201	141	60	788	32823
經濟	325	228	97	664	35680
軍事	249	174	75	727	33439
醫療	204	143	61	862	35527
電腦	198	139	59	774	30671
體育	450	315	135	956	37061

The results of the number of positive and negative terms in these ten classes are also shown in Table 4.

There is another Chinese document set from an unclassified warehouse (D_2). It is used for the process of term-to-term relationship mapping and concept clustering. The unclassified warehouse D_2 contains a relatively large amount of documents (57,218 documents) which are collected from a Chinese News Web site (人民網, http://www.people.com.cn), with an average of 2,349 Chinese characters in each news document.

4.1.1 *Samples of ontology graph generation from ten different domains*

Figure 13 visualizes the generated ontology graphs from ten different domains. The learning process selects the top 30 positive terms in each class ($k = 30$ in Algorithm 3) to generate the corresponding DOG. These pictures only visualize the terms and their relationships. The corresponding list of English translation of the 30 Chinese terms in each domain is provided in Fig. 14.

From Algorithm 3, we can see that two predefined parameters, that is, the maximum number of terms k and minimum term dependency θ, which will influence the generation of DOG. To take 文藝 for an example, we select the top 20 terms in this domain to generate DOGs with different parameters so that the process of DOG generation can be understood more clearly. These top 20 terms of domain 文藝 are listed in Table 5. And, the graphical results of generated DOGs are visualized in Fig. 15.

4.2 Performance of the ontology graph-based text classification

The text classification experiment described here has two purposes. On the one hand, we wish to evaluate the classification performance of the proposed ontology graph-based classification by comparing it with other classification approaches. On the other hand, we wish to determine the optimal size, the number of terms in DOG for a class that can produce the best classification result.

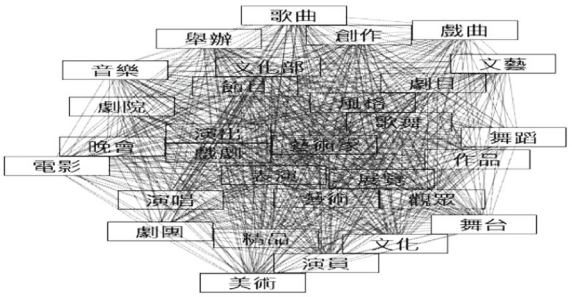
4.2.1 *Experiments description*

The designed experiments consist of two parts: testing the performance of ontology graph-based approach (experiment I) and evaluating the optimal parameters (i.e., k and θ) of DOG for best classification result (experiment II).

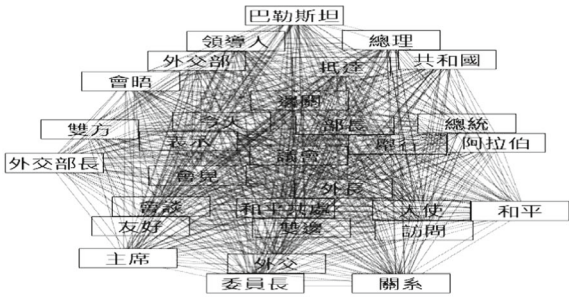
The first experiment presents a text classification comparison by using three different approaches to classify documents. The first is the traditional tf–idf approach [18]. The second is the term dependency approach [19] which replaces the tf–idf weight by the term dependency weight in DOG. The third is the ontology graph-based approach which scores a document (which is represented as DocOG) to a class by the weight of relationships between each concept in the ontology graph. We aim to evaluate and compare the performance of different text classification approaches in terms of their accuracies (e.g., precision, recall, and f-measure). The three different text classification approaches are introduced as follows:

1. Term frequency–inverse document frequency (tf–idf) approach. This approach uses a scoring function that scores the terms occurred in the document by the term frequency and the inverse document frequency. This scoring function is the same as the traditional tf–idf classification approach [18], and it is defined as:
$$\text{score}(t_i) = \text{tf}_{t_i} \times \text{idf}_{t_i}; \tag{9}$$
2. Term dependency approach. This approach [19] uses a scoring function that scores the terms occurred in the document by the term weight in the DOG. Term weights in the ontology graph are represented by the dependency measurement-R, and it is calculated in the DOG learning process. In our study, the term dependency scoring function is defined as:
$$\text{score}(t_i) = \text{tf}_{t_i} \times R_{t_i}; \tag{10}$$
3. Ontology graph approach. The ontology graph-based text classification approach is processed by measuring the similarity between DOG and the corresponding DocOG as described in Sect. 3.5. Therefore, the scoring function for comparing a document to a domain is defined as:
$$\text{score}\left(\text{doc}, \text{DOG}_j\right) = \text{score}\left(\text{DocOG}, \text{DOG}_j\right), \tag{11}$$

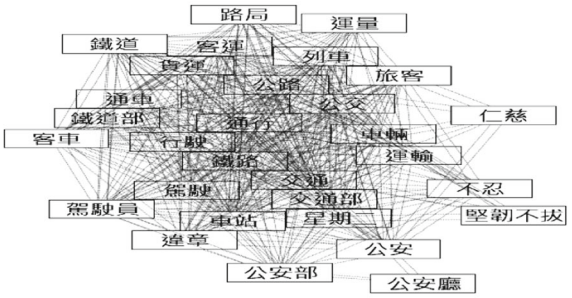
where term $t_j \in T, T = \{t_1, t_2, \ldots, t_n\}$; doc represents the input document, and $DocOG$ is the document ontology graph of doc; DOG_j is the domain ontology graph for the j-th domain, $j = 1, 2, \ldots, m$.



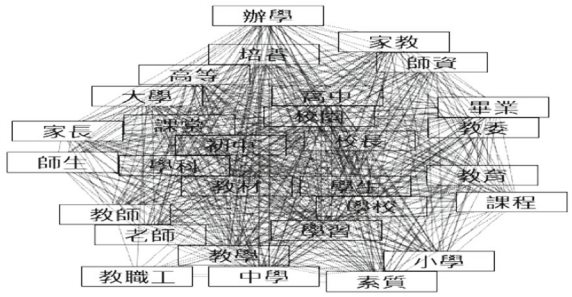
(a) DOG of Arts and Entertainments (文藝)



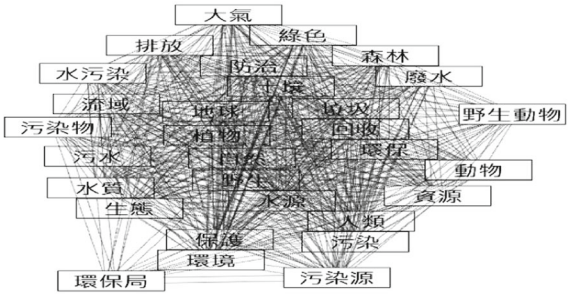
(b) DOG of Politics (政治)



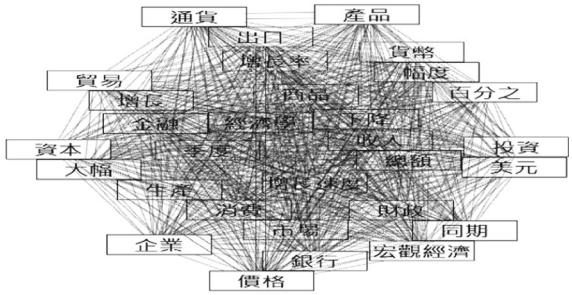
(c) DOG of Traffic (交通)



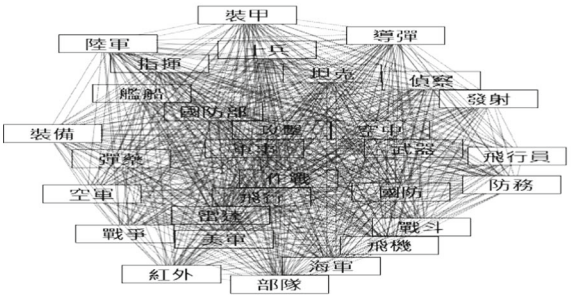
(d) DOG of Education (教育)



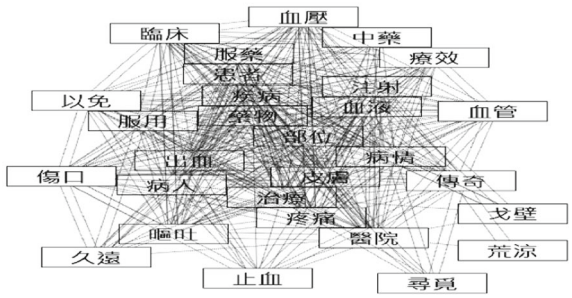
(e) DOG of Environment (環境)



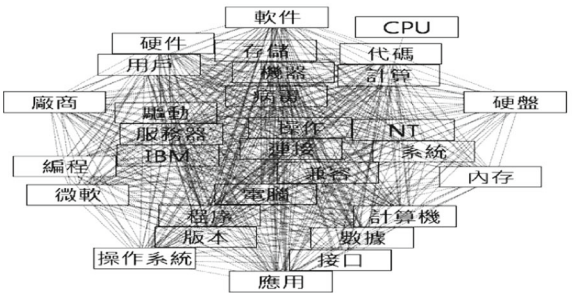
(f) DOG of Economics (經濟)



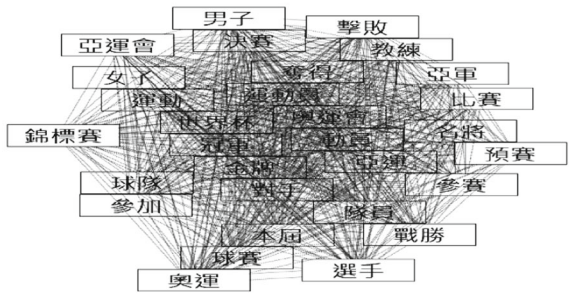
(g) DOG of Military (軍事)



(h) DOG of Health and Medical (醫療)



(i) DOG of Computer and Information Technology (電腦)



(j) DOG of Sports (體育)

Fig. 13 The ontology graphs generated from ten different domains

創作	Creation	總統	President	資源	Resources	生產	Manufacturing
藝術	Art	訪問	Visit	垃圾	Rubbish	同期	The corresponding period
演出	Perform	主席	Chairwoman	水源	Source of water	增長率	Rate of increase
作品	Works	外交	Diplomatic	動物	Animal	資本	Capital
觀眾	Audience	會見	Meet with	水污染	Water pollution	經濟學	Economics
藝術家	Artist	友好	Friendly	野生動物	Wild animals	貿易	Trade
文化	Culture	外長	Foreign minister	流域	Valley	價格	Price
演員	Actor	總理	Prime minister	人類	Humanity	季度	Quarterly
劇團	Troupe	會談	Talk	地球	The earth	幅度	Range
節目	Programme	外交部	Ministry of foreign affairs	水質	Water quality	貨幣	Currency
音樂	Music	和平	Mild	土壤	Soil	增長速度	Speed of increase
歌舞	Sing and dance	關系	Relationship	綠色	Green	總額	Sum total
劇院	Theatre	議會	Parliament	回收	Recovery	宏觀經濟	Macro economy
晚會	Evening party	領導人	Leader	防治	Do prevention and cure	通貨	Currency
戲劇	Drama	今天	Today	植物	Plant	大幅	Large-scale
文化部	Ministry of Culture	部長	Head of a department	武器	Weapon	治療	Treatment
舞蹈	Dance	雙邊	Bilateral	作戰	Fight	病人	Patient
文藝	Literature and art	雙方	The two parties	戰斗	Militant	藥物	Medicines
舉辦	Hold	表示	Express	美軍	U.S. Army	醫院	Hospital
表演	Performance	阿拉伯	Arabic	導彈	Missile	患者	Patient
舞台	Arena	會晤	Meet	海軍	Navy	療效	Curative effects
電影	Movie	邊關	Frontier pass	部隊	Troop	踟躕	Hesitate
歌曲	Song	抵達	Arrive	飛行	Flight	久遠	Far back
戲曲	Traditional opera	大使	Ambassador	艦船	Naval vessel	尋覓	Seek
劇目	A list of plays or operas	委員長	Head of committee	國防	National defense	戈壁	Desert
精品	Fine work	舉行	Hold	發射	Shoot	荒涼	Bleak and desolate
美術	Painting	巴勒斯坦	Palestine	坦克	Tank	傳奇	Mythical
風格	Manner	共和國	Republic	空軍	Air force	皮膚	Skin
演唱	Sing in a performance	外交部長	Minister for foreign affairs	雷達	Radar	血壓	Blood pressure
展覽	Exhibition	和平共處	Peaceful coexistence	裝備	Equipment	血液	Blood
運輸	Transport	教師	Teacher	陸軍	Army	疼痛	Pain
鐵路	Railway	學校	School	軍事	Military	嘔吐	Vomit
公路	Highway	教學	Teaching	偵察	Reconnoiter	服用	Take
車輛	Cars	學生	Student	國防部	Ministry of National Defense	疾病	Disease
交通	Traffic	辦學	Run a school	裝甲	Armored	血管	Blood vessel
公交	Public traffic	中學	Secondary school	攻擊	Attack	出血	Bleed
旅客	Passenger	培養	Training	指揮	Command	以免	In order to avoid
列車	Train	教育	Education	戰爭	War	病情	State of an illness
不忍	Cannot bear to	素質	Quality	士兵	Privates	臨床	Clinical
客運	Passenger transport	小學	Primary school	飛機	Aircraft	注射	Injection
堅韌不拔	Persistently	校長	Principal	彈藥	Ammunition	傷口	Wound
仁慈	Kindly	師資	Persons qualifies to teach	空中	In the sky	服藥	Take medicine
客車	Bus	校園	Campus	防務	Defense	止血	Stop bleeding
交通部	Ministry of Communications	高中	Senior middle school	红外	Infra-red	部位	Position
行駛	Travel	課程	Course	飛行員	Pilot	中藥	Traditional chinese medicine
運量	Freight volume	畢業	Graduate	軟件	Software	比賽	Competition
公安	Police	教材	Teaching material	用戶	User	冠軍	Champion
貨運	Freight transport	家教	Family education	程序	Program	選手	Player
鐵道	Railway	家長	Parent	計算機	Computer	決賽	Finals
公安部	Ministry of Public Security	學習	Study	硬盤	Hard disk	女子	Woman
星期二	Tuesday	課堂	Classroom	操作系統	Operating system	運動員	Sportsman
駕駛員	Driver	德育	Moral education	服務器	Server	亞運會	Asian games
通車	Be open to traffic	大學	University	微軟	Microsoft	亞運	Asian games
駕駛	Drive	初中	Junior middle school	接口	Interface	金牌	Gold medal
鐵道部	Ministry of Railway	老師	Teacher	版本	Edition	錦標賽	Championship
公安廳	Public security department	高等	Higher	兼容	Compatible	隊員	Team member
路局	Railway bureau	教委	State education commission	應用	Application	男子	Man
違章	Break rules and regulations	教職工	Teaching and administrative staff	計算	Compute	奪得	Compete for
通行	Have free passage	師生	Teacher and student	CPU	CPU	運動	Sports
車站	Station	學科	Discipline	內存	Inner memory	動員	Arouse
污染	Pollution	增長	Growth	硬件	Hardware	教練	Coach
生態	Ecology	出口	Export	操作	Operation	球隊	Team
環保	Environmental protection	企業	Enterprise	NT	NT	亞軍	Runner-up
保護	Protection	收入	Revenue	IBM	IBM	參賽	Participate in a match
森林	Forest	市場	Marketplace	機器	Machine	本屆	Current
排放	Discharge	銀行	Bank	數據	Data	戰勝	Defeat
污染物	Pollutant	財政	Finance	廠商	Factories and stores	世界杯	World cup
廢水	Liquid waste	美元	Dollar	存儲	Storage	奧運	Olympic games
大氣	Atmosphere	金融	Finance	驅動	Drive	球賽	Match
環境	Environment	消費	Consume	病毒	Virus	奧運會	Olympic games
環保局	State Bureau of Environmental Protection	產品	Product	系統	System	名將	Famous general
污染源	Pollution source	投資	Invest	代碼	Code	參加	Join
自然	Nature	下降	Descent	編程	Program	對手	Competitor
野生	Wild	百分之	Per cent	連接	Link	擊敗	Beat
污水	Waste water	商品	Goods	電腦	Computer	預賽	Trial match

Fig. 14 The English translation of Chinese terms

The second experiment presents an extended text classification case by using those three different approaches presented in experiment I and further varying the size of terms used in each approach and the parameter θ for ontology graph approach. The process used the same scoring functions presented in experiment I but with

Table 5 Top 20 terms in domain 文藝

Rank	Term	χ^2	R
1	人民網 (arts)	1014.18	7.552083
2	作品 (works)	979.3634	8.992248
3	創作 (creative)	975.1136	9.478261
4	演出 (perform)	748.1122	8.495575
5	文藝 (literature)	688.607	8.47619
6	觀眾 (audience)	666.3134	8.585859
7	文化 (culture)	585.201	5.131086
8	藝術家 (artist)	572.6829	9.305556
9	畫 (draw)	512.5542	6.27451
10	演員 (actor)	411.4805	9.090909
11	戲劇 (opera)	392.7607	9.387755
12	音樂 (music)	386.0206	6.542056
13	節目 (show)	357.0295	7.605634
14	舞台 (stage)	349.994	7.846154
15	表演 (act)	343.3402	6.595745
16	美術 (painting)	330.3707	7.051282
17	風格 (style)	329.2021	8.6
18	舉辦 (hold)	329.0934	5.125
19	劇團 (troupe)	327.732	9.5
20	歌舞(sing)	318.0413	9.069767

different sizes of the term vector (for if-idf and term dependency approaches) or DOG (for ontology graph approach) of each class to achieve the text classification. In this experiment, we aimed to evaluate how the size of terms in each approach and parameter θ for ontology graph approach affect the classification performance.

4.2.2 Evaluation methods

Error rate is the most practical measurement to evaluate the text classification performance. This measurement is aimed to calculate the classification accuracy, in terms of precision, recall, and *f*-measure. It is achieved by first observing the classification correctness from the result:

1. Precision [35]: It measures the accuracy of the retrieval model, by calculating the percentage of correctly retrieved documents to the whole retrieved result set. It is defined by:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \tag{12}$$

2. Recall [10]: It measures the ability of the retrieval model to retrieve the correct documents from whole data set, through calculating the percentage of correctly retrieved documents to all the documents which should be retrieved. It is defined by:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \tag{13}$$

3. *F*-measure [12]: It measures the harmonic average of precision and recall. It is defined by:

$$f\text{-}measure = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{14}$$

where TP (true positive) is the number of relevant documents, retrieved as relevant; FP (false positive) the number of relevant documents, not retrieved as relevant; and FN (false negative) the number of irrelevant documents, retrieved as irrelevant.

4.2.3 Result of ontology graph-based text classification

Table 6 shows the detailed results obtained by computing precision, recall, and *f*-measure for the three approaches using a term size of 30, that is, 30 of terms are used in the tf–idf method and 30-term sized DOG are used in the term dependency and ontology graph method. The table summarizes the precision, recall, and *f*-measure of each class in the testing document set, and also its average. The above experimental result has shown that the ontology graph approach obtains the highest classification accuracy (89.2 % of *f*-measure) and the term dependency method has the second highest classification accuracy (87.2 % of *f*-measure), while the tf–idf performs the lowest classification accuracy (84.8 % of *f*-measure) among the three methods that have been tested. This experiment has shown that the DOGs are useful to represent a domain of classes, and also it is useful to develop a classification system. By comparing to the term dependency method, it revealed that the relationship of concepts in the ontology graph is useful for representing knowledge. This is because using the relationship information in DOG (ontology graph approach) performs better results of text classification rather than not using the relationship (term dependency approach) in the first place. Therefore, this concludes that the ontology graph approach is effective for developing a text classification system.

4.2.4 Results of using different size of terms for text classification

In the previous experiment, we have found that the ontology graph-based approach performs the best in text classification among all three tested approaches. In this experiment, we further evaluate those three methods by varying the size of terms (the number of term nodes in DOG) used in the text classification process. The precision, recall, and *f*-measure values have been computed for this experiment by using different sizes of term nodes of DOG.

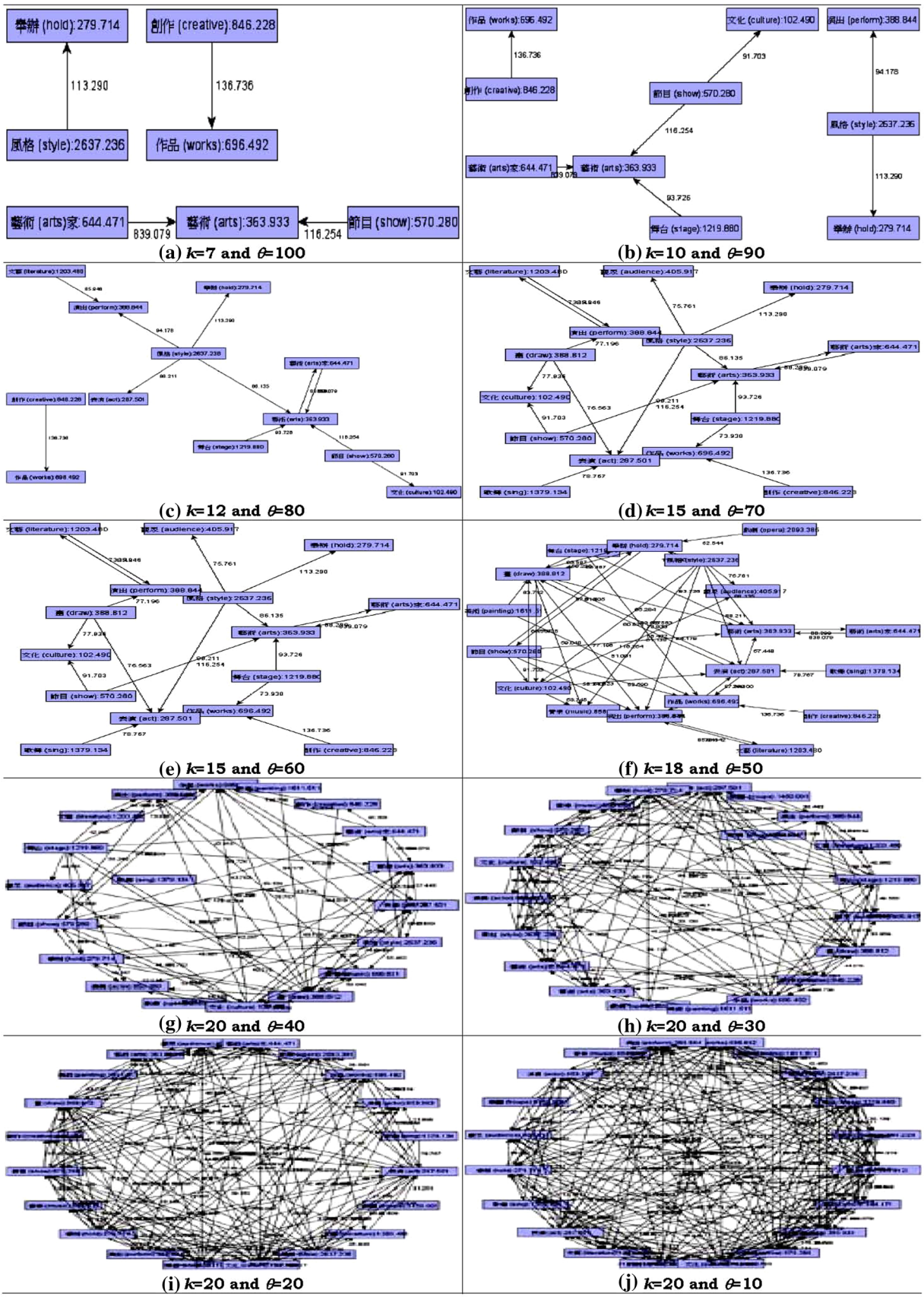


Fig. 15 The DOGs in domain 文藝 with different parameters k and θ

Table 6 Details of classification result of each class

Class	Approach	Precision	Recall	*F*-measure
文藝	Tf–idf	0.9426	0.8243	0.8795
	Term dependency	0.9306	0.9054	0.9178
	Ontology graph	0.9333	0.9200	0.9266
政治	Tf–idf	0.7165	0.9146	0.8035
	Term dependency	0.6032	0.9868	0.7487
	Ontology graph	0.8544	0.8940	0.8738
交通	Tf–idf	0.9831	0.9063	0.9431
	Term dependency	0.9825	0.8750	0.9256
	Ontology graph	0.9355	0.9063	0.9206
教育	Tf–idf	0.9649	0.8333	0.8943
	Term dependency	0.9836	0.9091	0.9449
	Ontology graph	0.9118	0.9394	0.9254
環境	Tf–idf	0.8727	0.8000	0.8348
	Term dependency	0.9245	0.8167	0.8673
	Ontology graph	0.9483	0.8800	0.9129
經濟	Tf–idf	0.6071	0.8763	0.7173
	Term dependency	0.8191	0.7938	0.8063
	Ontology graph	0.8058	0.8557	0.8300
軍事	Tf–idf	0.9111	0.5467	0.6833
	Term dependency	0.9756	0.5333	0.6897
	Ontology graph	0.8571	0.7546	0.8026
醫療	Tf–idf	0.9556	0.7049	0.8113
	Term dependency	1.0000	0.7049	0.8269
	Ontology graph	0.9792	0.7705	0.8624
電腦	Tf–idf	0.9600	0.8136	0.8807
	Term dependency	0.9808	0.8644	0.9189
	Ontology graph	0.8730	0.9257	0.8986
體育	Tf–idf	0.9474	0.9106	0.9286
	Term dependency	0.9918	0.8963	0.9416
	Ontology graph	0.9771	0.9256	0.9507
Ave	Tf–idf	0.8861	0.8130	0.8480
	Term dependency	0.9192	0.8286	0.8715
	Ontology graph	0.9076	0.8772	0.8921

The sizes of the term nodes in DOG tested in this experiment are as follows: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, and 300. Table 7 depicts the experimental results of the three approaches—tf–idf, term dependency, and ontology graph correspondingly, presenting the precision, recall, and *f*-measure with the optimum size highlighted.

As shown in Table 7, the tf–idf approach for the text classification gives accuracy in *f*-measure with 82 and 86.8 %. Using the term size of 10 gives the lowest precision (84.0 %), and using the term size of 80 gives the highest precision (90.5 %). Using the term size of 300 gives the lowest recall (79.1 %), and using the term size of 60 gives the highest recall (83.7 %). Using the term size of 10 gives the lowest *f*-measure (82.0 %), and using the term size of 60 gives the highest *f*-measure (86.8 %). The term dependency approach for the text classification gives accuracy in *f*-measure with 83 and 88.9 %. Using the term size of 10 gives the lowest precision (90.6 %), and using the term size of 300 gives the highest precision (92.1 %). Using the term size of 10 gives the lowest recall (76.6 %), and using the term size of 300 gives the highest recall (86.0 %). Using the term size of 10 gives the lowest *f*-measure (83.0 %), and using the term size of 300 gives the highest *f*-measure (88.9 %). The ontology graph-based approach for the text classification gives accuracy in *f*-measure with 86.6 and 92.3 %. Using the size of ontology graph of 10 gives the lowest precision (90.2 %), and using the size of ontology graph of 80 gives the highest precision (93.6 %). Using the size of ontology graph of 10 gives the lowest recall (83.3 %), and using the size of ontology graph of 80 gives the highest recall (91.0 %). Using the size of ontology graph of 10 gives the lowest *f*-measure (86.6 %), and using the size of ontology graph of 80 gives the highest *f*-measure (92.3 %).

We explore the combination (see Table 8) of results from previous experiments to figure out an optimal setting for the text classification process. We found that the ontology graph is the best approach for implementing the text classification. In addition, the DOG with a term size of 80 gives the best performance. From the results of precision and recall for the three approaches, we demonstrate that the use of ontology graph approach performs the best for every term size used. Generally, for the ontology graph-based text classification approach, the use of smaller sizes of ontology graph results in lower precision and recall. Although the result also shows that the precision and recall are optimized by using the size of 80, any size larger than 80 does not increase the accuracy. Similarly, the experimental results of *f*-measure for the three approaches show that the performance of text classification system is optimized by using the ontology graph approach with the term size of 80.

4.2.5 *Results of using different threshold θ for text classification*

From the previous experimental results, we get that when the ontology graph has 80 terms, the best classification performances, that is, precision, recall, and *f*-measure, can be obtained. In this experiment, we will fix the size of ontology graph (parameter $k = 80$) and further test the influence of parameter θ on the performance of ontology graph-based text classification. The parameter θ is used to filter the edge of which the dependency is lower than threshold θ. That is to say, if the weight of edge is smaller

Table 7 Classification result for tf–idf, term dependency, and ontology graph approaches

Size	Tf–idf			Term dependency			Ontology graph		
k	Precision	Recall	*F*-measure	Precision	Recall	*F*-measure	Precision	Recall	*F*-measure
10	0.8396	0.8011	0.8199	0.9061	0.7657	0.8300	0.9023	0.8329	0.8662
20	0.8723	0.8119	0.8410	0.9130	0.8016	0.8537	0.9116	0.8631	0.8867
30	0.8861	0.8130	0.8480	0.9192	0.8286	0.8715	0.9076	0.8772	0.8921
40	0.8838	0.8162	0.8487	0.9123	0.8310	0.8697	0.9102	0.8846	0.8972
50	0.8877	0.8261	0.8558	0.9107	0.8400	0.8739	0.9213	0.8913	0.9061
60	0.9002	**0.8372**	**0.8676**	0.9087	0.8370	0.8714	0.9239	0.8914	0.9074
70	0.8957	0.8286	0.8608	0.9138	0.8389	0.8747	0.9325	0.9078	0.9200
80	**0.9050**	0.8214	0.8612	0.9187	0.8466	0.8812	**0.9360**	**0.9103**	**0.9230**
90	0.9010	0.8237	0.8606	0.9136	0.8460	0.8785	0.9325	0.9078	0.9200
100	0.8986	0.8157	0.8551	0.9196	0.8544	0.8858	0.9293	0.9054	0.9172
150	0.9031	0.804	0.8506	0.9162	0.8544	0.8842	0.9240	0.9039	0.9138
200	0.8982	0.7962	0.8441	0.9177	0.8548	0.8851	0.9226	0.9015	0.9119
300	0.9034	0.7912	0.8436	**0.9206**	**0.8597**	**0.8891**	0.9254	0.9035	0.9143

Table 8 Summary of the optimized result

	Size for optimized precision	Size for optimized recall	Size for optimized *f*-measure
Tf–idf	80 (90.5 %)	60 (83.7 %)	60 (86.8 %)
Term dependency	300 (92.1 %)	300 (86.0 %)	300 (88.9 %)
Ontology graph	**80 (93.6 %)**	**80 (91.0 %)**	**80 (92.3 %)**

than θ, this edge will be excluded in the calculation of ontology graph generation. Therefore, the higher threshold will decrease more edges in ontology graph and further reduce the size of generated ontology graph. We let the values of θ be 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0, respectively, in this experiment. The experimental results are summarized in Table 9 which depicts the detailed precision, recall, and *f*-measure of ontology graph-based text classification. As shown in Table 9, the classification evaluations in terms of precision, recall, and *f*-measure all decrease gradually with the increase in threshold θ. When $\theta = 0$, that is, all edges are included in the generated ontology graph, the best classification performances are arrived. However, when $\theta = 1.0$, the precision, recall, and *f*-measure all decline to the minimum values. This tells us that the more information be included in the generated ontology graph, the more complete the domain knowledge be represented by the ontology graph, and further the better classification performances be attained by the ontology graph.

Table 9 Classification results with different θ

Threshold θ	Precision	Recall	*F*-measure
0.0	0.9361	0.9104	0.9230
0.1	0.9249	0.8940	0.9092
0.2	0.9141	0.8830	0.8983
0.3	0.9041	0.8728	0.8882
0.4	0.8793	0.8370	0.8576
0.5	0.8597	0.8064	0.8322
0.6	0.8329	0.7590	0.7942
0.7	0.8058	0.7039	0.7514
0.8	0.7895	0.6557	0.7164
0.9	0.7784	0.6038	0.6801
1.0	0.7784	0.5738	0.6606

5 Conclusion

This paper describes a comprehensive and innovative ontology-based information management system called KnowledgeSeeker. We proposed and implemented several different ontological components and processes in KnowledgeSeeker, which are required to develop different kinds of ontology-based intelligent applications. First, we describe a model for representing ontology knowledge called ontology graph and propose an approach for learning the ontology from a text corpus. The approach adopts a chi-square-based statistical learning method to extract and formalize knowledge from a document corpus in the form of domain ontology graph (DOG). Then, the structure of the ontology graph for semiautomatic generation purpose is defined and several ontological operations are presented

which can be carried out with the use of the ontology graph model. Finally, we carried out experiments to evaluate the performance and effectiveness of the proposed method of ontology graph modeling, learning, and the ontological operation. The experimental results showed that the ontology graph-based text classification approach achieved a higher accuracy in classification over other approaches in comparison. The accuracy can further be enhanced by increasing the size of DOGs. The high performance of the ontology graph-based text classification method reveals that the ontology graph learning method is indeed effective and successfully generates a set of small-sized DOGs that are able to represent the domain knowledge. The high performance as shown in the experimental results demonstrates the presented ontological operation with ontology graph is feasible and effective.

6 Limitations and future work

In this paper, we have developed the preliminary stage of ontology learning method for creating domain ontology knowledge. Since the ontology representation (i.e., ontology graph) is simplified for the application developments, the types of relationship between concepts cannot be generated semiautomatically by the ontology learning method. Furthermore, there is still lacking of an effective ontology validation and verification model to measure the integrity and legitimacy of the generated DOG, and this may require human efforts for the validation and verification. A number of enhancements and future research can be summarized as follows: (1) consider to incorporate the types of relationship into the current ontology graph model; (2) extend the proposed ontology graph learning process to other language or support multilingual standard for ontology knowledge sharing; (3) enhance the ontology learning process with the supervised learning methods (e.g., [38, 39, 41, 42]), so that the best ontology graph outcome (i.e., term size) can be optimized; and (4) explore the commercial applications for the proposed KnowledgeSeeker system.

Acknowledgments The authors are very grateful for the editors and anonymous reviewers. Their many valuable and constructive comments and suggestions helped us significantly improve this work. This work was supported in part by the GRF Grant 5237/08E, by the CRG Grant G-U756 of The Hong Kong Polytechnic University, and by the National Natural Science Foundations of China under Grant 60903088 and Grant 61170040.

References

1. Alani H, Sanghee K, Millard DE, Weal MJ, Hall W, Lewis PH, Shadbolt NR (2003) Automatic ontology-based knowledge extraction from web documents. IEEE Intell Syst 18(1):14–21
2. Besana P, Robertson D (2008) Probabilistic dialogue models for dynamic ontology mapping. Lect Notes Comput Sci 5327:41–51
3. Buitelaar P, Ciomiano P (2008) Ontology learning and population: bridging the gap between text and knowledge. IOS Press, The Netherlands
4. Busagala LSP, Ohyama W, Wakabayashi T, Kimura F (2008) Improving automatic text classification by integrated feature analysis. IEICE Trans Inf Syst E91(D4):1101–1109
5. Chen WQ, Mizoguchi R (1999) Communication content ontology for learner model agent in multi-agent architecture. Adv Res Comput Commun Educ 95–102
6. Cimiano P, Hotho A, Staab S (2005) Learning concept hierarchies from text corpora using formal concept analysis. J Artif Intell Res 24(1):305–339
7. Dahab MY, Hassan HA, Rafea A (2008) TextOntoEx: automatic ontology construction from natural English text. Expert Syst Appl 34(2):1474–1480
8. Dicheva D, Dichev C (2007) Authors support in the TM4L environment. Int J Inf Technol Knowl 1(3):215–219
9. Dong ZD, Dong Q (2006) HowNet and the computation of meaning. World Scientific Publishing Company, Singapore
10. Etzioni O, Cafarella M, Downey D,Popescu AM, Shaked T, Soderland S, Weld DS, Yates A (2005) Unsupervised named-entity extraction from the web: an experimental study. Artif Intell 165(1):91–134
11. Fensel D, van Harmelen F, Horrocks I, McGuinness DL, Patel-Schneider PF (2001) OIL: an ontology infrastructure for the semantic web. IEEE Intell Syst 16(2):38–45
12. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. Int J Mach Learn Res 3(7–8):1289–1305
13. Gacitua R, Sawyer P, Rayson P (2008) A flexible framework to experiment with ontology learning techniques. Knowl Based Syst 21(3):192–199
14. Gruber TR (2008) Ontology, encyclopedia of database systems. Springer, Berlin
15. Haase P, Völker J (2008) Ontology learning and reasoning-dealing with uncertainty and inconsistency. Lect Notes Comput Sci 5327:366–384
16. Hazman M, El-Beltagy SR, Rafea A (2009) Ontology learning from domain specific Web documents. Int J Metadata Semant Ontol 4(1/2):24–33
17. Koutero A, Fujita S, Sugawara K (2010) Design of an assisting agent using a dynamic ontology. Proc IEEE/ACIS Int Conf Comput Inf Sci 611–616
18. Lan M, Tan CL, Su J, Lu Y (2009) Supervised and traditional term weighting methods for automatic text categorization. IEEE Trans Pattern Anal Mach Intell 31(4):721–735
19. Lim E, Liu J, Lee R (2009) Knowledge discovery from text learning for ontology modelling. Proc Int Conf Fuzz Syst Knowl Discov 7:227–231
20. Lougheed P, Bogyo B, Brokenshire D, Kumar V (2005) Towards formalizing electronic portfolios. In: Proceedings of the international workshop applications of semantic Web techniques E-Learn. pp 9–18
21. Maedche A (2001) Ontology learning for the semantic web. IEEE Intell Syst 16(2):72–79
22. Mahinovs A, Tiwari A (2007) Text classification method review. Decis Eng Rep Ser 1–13
23. Missikoff M, Velardi P, Fabriani P (2003) Text mining techniques to automatically enrich a domain ontology. Appl Intell 18(3):323–340
24. Mochol M, Jentzsch A, Euzenat J (2006) Applying an analytic method for matching approach selection. In: Proceedings of the international workshop Ontology Match. pp 37–48

25. Navigli R, Velardi P (2004) Learning domain ontologies from document warehouses and dedicated web sites. Comput Linguist 30(2):151–179
26. Navigli R, Velardi P, Cucchiarelli R, Neri F (2004) Automatic ontology learning: supporting a per-concept evaluation by domain experts.in: Proceedinds of the European Conference on artificial intelligence. http://olp.dfki.de/ecai04/final-velardi.pdf
27. Noy NF, Musen MA (2000) PROMPT: algorithm and tool for automated ontology merging and alignment. In: Proceedings of the international conference on articial intelligence and conference on Innovative appliance articial intelligence. pp 450–455
28. Oberle D, Eberhart A, Staab S, Volz R (2004) Developing and managing software components in an ontology-based application server. Lect Notes Comput Sci 3231:459–477
29. Oddy RN (1981) Information retrieval research. Butterworths, London
30. Ottens K, Aussenac-Gilles N, Gleizes MP, Camps V (2007) Dynamic ontology co-evolution from texts: principles and case study. In: Proceedings of the international workshop on the emerging semantics ontology evolving. pp 70–83
31. Rosse C, Mejino JL Jr (2003) A reference ontology for biomedical informatics: the foundational model of anatomy. J Biomed Inform 36(6):478–500
32. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47
33. Shoham Y (1987) Temporal logics in AI: semantical and ontological considerations. Artif Intell 33(1):89–104
34. Simperl E (2009) Reusing ontologies on the semantic web: a feasibility study. Data Knowl Eng 68(10):905−925
35. Sun AX, Lim EP, Ng WK (2003) Performance measurement framework for hierarchical text classification. J Am Soc Inf Sci Tech 54(11):1014–1028
36. Vacura M, Svátek V, Smrž P (2008) Pattern-based framework for representation of uncertainty in ontologies.In: Proceedings of the international conference on text speech and dialogue. pp 227–234
37. Vagin V, Fomina M (2011) Problem of knowledge discovery in noisy databases. Int J Mach Learn Cyber 2(3):135–145
38. Wang XZ, Dong LC, Yan JH (2012) Maximum ambiguity based sample selection in fuzzy decision tree induction. IEEE Trans Knowl Data Eng 24(8):1491–1505
39. Wang XZ, He YL, Dong LC, Zhao HY (2011) Particle swarm optimization for determining fuzzy measures from data. Inf Sci 181(19):4230–4252
40. Warren P (2006) Knowledge management and the semantic web: from scenario to technology. IEEE Intell Syst 21(1):53−59
41. Yi WG, Lu MY, Liu Z (2011) Multi-valued attribute and multi-labeled data decision tree algorithm. Int J Mach Learn Cyber 2(2):67–74
42. Zahiri SH (2012) Classification rule discovery using learning automata. Int J Mach Learn Cyber 3(3):205–213
43. Zhang Y, Vasconcelos W, Sleeman D (2005) OntoSearch: an ontology search engine. Res Dev Intell Syst XXI 1a:58–69
44. Zhang Q, Xing CX, Zhou LZ, Feng JH (2003) An ontology-based method for querying the web data. In: Proceedings of the international conference on advanced information network application 628-631
45. Zhong ZM, Liu ZT, Li CH, Guan Y (2012) Event ontology reasoning based on event class influence factors. Int J Mach Learn Cyber 3(2):133–139