A New Method for Knowledge and Information Management Domain Ontology Graph Model

James N. K. Liu, *Senior Member, IEEE*, Yu-Lin He, *Student Member, IEEE*, Edward H. Y. Lim, and Xi-Zhao Wang, *Senior Member, IEEE*

***Abstract*—A new ontology learning model called domain ontology graph (DOG) is proposed in this paper. There are two key components in the DOG, i.e., the definition of the ontology graph and the ontology learning process. The former defines the ontology and knowledge conceptualization model from the domain-specific text documents; the latter offers the necessary method of semiautomatic domain ontology learning and generates the corresponding ontology graphs. Two kinds of ontological operations are also defined based on the proposed DOG, i.e., document ontology graph generation and ontology-graph-based text classification. The simulation studies focused upon Chinese text data are used to demonstrate the potential effectiveness of our proposed strategy. This is accomplished by generating DOGs to represent the domain knowledge and conducting the text classifications based on the generated ontology graph. The experimental results show that the new method can produce significantly better classification accuracy (e.g., with 92.3% in f-measure) compared with other methods (such as 86.8% in f-measure for the term-frequency–inverse-document-frequency approach). The high performance demonstrates that our presented ontological operations based on the ontology graph knowledge model are effectively developed.**

***Index Terms*—Chinese text analysis, domain ontology graph (DOG), information management, knowledge representation, ontology learning.**

I. Introduction

ONTOLOGY is an important foundation of knowledge that represents the real world. From the perspective of computer science, ontology [11], [28], [46], [47] defines a set of representational primitives with which one models a domain of knowledge or discourse. These representational primitives are able to be comprehended by human beings and come with machine-readable formats [29] that are composed of the classes, the attributes (properties), and the relationships between classes. A well-constructed ontology can help develop a knowledge-based information search and management system more effectively, such as Web search engine [53], [73], semiautomatic text classification [12], [48], [60], and content management system [15], [39], [51]. In addition, recent studies also reflect that the ontology-based induction can indeed obtain effective and outstanding performances in many real applications, e.g., risk management in e-Business and e-Commerce task [19], interoperability between enterprises [44], multimedia integrated service system [5], [6], multiagent e-Learning system [1], [2], telecom operator service [30], etc. To sum up, the ontology can be used as the standard knowledge representation for the semantic Web-based systems [9], [13], [24], [57], [75]. Ontology is widely recognized as an appropriate knowledge representation technology; therefore, research on ontology is becoming more in demand for developing knowledge-based information systems.

However, ontology learning for conceptual formalisms supported by present ontology engineering tools, namely, [25], [26], [54], [59], [63], and [64], might be insufficient and ineffective to represent the uncertain information that are commonly found in [65] and [66] due to the lack of essential knowledge as the core components used by many application domains (particularly Chinese text data). Hence, the creation and the maintenance of the domain ontology for Chinese text data are quite challenging in ontology research area. Ontology engineering [8], [27] is an association of ontology research for developing theories, methods, and software tools that help create and maintain ontology. The typical representatives of such engineering tools include Protégé, Onto-Builder, and Onto-Edit. Protégé [26], [54], first built by Musen *et al.*, provides a Web-based platform with a suite of tools for an expanding user-based community to construct domain models and knowledge-based applications with ontologies. The extracted knowledge with Protégé consists of a set of classes organized in a subsumption hierarchy, a set of slots associated with classes to describe their properties and relationships, and a set of instances of those classes—individual exemplars of the concepts that hold specific values for their properties. Onto-Builder [25], [59] supports the extraction of ontologies from Web search interfaces ranging from simple search engine forms to multiple-page and complex reservation systems. Ontologies from similar domains are then mapped, generating an ever-improving single ontology with which a domain can be queried. Onto-Edit [63], [64] collects all requirements of the envisaged ontology to extend the semiformal description into an appropriate representation language during the refinement phase and to evaluate the target ontology according to the requirement specifications and the formal evaluation criteria. Although many other ontology engineering studies [4], [14], [18], [45], [52] have been also published over

Manuscript received July 15, 2011; revised November 4, 2011; accepted January 18, 2012. Date of publication May 30, 2012; date of current version December 12, 2012. This work was supported in part by the General Research Fund 5237/08E, by the Central Research Grant G-U756 of The Hong Kong Polytechnic University, and by the National Natural Science Foundations of China under Grant 60903088 and Grant 61170040. This paper was recommended by Associate Editor W. Pedrycz.

J. N. K. Liu and E. H. Y. Lim are with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: csnkliu@comp.polyu.edu.hk; edward@iatopia.com).

Y.-L. He and X.-Z. Wang are with the Machine Learning Center, College of Mathematics and Computer Science, Hebei University, Baoding 071002, China (e-mail: csylhe@gmail.com; xizhaowang@ieee.org).

Digital Object Identifier 10.1109/TSMCA.2012.2196431

the last decade, most of them involved manual creation or maintaining the ontology, which is a time-consuming and inefficient task as every process requires deep analysis by domain experts. The direct human intervention may be required at the early stage (before the execution of method) [50], [74] or at the end (to correct the learned conceptualizations) [61], [71] of the domain ontology editing. Some colleagues [7], [37], [56] also argue that human intervention could be placed in the middle (in an iterative method of learning the concepts/properties). Thus, there is also a problem that domain experts may create ontology based on different and subjective views, so that the extracted ontology knowledge is not exact and may be irrelevant to the knowledge domains. Therefore, to tackle these types of problems, the simplified ontology learning method with little or minimal human intervention is more practical and feasible to handle the uncertainty information inherent in the available semantic Webs.

Moreover, ontology learning from text data is the most useful method in formalizing ontology, as text data is a rich and direct source of human knowledge. There are many common text learning technologies, such as [3], [10], [16], [55], and [72]. Analyzing the textual data by computer is a difficult task as it requires some natural language processing and semantic analysis. In the recent years, most research on ontology learning from text [17], [23], [31], [32], [49] have been carried out. Most of those researchers use artificial-intelligence approaches such as machine learning or statistical analysis to develop the methodologies, and they attempt to extract the domain ontology knowledge from text learning semiautomatically. Missikoff *et al.* [49] propose a symbolic ontology management system for the tourism domain in which the text mining approach is adopted herein to reduce the time consumption on constructing or updating the ontology. Dahab *et al.* [17] develop a chain-based ontology construction system (simply TextOntoEx) from natural English text. TextOntoEx supports the construction of domain relations (rather than ontological relations) and nontaxonomic conceptual relationships. Haase and Völker [31] present an approach to generate consistent ontologies from the learned ontology models by taking the uncertainty of knowledge into account. Gacitua *et al.* [23] develop a flexible framework for ontology learning from text that provides a cyclical process involving the successive application of various natural language processing techniques and learning algorithms for concept extraction and ontology modeling. Hazman *et al.* [32] present an accelerated engineering system for the ontology building process via semiautomatically learning a hierarchal ontology when given a set of domain-specific Web documents and a set of seed concepts. All developers have reported a better performance of the ontology learning systems previously mentioned and demonstrated the effectiveness and the usefulness of these proposed approaches. However, most of developers applied only to English text as the text data is language dependent; the algorithms applied on English text were found to be incompatible and untranslatable in Chinese text [40]–[43]. This is based upon the structure of Chinese characters that are more complex and multivariate compared with an English word. Subsequently, developing an effective and time-saving ontology learning engine in Chinese text is useful and valuable in the real applications of Chinese-based information search and management systems.

In this paper, we propose a new and comprehensive method about how to construct and generate the domain ontology from the Chinese text. Compared with existing ontology extraction engines (e.g., Protégé, Onto-Builder, and Onto-Edit), this new model needs minimal human intervention in the process of ontology learning. Moreover, the information search and management systems that mainly contain Chinese text data can be hence enhanced by the ontology, since many existing ontologies are developed in English and cannot be applied to a Chinese-based information system. The innovative ontology learning model designated domain ontology graph (DOG) contains two main components; one is the definition of ontology graph, and the other is the ontology learning process. The former defines the ontology and knowledge conceptualization model from the domain-specific text documents; the latter offers the necessary method of semiautomatic domain ontology learning and generates the corresponding ontology graphs. Two kinds of ontological operations are also defined based on the proposed DOG, i.e., document ontology graph (DocOG) generation and ontology-graph-based text classification. Based on the gathered Web document corpus (totally contains 2 814 Chinese documents with an average of 965 Chinese characters in each document) from ten different domains, i.e., 文藝(Arts and Entertainment), 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), 13 different types of domain ontology graphs with 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 the Chinese News Web site (人民網., www.people.com.cn), we have collected a very large amount of documents (57 218 documents) with an average of 2349 Chinese characters in each news document that belong to ten distinct topic classes previously described. Two classical text classification methods, i.e., term frequency–inverse document frequency (tf-idf) [36], [38], [58] and term dependence [41], [42], are employed as our competitors. The experimental comparisons are carried out according to the following three aspects: precision [62] (measures the accuracy of the retrieval model [21]), recall [20] (measures the ability of the retrieval model to retrieve correct documents from the whole data set [21]), and f-measure (measures the harmonic average of precision and recall) [22]. The experimental results showed the ontology-graph-based text classification approach (by using the 30-term-sized DOG) achieved a high accuracy (89.1% in f-measure) in classification when compared with other approaches. The accuracy can be further enhanced (92.3% in f-measure) by increasing the size of DOGs to 80 terms. The high performance of the ontology-graph-based text classification method reveals that the ontology graph learning method is highly effective and has successfully generated a set of small-sized DOGs (30 to 80 term sizes) capable of representing the domain knowledge. The small-sized DOGs have improved those traditional text classification approaches normally achieved in high-dimensional space. The high performance given in the experimental result also demonstrates that the presented ontological operation with ontology graph is effectively developed.

The rest of this paper is organized as follows: In Section II, we summarize the research background on which our developments are based. Section III 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. The experimental simulations are carried out, and the corresponding analyses to empirical observations are presented in Section IV. Finally, we conclude and outline the main directions for future research in Sections V and VI, respectively.

II. Background

A. Ontology

"Ontology" originates from philosophy, and it has been growing into popular research in computer science [37], [46], [50] and information systems [12], [36], [40]. Within the computer science perspective, ontology defines a set of representational primitives with which one models a domain of knowledge or discourse [17], [19], [50]. The representational primitives of the ontology contains 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 five-tuple-based structure [17], [19], [50] is a commonly used formal description to summarize the concepts and their relationships in a domain. The five-tuple core ontology structure is defined as

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

where

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

Term rel is defined as a set of three-tuple relations, i.e., $\mathrm{rel} = (s, r, o)$, 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 five-tuple ontological structure, knowledge is mainly represented by the logic sentences A, and the most important component is rel, where it defines three-tuple-based concept relationship.

B. Ontology Learning

Text is the most direct resource of human knowledge. Human beings write texts about what they perceive and think about the world; thus, 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 have been widely developed in the recent years [4], [15], [22], [51], [67]. Most of them use artificial-intelligence approaches to develop the methodologies, and the semiautomatic text learning process is the goal of these research. They use many artificial intelligence approaches such as information retrieval [49], machine learning [35], natural language processing [23], and statistical mathematics [31] to build the ontology learning system. However, the ontology learning outcome is sometimes not satisfactory to represent human knowledge [4], [14], [18], [45], [52]. This is because the computational ontology is explicitly defined, but the knowledge in textual data is vague and implicit. There are difficulties in converting the implicit knowledge from text to a formalized ontology representation, in terms of both its quantity and quality. Quantity refers to the fact 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 the fact that the ontology learning outcome cannot express the relevant knowledge. In other words, from the standpoint of human understanding, the formalized knowledge from existing semiautomatic learning processes could partly be irrelevant or wrongly generated due to insufficient and inappropriate knowledge representation.

However, since manually 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 in view of the ontology engineering tool. With the use of semiautomatic ontology learning method, the extracted ontology can serve for two main purposes. First, the ontology outcome can improve the performance of the 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. Second, the ontology outcome can serve as an intermediate ontology or a base ontology for humans to further develop and revise it. The incomplete ontology or the ontology with unsatisfied quality can aid human beings to develop a desired ontology for the knowledge domain, so that it is not required to build the entire ontology from the beginning.

C. Text Classification

With the rapid growth of Internet technologies, vast amounts of Web information are now available online. Information retrieval [9] such as text classification [12], [48], [60] on Web data is 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 $C = \{c_1, c_2, \cdots, c_m\}$ and document d, which determines the relevancy of class c_i in order to find out where document d belongs to. The classifier is function $f_i(d) \rightarrow \{0, 1\}$ that expresses the relevancy value of document d for class c_i. A classical text classification model consists of documents as input, processes with natural language processing, feature extraction, feature weighting, feature reduction, classification engine, and output to relevant classes or categories.

Traditional text classification systems [3], [10], [16], [55], [72] are mostly keyword based without many intelligent features, providing inaccurate results in various 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. In this paper, we proposed the ontology-graph-based text classification approach that improves the traditional text classification system with higher accuracy.

III. Methodology

This section describes the entire operation of the ontology learning and the ontology-graph-based text classification process. We describe the conversion from the original source of the text to the proposed ontology graph model.

A. *Ontology Learning Framework—KnowledgeSeeker*

KnowledgeSeeker is a comprehensive system framework that defines and implements four components, i.e., ontology graph modeling (the ontology graph structure), ontology learning (the learning algorithm), 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 such as 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.

B. *Ontology Graph Modeling*

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

Fig. 1 illustrates the conceptual view of the ontology graph model that is created based on the structure of term nodes and relations. The ontology graph consists of four types of CUs according to their level of complexity exhibiting in knowledge. We define four CUs as any objects (nodes) in the ontology graph that give semantic expression. All of these CUs are linked up and associated by conceptual relations (CR) within each other to comprise the entire conceptual structure of ontology graph and to model an area (a domain) of knowledge.

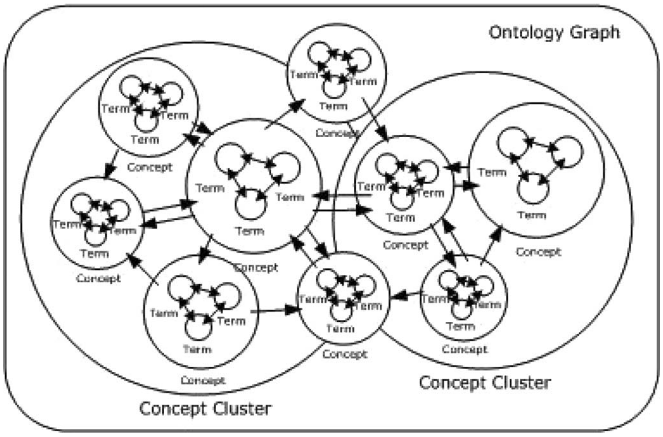


Fig. 1. Conceptual structure of ontology graph in KnowledgeSeeker.

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

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

C. *Ontology Learning Method in Chinese Text*

The ontology learning is the process of learning and creating a domain of knowledge (a particular area of interest such as art, science, entertainment, and sport) in the form of the ontology graph, which is an ontology representation model described in Section III-B. The ontology graph creation is considered as a knowledge extraction process. As described in Section III-B, we defined different levels of knowledge objects, in the form of CU and CR that 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, i.e., term T, to the highest level, i.e., the DOG.

We focused on the ontology learning in Chinese text, because the relationships between Chinese words are more difficult to analyze simply by grammar and word patterns (such as by regular expression) than the English text. Therefore, we use Chinese text as the data source to learn and create the ontology graph, which can reveal the feasibility and the effectiveness of learning ontology based on term relations, through the proposed

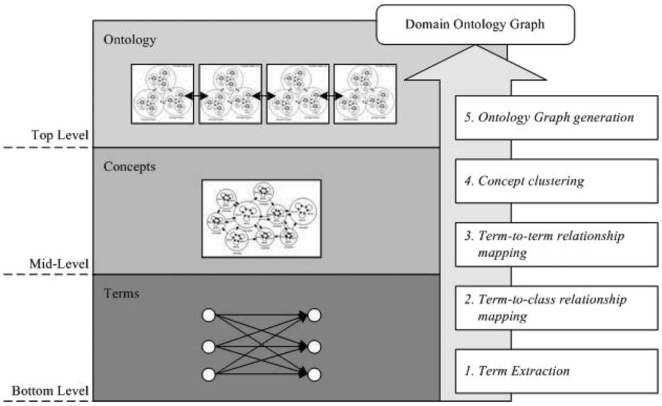


Fig. 2. Bottom–up approach of the DOG learning process.

learning approach. The five learning subprocesses in the bottom–up learning approach are given below.

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 a graph-based ontology as the knowledge representation for practical applications.

Fig. 2 shows all the subprocesses in the bottom–up approach of the DOG learning method. All of these subprocesses correspond to identifying different levels of CU. Thus, the knowledge is learnt from the smallest CU (term) toward the largest CU (DOG).

1) Term Extraction: Our approach focuses on learning the DOG from Chinese text corpus. Since there is no space between characters in Chinese writing, a useful means of word disambiguation is not available in Chinese, which is available in English. For this reason, Chinese term extraction typically relies on dictionaries. An existing electronic dictionary is available such as HowNet [34]. It contains over 50 000 distinct Chinese words, and it is useful to identify a meaningful word inside a text and serve as our initial term list for achieving term extraction processes. 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 a relevant concept and to be thus extracted from the learning process) while filtering out other unnecessary terms that do not appear in the text corpus. n numbers of candidate terms $T = \{t_1, t_2, \cdots, t_n\}$ are thus extracted, where every term $t_i (i = 1, 2, \cdots, n)$ in the term list T appears at least once in the text corpus.

As well as 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 the named person/organization, brand/building names, and new technologies, are usually 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.

2) Term-to-Class Relationship Mapping: The candidate term list T extracted from the Chinese text corpus, however, has no meaning and relationship to any CUs in the ontology graph model. Thus, the second process applied to the candidate term list is the term-to-class relationship mapping. This mapping process acts like a 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 classified into different labels of class or domain topic), and then, we can measure how the terms are related to each class. We then select a sublist of terms in the candidate term list for each class. The mapping process means that we place every term in the sublist associated with a class, by a weighted and directed relation between a term and a class.

The term-to-class relationship mapping applies a χ^2 statistical term-to-class independence measurement to measure the degree of interdependence 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 term t. $O_{t,\neg c}$ is the observed frequency of documents that are not in class c and contain term t. $O_{\neg t,c}$ is the observed frequency of documents that are in class c and do not contain term t, and $O_{\neg t,\neg c}$ is the observed frequency of documents that are neither in class c nor contain term t. The observed frequency is compared with the expected frequency $E_{i,j}$, where $i \in \{t, \neg t\}$ and $j \in \{c, \neg c\}$. $E_{i,j}$ is defined as

$$E_{i,j} = \frac{\sum_{a \in \{t,\neg t\}} O_{a,j} \sum_{b \in \{c,\neg c\}} O_{i,b}}{N} \tag{1}$$

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.

The χ^2 statistic independence measurement for term t and class c is defined as

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

An alternative χ^2-statistical-based word-class dependence measure defines $R_{t,c}$ as

$$R_{t,c} = \frac{O_{t,c}}{E_{t,c}} \tag{3}$$

where $R_{t,c}$ is the ratio between $O_{t,c}$ and $E_{t,c}$. Term t is measured as the positive dependence on class c if $R_{t,c} > 1$, or term t is measured as the negative dependence on class c if $R_{t,c} < 1$. $R_{t,c} = 1$ means that there is no dependence between t and c. In summary, $\chi^2_{t,c}$ measures the dependence between a term and a class in a corpus distribution, whereas $R_{t,c}$ measures whether the dependence is positive or negative, i.e.,

$$t \begin{cases} \text{positive dependence on } c & \text{if} \quad R_{t,c} > 1 \\ \text{negative dependence on } c & \text{if} \quad R_{t,c} < 1. \end{cases}$$

3) Term-to-Term Relationship Mapping: The term-to-term relationship mapping is a further learning process that aims to calculate the interrelationships 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, although we do not know how those terms are related to each other inside the class. Subsequently, the term-to-term relationship mapping further finds out and calculates this weighted relationship between those terms.

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

1) The k-number of highest ranked positive terms in each class, for there are m number of classes $V = \{v_1, v_2, \cdots, v_m\}$ for each $v_i = \{(t_1, \chi^2_{t_1,c_i}, R_{t_1,c_i}), (t_2, \chi^2_{t_2,c_i}, R_{t_2,c_i}), \cdots, (t_k, \chi^2_{t_k,c_i}, R_{t_k,c_i})\}$, where $R_{t_j,c_i} = O_{t_j,c_i}/E_{t_j,c_i} > 1$ denotes the dependence of term $t_j (t_j \in T, T = \{t_1, t_2, \cdots, t_k\})$ on class $c_i (c_i \in C, C = \{c_1, c_2, \cdots, c_m\})$ is positive;
2) If the number of positive terms $(R_{t_j,c_i} > 1)$ in a class is smaller than 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 χ^2 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 independence between two terms, instead of between a term and a class in the previous term-to-class mapping process. The co-occurrence frequency between 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_{t_a,t_b} 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 that contains term t_a but does not contain term t_b. $O_{\neg t_a,t_b}$ is the observed frequency of documents that does not contain term t_a but contains term t_b. $O_{\neg t_a,\neg t_b}$ is the observed frequency of documents that does not contain both terms t_a and t_b, and $N = O_{t_a,t_b} + O_{\neg t_a,t_b} + O_{t_a,\neg t_b} + O_{\neg t_a,\neg t_b}$ is also the size of a classified document corpus.

Equations (2) and (3) given in Section III-C2 are thus changed to (4) and (5), respectively, to measure the dependence between terms t_a and t_b, i.e.,

$$\chi^2_{t_a,t_b} = \sum_{i \in \{t_a, \neg t_a\}} \sum_{j \in \{t_b, \neg t_b\}} (\frac{O_{i,j} - E^2_{i,j})^2}{E_{i,j}} \tag{4}$$

$$R_{t_a,t_b} = \frac{O_{t_a,t_b}}{E_{t_a,t_b}}. \tag{5}$$

$\chi^2_{t_a,t_b}$ measures the dependence between terms t_a and t_b in a corpus distribution, whereas R_{t_a,t_b} measures whether this dependence is positive or negative. The normalized $\chi^2_{t_a,t_b}$ of class $c_i (c_i \in C, C = \{c_1, c_2, \cdots, c_m\})$ can be calculated as

$$n\chi^2_{t_a,t_b} = \frac{\chi^2_{t_a,t_b}}{\chi^2_{t_a,c_i}}. \tag{6}$$

Therefore, after the term-to-term relationship mapping and normalization process, we can obtain two $k \times k$ term-to-term dependence matrices containing the values of $n\chi^2_{t_a,t_b}$ and R_{t_a,t_b} that represent the relationship of every term-to-term pair within class $c_i (c_i \in C, C = \{c_1, c_2, \cdots, c_m\})$.

The term independence representations can be converted into a directed graph $G = (V, A)$, where V is the set of selected terms, $V = \{t_1, t_2, \cdots, t_{k-1}, t_k\}$, and A is the set of directed and weighted edges $A = \{(t_i, t_j, n\chi^2_{t_i,t_j}) | i \in \{1, 2, \cdots, k\}, j \in \{1, 2, \cdots, k\}\}$. Whether the edge between 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 $R_{t_i,t_j} > 1, i \in \{1, 2, \cdots, k\}, j \in \{1, 2, \cdots, k\}$ that denotes the dependence between 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, $R_{t_i,t_j} < 1, i \in \{1, 2, \cdots, k\}, j \in \{1, 2, \cdots, k\}$, denoting that the dependence between two terms t_i and t_j is negative, and then, the associated link between these two terms will not be represented in the directed graph G.

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 that need to be created. The highest weighted edge e_x with two vertices t_a and t_b is first 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.

The concept clustering therefore creates a second taxonomical relationship. The first layer of concept hierarchy is created in the 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.

5) DOG Generation: In KnowledgeSeeker, we define the 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 and are automatically generated by the statistical learning method, as presented in Section III-C. In the following, we formalize the definition of DOG.

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

$$\mathrm{OG}_d = (T, F, H, R, C, A)$$

where

- d defines the domain of the ontology graph that is associated with;
- T is a set of terms t_i of OG_d;
- F is a set of word functions of terms $t_i \in T$;
- H is a set of taxonomy relationship of T;
- R is a set of relations between t_i and t_j, where $t_i, t_j \in T$;
- C is a set of clusters of $t_1, \cdots, t_n$, where $t_1, \cdots, t_n \in T$;
- A is a set of axioms characterizing 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 any existing Chinese dictionary), defining and mapping the types of word function (may be also obtained from the same dictionary), and labeling the concept clusters. The semiautomatic processes include the domain-term extraction, the term relationship extraction, and the concept cluster extraction.

D. DocOG Generation

A DocOG is another type of ontology graph which is used to represent the content of a single text document. Traditional information retrieval system usually represents documents in term vectors. In the 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 achieved 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. A 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 an intersection of same terms. 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 1 describes the detailed process of the DocOG generation.

Algorithm 1 DocOG generation

```
Input: A text document and a generated DOG $\mathrm{OG}_d$ for domain $d$;
Output: A DocOG ($\mathrm{OG}_{\mathrm{doc}}$) representing the input document $\mathrm{doc}$;
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 the relation set $R_d$ of a DOG $\mathrm{OG}_d = (T_d, F_d, H_d, R_d, C_d, A_d)$;
Transform the document to the weighted term vector $V_{\mathrm{doc}} = \{(t_1, w_{t_1}), (t_2, w_{t_2}), \ldots, (t_k, w_{t_k})\}$, where $w_{t_i}$, $t_i \in T$ denotes the weight of term $t_i$ in document $\mathrm{doc}$. (Weight $w_{t_i}$ is defined as $w_{t_i} = \mathrm{tf}_i/\mathrm{dl}$, where $\mathrm{tf}_i$ is the frequency of term $t_i$ appearing in document $\mathrm{doc}$ and $\mathrm{dl}$ is the length of document $\mathrm{doc}$, i.e., the size of the term list of document $\mathrm{doc}$.)
Ontology graph matching for creating the term set $T_{\mathrm{doc}}$ and the relation set $R_{\mathrm{doc}}$ of DocOG ($\mathrm{OG}_{\mathrm{doc}}$);
for every term $t_i$ in $V_{\mathrm{doc}}$ do
    if term $t_i$ exits in $T_d$ of $\mathrm{OG}_d$ then
        for every related term $t_j$ in $R_d$ of $\mathrm{OG}_d$ do
            Add both $t_i$ and $t_j$ to the term set $T_{\mathrm{doc}}$ of $\mathrm{OG}_{\mathrm{doc}}$;
            Add the relation to the relation set $R_{\mathrm{doc}}$ of $\mathrm{OG}_{\mathrm{doc}}$ for terms $t_i$ and $t_j$;
            Calculate 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 $T_{\mathrm{doc}}$ and $R_{\mathrm{doc}}$ values of $\mathrm{OG}_{\mathrm{doc}}$.
```

E. Ontology-Graph-Based Text Classification

Ontology-graph-based text classification is accomplished 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 specified domain. When a DocOG is compared with more than one DOG, the highest scored DOG in the result is the domain that the document is mostly related to. The detailed text classification process can be summarized as follows: 1) Generate DocOGs by matching the document to every DOG (class). (2) Obtain the scores vectors of every DocOG. (3) Select the highest scored DocOG as the classified domain.

IV. Experimental Results

A. Experimental Data Set

The source of knowledge serves for the data prepared for DOG generation. In the entire experiment, from the learning source to the 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 are labeled as belonging to ten distinct topic classes, which are 文藝(Arts and Entertainment), 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 into 70% for the learning set ($D_1 - \mathrm{Learn}$) and 30% for the testing set ($D_1 - \mathrm{Test}$). The document distributions (including the total document count, the size of training set, and the size of testing set) of the ten classes are shown in Table I. We use only the 70% classified documents

TABLE I
DETAILED DOCUMENT DISTRIBUTION OF CORPUS D_1

Class	(English)	Number of documents	$D_1 - Learn(70\%)$	$D_1 - Test(30\%)$	Number of positive terms	Number of negative terms
文藝	(Arts and Entertainments)	248	174	74	867	29281
政治	(Politics)	505	354	151	966	37481
交通	(Traffic)	214	150	64	769	34691
教育	(Education)	220	154	66	904	30604
環境	(Environment)	201	141	60	788	32823
經濟	(Economics)	325	228	97	664	35680
軍事	(Military)	249	174	75	727	33439
醫療	(Health and Medical)	204	143	61	862	35527
電腦	(Computer and Information Technology)	198	139	59	774	30671
體育	(Sports)	450	315	135	956	37061

(1972 documents) for the process of the term extraction and the term-to-class relationship mapping. The results of the number of positive and negative terms in these ten classes are also shown in Table I.

There is another Chinese document set from an unclassified warehouses (D_2). It is used for the process of term-to-term relationship mapping and concept clustering. The unclassified warehouses D_2 contains a relatively large amount of documents (57 218 documents) that are collected from a Chinese News Web site (人民網, www.people.com.cn), with an average of 2349 Chinese characters in each news document.

B. Performance of the Ontology-Graph-Based Text Classification

The text classification experiment described here has two purposes. On 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 and the number of terms in the DOG for a class that can produce the best classification result.

1) Experiments Description: The designed experiments consist of two parts, i.e., evaluating the performance of ontology-graph-based approach (experiment 1) and evaluating the optimal size of DOG for the best classification result (experiment 2).

The first experiment presents a text classification case by using three different approaches to classify documents. The first is the traditional tf-idf approach [36], [38], [58]. The second is the term-dependence approach [41], [42], which replaces the tf-idf weight by the term-dependence weight in the 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 (recall/precision). The three different text classification approaches are described below.

- The 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 [36], [38], [58], and it is defined as:

$$\mathrm{score}(t_i) = \mathrm{tf}_{t_i} \times \mathrm{idf}_{t_i}.$$

- Term-dependence approach. This approach [41], [42] 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 dependence measurement R, and it is calculated in the DOG learning process. In our paper, the term-dependence scoring function is defined as:

$$\mathrm{score}(t_i) = \mathrm{tf}_{t_i} \times R_{t_i}.$$

- Ontology graph approach. The ontology-graph-based text classification approach is processed by measuring the similarity between the DOG and the corresponding DocOG, as described in Section III-E. Therefore, the scoring function for comparing a document to a domain is defined as

$$\mathrm{score}(\mathrm{doc}, \mathrm{DOG}_j) = \mathrm{score}(\mathrm{DocOG}, \mathrm{DOG}_j).$$

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

The second experiment presents an extended text classification case by using those three different approaches presented in experiment 1 and further varying the size of dimension of terms used in each approach. The process used the same scoring functions presented in experiment 1 and tried to apply them into different sizes of the term vector (for tf-idf and term-dependence 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 affects the classification performance.

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.

- Precision [21], [62]—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

$$\text{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}.$$

- Recall [20], [21]—It measures the ability of the retrieval model to retrieve correct documents from the whole data set by calculating the percentage of correctly retrieved documents to all the documents that should be retrieved. It is defined by

$$\text{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}.$$

- F-measure [22]—It measures the harmonic average of precision and recall. It is defined by

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

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

3) Performance on Ontology-Graph-Based Text Classification: Precision and recall values have been computed for the three classification approaches. Table II shows the detailed results obtained by computing precision, recall, and f-measure for the three approaches using a term size of 30, i.e., 30 of the terms are used in the tf-idf method and 30-term-sized DOG (as presented in Section IV-B) is used in the term-dependence and ontology-graph methods. The table summarizes the precision, the recall, and the f-measure of each class in the testing document set, and also its average. The aforementioned experimental result has shown that the ontology-graph approach performs the highest classification accuracy (89.2% of f-measure). The term-dependence method performs the second highest classification accuracy (87.2% of f-measure), whereas 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 with the term-dependence method, it is revealed that the relationship of concepts in the ontology graph is useful for representing knowledge. This is because using the relationship information in the DOG (ontology-graph approach) performs better results of text classification rather than not using the relationship (term-dependence approach) in the first place. Therefore, this concludes that the ontology-graph approach is effective for developing a text classification system.

4) Performance on Using Different Size of Terms (Dimensionality) 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 the 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 the DOG. The sizes of the term nodes in the DOG tested in this experiment are 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, and 300. Table III displays the experimental results of the three approaches, i.e., tf-idf, term dependence, and ontology graph, correspondingly, presenting the precision, the recall, and the f-measure with the optimum size highlighted.

TABLE II
DETAILS OF CLASSIFICATION RESULT OF EACH CLASS

Class	Approach	Precision	Recall	F-measure
文藝 (Arts)	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 (Politics)	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 (Traffic)	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 (Education)	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 (Environment)	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 (Economic)	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 (Military)	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 (Health)	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 (Computer)	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 (Sport)	tf-idf	0.9474	0.9106	0.9286
	term-dependency	0.9918	0.8963	0.9416
	ontology-graph	0.9771	0.9256	0.9507
Average	tf-idf	0.8861	0.8130	0.8480
	term-dependency	0.9192	0.8286	0.8715
	ontology-graph	0.9076	0.8772	0.8921

As shown in Table III, the tf-idf approach for the text classification gives accuracy in the f-measure in ranges 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-dependence approach for the text classification gives accuracy in the f-measure in ranges 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

TABLE III
CLASSIFICATION RESULT FOR TF-IDF, TERM-DEPENDENCY, AND ONTOLOGY-GRAPH APPROACHES

Size	Tf-idf			Term-dependency			Ontology-graph		
	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

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 the f-measure in ranges 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 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 the 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 the recall are optimized by using the size of 80, any size larger than 80 does not increase the accuracy. Similarly, the experimental results of the 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.

V. Conclusion

This paper has described a comprehensive and innovative ontology-based system framework called KnowledgeSeeker. We have proposed and implemented different ontological components and processes in the KnowledgeSeeker that are required to develop different kinds of ontology-based intelligent applications. First, we have described a model for representing ontology knowledge called ontology graph and have proposed 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 the DOG. Then, the structure of the ontology graph for semiautomatic generation purpose has been defined, and several ontological operations have been presented, which can be carried out with the use of the ontology graph model. Finally, we have carried out experiments to evaluate the performance and the effectiveness of the proposed method of ontology graph modeling and learning, and the ontological operation. The experimental results showed that the ontology-graph-based text classification approach achieved a high accuracy in classification over other approaches for comparison. The accuracy can be further 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 highly effective and has successfully generated a set of small-sized DOGs that were able to represent domain knowledge. The high performance, as shown in the experimental result, demonstrates that the presented ontological operation with the ontology graph is effectively developed.

VI. Limitations and Future Works

In this paper, we have developed the preliminary stage of ontology learning method for creating domain ontology knowledge. Since the ontology representation (ontology graph) is simplified for the application developments, the types of relationship between concepts cannot be semiautomatically generated by the ontology learning method. Furthermore, there is still lacking of an effective ontology validation and verification model to measure the integrity and the legitimacy of the generated DOG, and this may require human efforts for the validation and the 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 supports multilingual standard for ontology knowledge sharing, (3) use other ontology language such as resource description framework and web ontology language to extend and convert our ontology graph generation; 4) enhance the semiautomatic ontology learning process with the supervised learning methods (e.g., [33], [35], [67]–[70]), so that the best ontology graph outcome (such as the best term size) can be optimized through iterative and supervised learning process; and 5) explore the commercial applications for the proposed KnowledgeSeeker system.

Acknowledgment

The authors would like to thank the editors and anonymous reviewers. Their valuable and constructive comments and suggestions helped them in significantly improving this paper.

References

[1] G. Acampora, M. Gaeta, and V. Loia, "Combining multi-agent paradigm and memetic computing for personalized and adaptive learning experiences," *Comput. Intell.*, vol. 27, no. 2, pp. 141–165, May 2011.

[2] G. Acampora, V. Loia, and M. Gaeta, "Exploring e-learning knowledge through ontological memetic agents," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 66–77, May 2010.

[3] C. C. Aggarwal, S. C. Gates, and P. S. Yu, "On using partial supervision for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 2, pp. 245–255, Feb. 2004.

[4] H. Alani, K. Sanghee, D. E. Millard, M. J. Weal, W. Hall, P. H. Lewis, and N. R. Shadbolt, "Automatic ontology-based knowledge extraction from Web documents," *IEEE Intell. Syst.*, vol. 18, no. 1, pp. 14–21, Jan./Feb. 2003.

[5] L. Ballan, M. Bertini, A. Del Bimbo, and G. Serra, "Video annotation and retrieval using ontologies and rule learning," *IEEE Multimedia*, vol. 17, no. 4, pp. 80–88, Oct.–Dec. 2010.

[6] M. Bertini, G. Becchi, A. Del Bimbo, A. Ferracani, and D. Pezzatini, "A Web system for ontology-based multimedia annotation, browsing and search," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 11–15, 2011, pp. 1–4.

[7] P. Besana and D. Robertson, "Probabilistic dialogue models for dynamic ontology mapping," *Lect. Notes Comput. Sci.*, vol. 5327, pp. 41–51, 2008.

[8] P. K. Bhowmick, D. Roy, S. Sarkar, and A. Basu, "A framework for manual ontology engineering for management of learning material repository," *Int. J. Comput. Sci. Appl.*, vol. 7, no. 2, pp. 30–51, 2010.

[9] I. I. Bittencourt, E. Costa, M. Silva, and E. Soares, "A computational model for developing semantic Web-based educational systems," *Knowl.-Based Syst.*, vol. 22, no. 4, pp. 302–315, May 2009.

[10] D. Bonino, F. Corno, and F. Pescarmona, "Automatic learning of text-to-concept mappings exploiting wordnet-like lexical networks," in *Proc. ACM Symp. Appl. Comput.*, Mar. 13–17, 2005, pp. 1639–1644.

[11] P. Buitelaar and P. Ciomiano, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*. Amsterdam, The Netherlands: IOS Press, 2008.

[12] L. S. P. Busagala, W. Ohyama, T. Wakabayashi, and F. Kimura, "Improving automatic text classification by integrated feature analysis," *IEICE Trans. Inf. Syst.*, vol. E91-D, no. 4, pp. 1101–1109, Nov. 2008.

[13] M. Cai, W. Y. Zhang, and K. Zhang, "ManuHub: A semantic Web system for ontology-based service management in distributed manufacturing environments," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 3, pp. 574–582, May 2011.

[14] W. Q. Chen and R. Mizoguchi, "Communication content ontology for learner model agent in multi-agent architecture," in *Proc. Adv. Res. Comput. Commun. Educ.*, 1999, pp. 95–102.

[15] P. Cimiano, A. Hotho, and S. Staab, "Learning concept hierarchies from text corpora using formal concept analysis," *J. Artif. Intell. Res.*, vol. 24, no. 1, pp. 305–339, Jul. 2005.

[16] D. A. Cruse, *Word Meanings and Concepts, Meaning in Language: An Introduction to Semantics and Pragmatics*. London, U.K.: Oxford Univ. Press, 2004.

[17] M. Y. Dahab, H. A. Hassan, and A. Rafea, "TextOntoEx: Automatic ontology construction from natural English text," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1474–1480, Feb. 2008.

[18] D. Dicheva and C. Dichev, "Authors support in the TM4L environment," *Int. J. Inf. Technol. Knowl.*, vol. 1, no. 3, pp. 215–219, 2007.

[19] A. Ekelhart, S. Fenz, and T. Neubauer, "Ontology-based decision support for information security risk management," in *Proc. Int. Conf. Syst.*, Mar. 1–6, 2009, pp. 80–85.

[20] O. Etzioni, M. Cafarella, D. Downey, A. M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the Web: An experimental study," *Artif. Intell.*, vol. 165, no. 1, pp. 91–134, Jun. 2005.

[21] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[22] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Int. J. Mach. Learn. Res.*, vol. 3, no. 7–8, pp. 1289–1305, Mar. 2003.

[23] R. Gacitua, P. Sawyer, and P. Rayson, "A flexible framework to experiment with ontology learning techniques," *Knowl.-Based Syst.*, vol. 21, no. 3, pp. 192–199, Apr. 2008.

[24] M. Gaeta, F. Orciuoli, S. Paolozzi, and S. Salerno, "Ontology extraction for knowledge reuse: The e-learning perspective," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 4, pp. 798–809, Jul. 2011.

[25] A. Gal, G. Modica, and H. Jamil, "OntoBuilder: Fully automatic extraction and consolidation of ontologies from Web sources," in *Proc. ICDE*, Mar. 30–2, Apr. 2004, p. 853.

[26] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubzy, H. Eriksson, N. F. Noy, and S. W. Tu, "The evolution of protégé: An environment for knowledge-based systems development," *Int. J. Hum. Comput. Stud.*, vol. 58, no. 1, pp. 89–123, Jan. 2003.

[27] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-commerce and the Semantic Web*. Berlin, Germany: Springer-Verlag, 2004.

[28] T. R. Gruber, *Ontology, Encyclopedia of Database Systems*. Berlin, Germany: Springer-Verlag, 2008.

[29] N. Guarino and P. Giaretta, "Ontologies and knowledge bases: Towards a terminological clarification," *Towards Very Large Knowl. Bases*, vol. 1, no. 9, pp. 25–32, 1995.

[30] P. Gutheim, "An ontology-based context inference service for mobile applications in next-generation networks," *IEEE Commun. Mag.*, vol. 50, no. 1, pp. 60–66, Jan. 2011.

[31] P. Haase and J. Völker, "Ontology learning and reasoning-dealing with uncertainty and inconsistency," *Lect. Notes Comput. Sci.*, vol. 5327, pp. 366–384, 2008.

[32] M. Hazman, S. R. El-Beltagy, and A. Rafea, "Ontology learning from domain specific Web documents," *Int. J. Metadata Semant. Ontol.*, vol. 4, no. 1/2, pp. 24–33, Jun. 2009.

[33] Q. He and C. X. Wu, "Separating theorem of samples in Banach space for support vector machine learning," *Int. J. Mach. Learn. Cyber.*, vol. 2, no. 1, pp. 49–54, Mar. 2011.

[34] HowNet. [Online]. Available: http://www.keenage.com/

[35] G. B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: A survey," *Int. J. Mach. Learn. Cyber.*, vol. 2, no. 2, pp. 107–122, Jun. 2011.

[36] D. Isa, L. H. Lee, V. P. Kallimani, and R. RajKumar, "Text document preprocessing with the Bayes formula for classification using the support vector machine," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1264–1272, Sep. 2008.

[37] A. Koutero, S. Fujita, and K. Sugawara, "Design of an assisting agent using a dynamic ontology," in *Proc. IEEE/ACIS Int. Conf. Comput. Inf. Sci.*, Aug. 18–20, 2010, pp. 611–616.

[38] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 721–735, Apr. 2009.

[39] Y. Li, C. Lao, and S. M. Chung, "Text clustering with feature selection by using statistical data," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 5, pp. 641–652, May 2008.

[40] E. H. Y. Lim and R. S. T. Lee, "iJADE InfoSeeker: On using intelligent context-aware agents for retrieving and analyzing Chinese Web articles," *Comput. Intell. Agent-Based Syst.*, vol. 72, pp. 127–153, 2007.

[41] E. H. Y. Lim, R. S. T. Lee, and J. N. K. Liu, "KnowledgeSeeker—An ontological agent-based system for retrieving and analyzing Chinese Web articles," in *Proc. IEEE Int. Conf. Fuzz. Syst.*, Jun. 1–6, 2008, pp. 1034–1041.

[42] E. H. Y. Lim, J. N. K. Liu, and R. S. T. Lee, "Knowledge discovery from text learning for ontology modelling," in *Proc. Int. Conf. Fuzz. Syst. Knowl. Discov.*, Aug. 14–16, 2009, vol. 7, pp. 227–231.

[43] E. H. Y. Lim, H. W. K. Tam, S. W. K. Wong, J. N. K. Liu, and R. S. T. Lee, "Collaborative content and user-based Web ontology learning system," in *Proc. IEEE Int. Conf. Fuzz. Syst.*, Aug. 20–24, 2009, pp. 1050–1055.

[44] H. Liu, A. F. C. Decelle, and J. P. Bourey, "Use of ontology for solving interoperability problems between enterprises," in *Proc. IFIP Adv. Inf. Commun. Technol.*, 2010, vol. 336, pp. 730–737.

[45] P. Lougheed, B. Bogyo, D. Brokenshire, and V. Kumar, "Towards formalizing electronic portfolios," in *Proc. Int. Workshop Appl. Semant. Web Tech. E-Learn*, Oct. 2–5, 2005, pp. 9–18.

[46] A. Maedche, "Ontology learning for the semantic Web," *IEEE Intell. Syst.*, vol. 16, no. 2, pp. 72–79, Mar./Apr. 2001.

[47] A. Maedche, *Ontology Learning for the Semantic Web*. Norwell, MA: Kluwer, 2002.

[48] A. Mahinovs and A. Tiwari, "Text classification method review," Decis. Eng. Rep. Ser., Cranfield University, Swindon, U.K., 2007.

[49] M. Missikoff, P. Velardi, and P. Fabriani, "Text mining techniques to automatically enrich a domain ontology," *Appl. Intell.*, vol. 18, no. 3, pp. 323–340, May 2003.

[50] M. Mochol, A. Jentzsch, and J. Euzenat, "Applying an analytic method for matching approach selection," in *Proc. Int. Workshop Ontol. Match.*, Nov. 5, 2006, pp. 37–48.

[51] R. Navigli and P. Velardi, "Learning domain ontologies from document warehouses and dedicated Web sites," *Comput. Linguist.*, vol. 30, no. 2, pp. 151–179, Jun. 2004.

[52] R. Navigli, P. Velardi, R. Cucchiarelli, and F. Neri, "Automatic ontology learning: Supporting a per-concept evaluation by domain experts," in *Proc. 2004 Eur. Conf. Artif. Intell.*, Aug. 22–27, 2004.

[53] A. D. Nicole, M. Missikoff, and R. Navigli, "A software engineering approach to ontology building," *Inf. Syst.*, vol. 34, no. 2, pp. 258–275, Apr. 2009.

[54] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen, "Creating semantic Web contents with protégé-2000," *IEEE Intell. Syst.*, vol. 16, no. 2, pp. 60–71, Mar./Apr. 2001.

[55] R. N. Oddy, *Information Retrieval Research*. London, U.K.: Butterworth, 1981.

[56] K. Ottens, N. Aussenac-Gilles, M. P. Gleizes, and V. Camps, "Dynamic ontology co-evolution from texts: Principles and case study," in *Proc. Int. Workshop Emerg. Semant. Ontol. Evol.*, Nov. 12, 2007, pp. 70–83.

[57] L. Razmerita, "An ontology-based framework for modeling user behavior-a case study in knowledge management," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 4, pp. 772–783, Jul. 2011.

[58] Y. Rezgui, "Text based domain ontology building using Tf-idf and metric clusters techniques," *Knowl. Eng. Rev.*, vol. 22, no. 4, pp. 379–403, Dec. 2007.

[59] H. Roitman and A. Gal, "OntoBuilder: Fully automatic extraction and consolidation of ontologies from Web sources using sequence semantics," *Lect. Notes Comput. Sci.*, vol. 4254, pp. 573–576, 2006.

[60] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002.

[61] E. Simperl, "Reusing ontologies on the semantic Web: A feasibility study," *Data Knowl. Eng.*, vol. 68, no. 10, pp. 905–925, Oct. 2009.

[62] A. X. Sun, E. P. Lim, and W. K. Ng, "Performance measurement framework for hierarchical text classification," *J. Amer. Soc. Inf. Sci. Tech.*, vol. 54, no. 11, pp. 1014–1028, Sep. 2003.

[63] Y. Sure, J. Angele, and S. Staab, "OntoEdit: Guiding ontology development by methodology and inferencing," *Lect. Notes Comput. Sci.*, vol. 2519, pp. 1205–1222, 2002.

[64] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke, "OntoEdit: Collaborative ontology development for the semantic Web," *Lect. Notes Comput. Sci.*, vol. 2342, pp. 221–235, 2002.

[65] Q. T. Tho, S. C. Hui, A. C. M. Fong, and T. H. Cao, "Automatic fuzzy ontology generation for semantic Web," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 6, pp. 842–856, Jun. 2006.

[66] M. Vacura, V. Svátek, and P. Smr, "A pattern-based framework for representation of uncertainty in ontologies," in *Proc. Int. Conf. Text, Speech Dialogue*, Sep. 2008, pp. 227–234.

[67] L. J. Wang, "An improved multiple fuzzy NNC system based on mutual information and fuzzy integral," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 1, pp. 25–36, Mar. 2011.

[68] X. Z. Wang and C. R. Dong, "Improving generalization of fuzzy if-then rules by maximizing fuzzy entropy," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 3, pp. 556–567, Jun. 2009.

[69] X. Z. Wang, L. C. Dong, and J. H. Yan, "Maximum ambiguity based sample selection in fuzzy decision tree induction," *IEEE Trans. Knowl. Data Eng.*, DOI: 10.1109/TKDE.2011.67. [Online]. Available: http://www.computer.org/csdl/trans/tk/preprint/ttk2011990045-abs.html

[70] X. Z. Wang, Y. L. He, L. C. Dong, and H. Y. Zhao, "Particle swarm optimization for determining fuzzy measures from data," *Inf. Sci.*, vol. 181, no. 19, pp. 4230–4252, Oct. 2011.

[71] P. Warren, "Knowledge management and the semantic Web: From scenario to technology," *IEEE Intell. Syst.*, vol. 21, no. 1, pp. 53–59, Jan./Feb. 2006.

[72] D. H. Widyantoro and J. Yen, "Relevant data expansion for learning concept drift from sparsely labelled data," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 401–412, Mar. 2005.

[73] Y. Zhang, W. Vasconcelos, and D. Sleeman, "OntoSearch: An ontology search engine," in *Proc. Res. Dev. Intell. Syst. XXI, Sess. 1a*, 2005, pp. 58–69.

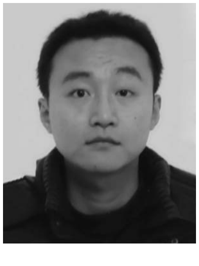
[74] Q. Zhang, C. X. Xing, L. Z. Zhou, and J. H. Feng, "An ontology-based method for querying the Web data," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 27–29, 2003, pp. 628–631.

[75] Y. J. Zhao, J. Dong, and T. Peng, "Ontology classification for semantic-Web-based software engineering," *IEEE Trans. Serv. Comput.*, vol. 2, no. 4, pp. 303–317, Oct.–Dec. 2009.

James N. K. Liu (SM'07) received the Ph.D. degree in computer science from La Trobe University, Melbourne, Australia, in 1991.

He was a Research Scientist with the Aeronautical Research Laboratory and the Defence Science and Technology Organizations in Australia in the early 1990s. He is currently an Associate Professor with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong. His research interests include forecasting technology, uncertainty-based decision support, ontology modeling for content management, e-Learning systems, and e-Commerce development and applications.

Dr. Liu is the Chairman of the 2012 IEEE Hong Kong Systems, Man, and Cybernetics (SMC) Chapter and the Special Sessions Cochair of the 2012 IEEE International Conference on SMC. He has been the Session Chair of the International Workshop on Reliability Issues of Knowledge Discovery (2006, 2008, and 2010), the International Conference on Fuzzy Systems and Knowledge Discovery 2009, and the World Congress on Computational Intelligence 2008.



Yu-Lin He (S'08) received the B.S. degree in applied mathematics and the M.S. degrees in computer science in 2005 and 2009, respectively, from Hebei University, Baoding, China, where he is currently working toward the Ph.D. degree in the College of Mathematics and Computer Science.

From February 2011 to January 2012, he was a Research Assistant with the Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong. His research interests include computational intelligence in game, artificial neural networks, evolutionary optimization, approximate reasoning, and ontology-based knowledge representation.

Edward H. Y. Lim received the B.A. and M.Phil. degrees from The Hong Kong Polytechnic University, Kowloon, Hong Kong.

He is currently a Researcher with IATOPIA.com for various intelligent systems research and development. His research interests include ontology-based knowledge system, intelligent agent technologies, and intelligent indoor and outdoor positing systems.

Xi-Zhao Wang (SM'04) received the Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, China, in 1998.

From September 1998 to September 2001, he was a Research Fellow with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong. Since October 2001, he has been the Dean and a Full Professor of the College of Mathematics and Computer Science, Hebei University, Baoding, China. His main research interests include learning from examples with fuzzy representation, fuzzy measures and integrals, neuro-fuzzy systems and genetic algorithms, feature extraction, multiclassifier fusion, and applications of machine learning.

Dr. Wang is an IEEE Board of Governor member in 2005 and 2007–2009); the Chair of the IEEE Systems, Man, and Cybernetics (SMC) Technical Committee on Computational Intelligence; the Editor-in-Chief of the International Journal of Machine Learning and Cybernetics; an Associate Editor of the IEEE TRANSACTIONS ON SMC, PART B: CYBERNETICS; an Associate Editor of the Information Science; an Associate Editor of the Pattern Recognition and Artificial Intelligence. He is a distinguished lecturer of the IEEE SMC Society.