Ontology-Based Semantic Retrieval for Education Management Systems

Lijun Tang[1] and Xu Chen[2]

[1]School of Mathematics and Computer Science, Ningxia University, Yinchuan, China
[2]Network Information Technology Center, Beifang University of Nationalities, Yinchuan, China

The traditional information retrieval technologies are based on keywords, and therefore provide limited capabilities to capture the conceptualizations associated with user needs and contents. As a new technology of information retrieval, *semantic retrieval* can retrieve information resource fully and precisely based on the knowledge understanding and knowledge reasoning. *Ontology*, which can well represent and reason about the domain knowledge, is proved to be very useful in the semantic retrieval. On this basis, in this paper, we propose a complete *ontology-based semantic retrieval approach and framework* for education management system. *Firstly*, we present some rules for constructing domain ontology from the education management system; *Then*, a semantic annotation method of the constructed ontology is given; *Further*, the ontology-based semantic retrieval algorithm is proposed; *Finally*, a complete framework is developed and some experiments are done. Conducted experiments show that our semantic retrieval model obtained comparable and better performance results than the traditional information retrieval technology for education management system.

Keywords: ontology, semantic retrieval, education management system

1. Introduction

Information retrieval, which is a stage in the consumer decision process during which a consumer searches for internal or external information, is used in many real applications such as databases, business, education, and so on [14]. The traditional information retrieval technology, which is based on the key word match, does not have a good support of semantic match. Its properties depend on users' understanding upon their means of application, and therefore provides limited capabilities to capture the conceptualizations associated with user needs and contents [5].

Aiming to solve the limitations of keyword-based models, the idea of conceptual search, understood as searching by meanings rather than literal strings, has been the focus of a wide body of research in the information retrieval field. As a new technology of information retrieval, *semantic retrieval* can retrieve information resource fully and precisely based on the knowledge understanding and knowledge reasoning. In particular, *Ontology* [1], [3], which comes from Philosophy and is firstly used as other meaning in the domain of Artificial Intelligence, is very useful in the semantic retrieval. It can provide a flexible and full-scale mechanism of description and storage for the share and creation of domain knowledge. Thanks to the advantages of ontologies, their potential to overcome the limitations of keyword-based search in the information retrieval context was soon envisaged, and was explored by some researchers in the Semantic Web area [16], [6], [12].

Subsequently, *ontology-based information retrieval* technologies are discussed in many studies from different aspects [5], [11], [15], [22], [13], [23], etc. A new text-mining system-based on ontology for scientific literature is developed in [15], the system's capabilities go far beyond those of a simple keyword search engine. The study of [22], which handles ontology based information retrieval, is focused on developing an ontology which makes MPEG-7/21 standards interoperable with domain and application ontologies. Moreover, the use of ontologies for information retrieval in organizational memories is discussed in [13]. Also, an ontology-based information extraction and retrieval system and its

application in the soccer domain is investigated in [11]. The comprehensive review of ontology-based information retrieval can be found in [9], [19].

In recent years, with the rapid development of internet technology and the normalization of higher education, a lot of excellent courses, on-line courses, discipline websites and internet teaching resources libraries are constructed. The amount of the teaching resources is growing very fast. With the traditional education management systems, the information retrieval technology is based on the simple matching of the key words. However, due to the different hobbies of the users, the description of the needed retrieval information is different. Traditional retrieval method of the teaching resources hardly satisfies the teachers' and the students' need of the precision and recall from the massive resources. Thus, the urgent problem of the education management system is how to retrieve the needed information from the massive teaching resources. As mentioned above, *ontology-based information retrieval* technologies may be useful for education management. With domain ontology, semantic instruction can be given in users' choice of the retrieval structure, which can help to eliminate the ambiguity of natural language, clearly define the concept and therefore make users question retrieval type better and accurately reflect users' real need of information. It is easier for users to extend and narrow the retrieval. It can also strengthen the logical function of the system. If the search of information resources is finished, it can also infer and figure out the information hidden in the text on the basis of the relevant concept and background knowledge. In [21], a brief framework of semantic retrieval based on the ontology of teaching management is initially presented, but the complete approach and framework for retrieving teaching management based on ontology, including the construction of ontology from the teaching management domain, the semantic annotation method of the constructed ontology, the semantic retrieval algorithm, detailed discussions and experiments, are missed in [21].

To this end, *in this paper, we propose a complete ontology-based information retrieval approach and framework for education management systems. Firstly*, we present some rules for con-structing domain ontology from the education management system; *Then*, a semantic annotation method of the constructed ontology is given; *Further*, the ontology-based information retrieval algorithm is proposed; *Finally*, a complete framework is developed and some experiments are done. Conducted experiments show that our semantic retrieval model obtained comparable and better performance results than the traditional information retrieval technology.

The remainder of this paper is organized as follows. Section 2 recalls some preliminaries on ontologies. Section 3 proposes a complete ontology-based information retrieval approach for education management. A framework is developed and some experiments are done in Section 4. Section 5 shows conclusions and the future work.

2. Preliminaries on Ontologies

The vocabulary "ontology" often appears in various applications. While having its roots in philosophy, the term ontology is today popular also in computer science. In general terms, ontologies are a formal, explicit specification of a shared conceptualization [20]:

- *Conceptualization* refers to an abstract model of some part of the world which identifies the relevant concepts and relations between these concepts.

- *Explicit* means that the type of concepts, the relations between the concepts, and the constraints in their usage, are explicitly defined.

- *Formal* refers to the fact that the ontology should be machine-readable.

- *Shared* means that the ontology should reflect the understanding of a community and should not be restricted to the comprehension of some individuals.

Ontologies can be defined by ontology definition languages such as RDFS, DAML+OIL, or OWL [7]. *OWL (Web Ontology language)* [17], a W3C recommendation, is to be de-facto standard for ontologies. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their

interrelationships is called ontology. OWL describes the domain knowledge by defining some classes, properties, individuals, and axioms.

In the following we give a formal definition of OWL ontologies in order to provide readers with a general understanding of the components of ontologies.

Definition 1 (*ontologies*). An OWL ontology can be formally defined as a tuple $\mathcal{O} = \{\mathcal{I}, \mathcal{P}, \mathcal{C}, \mathcal{D}, \mathcal{A}\}$:

- $\mathcal{I}$ is a set of *individuals*; Each individual is an instance of a class;

- $\mathcal{P}$ is a set of *properties*; A property can be classified into two kinds of properties: *object properties* $\mathcal{R}$ and *datatype properties* $\mathcal{T}$, the former link individuals to individuals and the latter link individuals to data values;

- $\mathcal{C}$ is a set of *classes*; $\mathcal{C}$, $\mathcal{P}$ and $\mathcal{I}$ form the primitive terms of an ontology, e.g., a class *Person* can be used to represent the set of all people, the object property *parentOf* can be used to represent the parent-child relationship, and the individual *Peter* can be used to represent a particular person called *Peter*;

- $\mathcal{D}$ is a set of *data range identifiers*; Each data range identifier is a predefined XML Schema datatype, e.g., xsd:integer and xsd:string;

- $\mathcal{A}$ is a set of axioms defined over $\mathcal{I} \sqcup \mathcal{P} \sqcup \mathcal{C} \sqcup \mathcal{D}$; For example, *subClassOf* (Person, Mammal) is a class axiom, and it denotes that the class Person is a subclass of the class Mammal. From this a reasoner can deduce that if an individual is a Person, then it is also a Mammal; Similarly, *subPropertyOf* (hasSibling, hasRelative) is a property axiom, and it denotes that the hasSibling may be stated to be a subproperty of hasRelative. From this a reasoner can deduce that if an individual is related to another by the hasSibling property, then it is also related to the other by the hasRelative property.

The semantics for OWL ontologies are given based on Description Logics [2] (Description logics, which are a family of knowledge representation languages that are widely used in ontological modeling, are the logical underpinnings of OWL). The semantics allows humans and computer systems to exchange ontologies without ambiguity as to their meaning, and also

makes it possible to use logical deduction to infer additional information from the facts stated explicitly in an ontology. The detailed syntax and semantic of OWL can be found in [17].

3. Ontology-Based Semantic Retrieval Approach for Education Management System

In this section, we propose an Ontology-based Semantic Retrieval approach for Education Management systems, called OSR-EM.

To achieve semantic retrieval under the Semantic Web environment, it is especially significant to tackle the following several key steps: (*i*) Constructing ontologies from the education management system resources, i.e., representing the education management system resources as the ontology formalizations; (*ii*) Semantic annotation of the constructed ontologies, so that the semantic information can be used in the retrieval of information resources; (*iii*) The ontology-based semantic retrieval algorithm should be provided. All of these key steps are missed in the previous work [21], and will be presented in this paper.

3.1. Construction of Ontologies From Education Management System

Ontologies are specifications of concepts and relations among them. They play a central role in Semantic Web applications by providing a shared knowledge about the objects in real world, which promotes reusability and interoperability among different modules. The quality of the ontology should be the first concern in semantic applications.

Therefore, in order to achieve semantic retrieval for education management system, a precondition here is that we can represent education management system information with ontologies first, i.e., construct ontologies by extracting knowledge from education management systems. In the following, we present some rules for constructing a basic and core ontology, which considers some main and common knowledge in the domain, and some complex ontologies can be extended from the basic ontology.

(1) Classes

In education management domain, some main concepts can be defined as *classes*, such as College, Teacher, Assistant, Lecturer, Associate Professor, Professor, Student, Master, PhD Candidate, Dean, Course, and so on. Further, classes can be grouped together to form a class hierarchy by the OWL constructor *subClassOf*. Moreover, OWL provides additional constructors with which to form classes. These constructors can be used to create so-called *class axioms*. OWL supports the basic set operations, namely *union*, *intersection* and *complement*. These are named *owl:unionOf*, *owl:intersectionOf*, and *owl:complementOf*, respectively. Additionally, classes can be *enumerated*. Class extensions can be stated explicitly by means of the *oneOf* constructor. And it is possible to assert that class extensions must be disjoint. In the following, several examples are provided to illustrate the definitions of classes.

For example, the Lecturer class can be defined as follows:

<owl:Class rdf:about="#Lecturer">
</owl:Class>

Also, the class Lecturer is a subclass of another class Teacher, and the classes Dean and Course, which are disjoint, can be defined as follows:

<owl:Class rdf:about="Lecturer">
  <rdfs:subClassOf>
    <owl:Class rdf:about="Teacher">
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="Dean">
  <owl:disjointWith rdf:resource=
  "#Course"/>
</owl:Class>

(2) Properties

In OWL, properties let us assert general facts about the members of classes and specific facts about individuals. OWL provides a powerful mechanism for defining properties. A property can be classified into two kinds of properties: *object properties* and *datatype properties*. The former link individuals to individuals and the latter link individuals to data values. Moreover,

the following mechanisms can be used to define properties.

- *Domain* and *Range*: for a property, its domain and range can be specified.

- *Characteristics*: a property may have its *characteristics* (e.g., *Symmetric*, *Functional*, *Transitive*, *inverseOf*, and etc.). If a property, P, is tagged as functional, then for all x, y, and z: $P(x, y)$ and $P(x, z)$ implies $y = z$; If a property, P, is tagged as *symmetric*, then for any x and y: $P(x, y)$ iff $P(y, x)$; If a property, P, is specified as *transitive*, then for any x, y, and z: $P(x, y)$ and $P(y, z)$ implies $P(x, z)$; If a property, P_1, is tagged as the owl:inverseOf P_2, then for all x and y: $P_1(x, y)$ iff $P_2(y, x)$

- *Axioms*: the property can be defined to be a specialization (subproperty) of an existing property by *subPropertyOf*. Also, some properties may be equivalent by *equivalentProperty*.

In the following, several examples are provided to illustrate the definitions of properties. For example, the relationships between "Student" and "Course" can be defined by the property "choose" as follows:

<owl:ObjectProperty rdf:about="#choose">
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range rdf:resource="#Course"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:ID="chosen">
  </owl:inverseOf>
</owl:ObjectProperty>

Also, each *thesis* has its *author* and *department*, which can be defined by the datatype properties as follows:

<owl:DatatypeProperty rdf:ID="*author*">
  <rdfs:domain rdf:resource="#thesis"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="*department*">
  <rdfs:domain rdf:resource="#thesis"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>

(3) Restrictions

Besides the classes and properties, OWL also provides a powerful mechanism for defining some restrictions. The *owl:allValuesFrom* restriction requires that for every instance of the class that has instances of the specified property, the values of the property are all members of the class indicated by the *owl:allValuesFrom* clause. *owl:someValuesFrom* is similar. Also, OWL permits the specification of the number of elements in a relation by Cardinality. *owl:maxCardinality* can be used to specify an *upper* bound. *owl:minCardinality* can be used to specify a *lower* bound. In combination, the two can be used to limit the property's cardinality to a numeric interval.

For example, a department must have at least ten and at most thirty members.

```
<owl:Class rdf:about="#department">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#has
      Member"/>
        <owl:minCardinality rdf:datatype=
        "&xsd;nonNegativeInteger">
        10
        </owl:minCardinality>
        <owl:maxCardinality rdf:datatype=
        "&xsd;nonNegativeInteger">
        30
        </owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

(4) Individuals

The instances in education management domain can be represented as individuals in OWL ontology. For example, a postgraduate instance "Li" can be defined as follows:

```
<postgraduate rdf:ID= "Li">
  <Birthplace rdf:datatype="http://www.w3.
  org/2001/XMLSchema#string">
    China
  </Birthplace>
  <Number rdf:datatype="http://www.w3.
  org/2001/XMLSchema#string">
    110819
```

```
  </Number>
  <Birthday rdf:datatype="http://www.w3.
  org/2001/XMLSchema#data">
    1984-06-10
  </Birthday>
</postgraduate>
```

(5) Additional rules for ontology development process

Besides the rules above, as we have known, the information and data in an education management system are usually stored as tables in databases. Therefore, in the following we provide more detailed rules to describe how to develop ontologies from the tables in education management systems. For ease of description, the structure of a table R can be formally represented as a tuple of the form $R(A_1/D_1, \ldots, A_n/D_n)$ as mentioned in well-known relational databases, which denotes that R has attributes $A_1, \ldots,$ and A_n with associated domains $D_1, \ldots,$ and D_n. In this case, the information and data in the tables in an education management system can be represented as ontologies as the following construction rules:

- Each table symbol R is represented as an OWL class R;

- Each non-foreign attribute symbol $A_i \in \{A_1, \ldots, A_n\}$ is represented as an OWL datatype property A_i;

- Each foreign key attribute symbol $A' \in \{A_1, \ldots, A_n\}$ is represented as a pair of inverse OWL object properties A' and owl:inverseOf (A');

- Each domain symbol D_i is represented as an OWL data range D_i;

- If $R(A_1/D_1, \ldots, A_n/D_n)$ has no foreign keys, then creating OWL axioms:

  Class (R partial restriction (A_1 allValuesFrom (D_1) cardinality (1))...restriction (A_n allValuesFrom (D_n) cardinality (1)));
  DatatypeProperty (A_i domain (R) range (D_i) [Functional]);

  Here, we should note that, OWL has two *interchangeable syntaxes* [17], i.e., the *RDF/XML syntax* as shown in our previous rules in (1)-(4) of this section, and the *frame-like style abstract syntax* as shown in the above OWL axioms. Here, *for ease of description and*

for the sake of space, we choose the frame-like style abstract syntax to represent OWL axioms. The detailed syntaxes of OWL can be found in [17].

- If $R(A_1/D_1, \ldots, A_n/D_n)$ has exactly one foreign key attribute $A' \in \{ A_1, \ldots, A_n \}$ that references the primary key of another table R', then creating OWL axioms:

  Class (R partial restriction (A_1 allValuesFrom (D_1) cardinality (1)) ... restriction (A_n allValuesFrom (D_n) cardinality (1)));
  Class (R partial restriction (A' allValuesFrom (R') cardinality (1)));
  Class (R' partial restriction (owl:inverseOf (A') allValuesFrom (R) mincardinality (1)));
  ObjectProperty (A' domain (R) range (R') [Functional]);
  ObjectProperty (owl:inverseOf(A') domain (R') range (R) inverseOf A');
  DatatypeProperty (A_i domain (R) range (D_i) [Functional]);

- If $R(A_1/D_1, \ldots, A_n/D_n)$ has a foreign key A' that is also a primary key, and A' references the primary key of another table R', i.e., R' subsumes R, then creating an OWL class axiom: subClassOf (RR').

- If $R(A_1/D_1, \ldots, A_n/D_n)$ has more than one foreign key $A'_j \in \{ A_1, \ldots, A_n \}$ that respectively reference the primary keys of other tables $R_1, \ldots, R_k$, where $j \in \{ 1, \ldots, k \}$ and $k \geq 2$, then creating OWL axioms:

  Class (R partial restriction (A_1 allValuesFrom (D_1) cardinality (1)) ... restriction (A_n allValuesFrom (D_n) cardinality (1)));
  Class (R partial restriction (A'_1 allValuesFrom (R_1) cardinality (1)) ... restriction (A'_k allValuesFrom (R_n) cardinality (1)));
  Class (R_j partial restriction (owl:inverseOf(A'_j) allValuesFrom (R) mincardinality (1)));
  DatatypeProperty (A_i domain (R) range (D_i) [Functional]);
  ObjectProperty (A'_j domain (R) range (R_j) [Functional]);
  ObjectProperty (owl:inverseOf(A'_j) domain (R_j) range (R) inverseOf A'_j).

Based on the observations from the rules above, the complexity of the developed domain ontology is dependent on the scale of the education management system. Here we briefly analyze the complexity of the developed domain ontology by measuring roughly the amount of work done in the ontology construction process. In general, the information and data in a real education management system (e.g., classes, properties, restrictions, and individuals) are often stored as tables in a database. That is, the scale of the education management system may be roughly tested by the scale of the tables in the database. In this case, suppose the scale of the tables in the database is $N = N_R + N_A + N_{A'} + N_D + N_C$, where N_R, N_A, $N_{A'}$, N_D, and N_C denote the cardinality of sets of table symbols, non-foreign key attribute symbols, foreign key attribute symbols, domain symbols, and other integrity constraint symbols, respectively. According to the construction rules above, the complexity of the created ontology axioms may be measured as follows: if the table has no foreign keys, then the creating times of the corresponding axioms are $N_R + N_A + N_D$ at most; if the table has exactly one foreign key attribute that references the primary key of another table, then the creating times of the corresponding axioms are $N_R + N_A + N_D + 2N_{A'}$ at most; if the table has a foreign key attribute that is also a primary key, then the creating times of the corresponding axioms are $N_{A'}$ at most; if the table has more than one foreign key attributes, then the creating times of the corresponding axioms are $N_R + N_A + N_D + 3N_{A'}$ at most; moreover, the creating times of the other axioms are N_C at most; therefore, the total creating times $T = 3N_R + 3N_A + 3N_D + 6N_{A'} + N_C < 6N$, that is, the time complexity of the ontology construction process is $O(N)$ at most. Until now, based on the similar rules above, an ontology can be constructed from a real education management system, which is an iterative development process in the ontology engineering phase. These similar steps are repeated until we end up with all information from a real education management system.

3.2. Semantic Annotations of the Constructed Ontologies

Some pages in an education management system are often dynamic, and thus after constructing the core ontology from the education management system, we need to further semanti-

cally annotate the dynamic Web pages with ontologies. This problem has been referred to as deep annotation [24], [8] that means the process of creating ontological instances for the dynamic contents by reaching out to the 'deep Web' and directly annotating the underlying dynamic Web site.

When we semantically annotate the pages, the pages can often be divided into *contents* and *functions*. In an education management system, in terms of content, a page includes three parts: the information related to the faculties, the information related to the students, and the information related to the teaching; with reference to function, a page includes some operations on information: add, update, delete, query, and show.

In the following, we provide several examples to illustrate semantic annotations of the contents and functions of the page. First, we use the constructed ontology information (e.g., Faculties, Students, Teaching, and their classes) to annotate the information related to the faculties, students, and teaching, respectively. The annotation information is inserted into the head part of the page (i.e., between <head> and </head>). For example, we annotate a page (ss.jsp) of showing the basic information of students as follows:

```
<head>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/
   22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/Ol
   /rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07
   /owl#"
   xmlns="http://www.owl-ontologies.com/
   unnamed.owl#"
   xml:base=http://www.owl-ontologies.
   com/unnamed.owl">
   <Student rdf:ID="student">
     <studentBasiclnfo
        rdf:resource="http://portal.gfkd.mtn:
        3036/portal/ss.jsp"/>
   </Student>
</rdf:RDF>
<metahttp-equiv="Content-Type" content=
"text/html;Charset= gb2312">
```

<LINK href="css/platform.css" type=text/css rel=stylesheet>
</head>

Similarly, in order to annotate the functions of the page, several object properties, named addInfor, updateInfor, deleteInfor, queryInfor, and showInfor, are created, and the domains of these properties are the classes Faculty and Student. On this basis, the above page (ss.jsp) of showing the basic information of students can be annotated as follows:

```
<head>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/
   22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/Ol
   /rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07
   /owl#"
   xmlns="http://www.owl-ontologies.com/
   unnamed.owl#"
   xml:base=http://www.owl-ontologies.com
   /unnamed.owl">
   <Student rdf:ID="student">
     <queryInfor
        rdf:resource="http://portal.gfkd.mtn:
        3036/portal/ss.jsp"/>
     <studentBasiclnfo
        rdf:resource="http://portal.gfkd.mtn:
        3036/portal/ss.jsp"/>
   </Student>
</rdf:RDF>
<metahttp-equiv="Content-Type" content=
"text/html; Charset= gb2312">
<LINK href="css/platform.css" type=text/css rel=stylesheet>
</head>
```

3.3. Ontology-Based Semantic Retrieval Algorithm

The core of the semantic retrieval is computing the similarity between the users' retrieval condition and the data documents. Based on the approaches mentioned in the previous sections, all of the data documents in an education management system can be annotated as the

semantics-based documents. Therefore, similarity between the users' retrieval condition and the data documents can be measured by the similarity between the concepts of the constructed ontologies. Currently, there are many tools which can help users edit and handle ontologies, such as Jena, Protégé, WebODE, OntoEdit, and so on [4]. Also, in our work, we use the ontology tool Jena to edit the constructed ontologies, and the similarity between the concepts of the ontologies can be computed as the following Algorithm 1:

Based on Algorithm 1, the similarities between concepts of ontologies from the education management system can be computed, and some experiment results will be showed in the following section.

4. Retrieval Framework and Examples

On the basis of the approach proposed in Section 3, we develop an ontology-based semantic retrieval framework for education management systems. In the following we briefly introduce the design and implementation of the framework. The framework is initially presented in [21], here we will introduce the framework in detail and add some experiments and the comparison of results.

Figure 1 shows the ontology-based semantic retrieval framework for education management systems. The framework adopts Jena (it is a free and open source Java framework for building Semantic Web applications [10]) as its semantic analysis tools and achieves the browse and

query of education management ontology by analyzing the ontology files. At the same time, on the ground of SPARQL language (a W3C recommendation query language [18]), a retrieval system based on natural language query is developed. The system allows users to ask questions in natural language. It also allows users to retrieve all sorts of semantic relationships and the combination of those semantic relationships concerning education management systems.
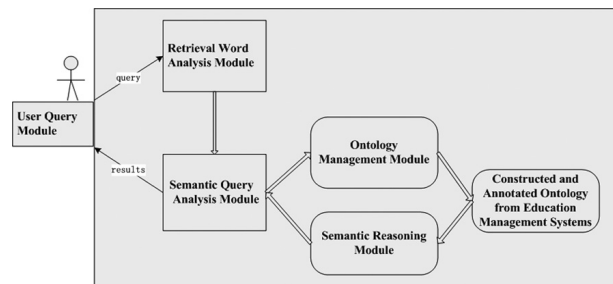

Figure 1. The ontology-based semantic retrieval framework for education management systems.

The specific retrieval process can be found in Figure 1: when a user submits a query, it will be sent to the retrieval word analysis module first, after processing and assembling the retrieval words, and then they will be passed on to the semantic query analysis module. The semantic query analysis module will check ontology base, which will further recall the constructed and annotated ontology from education management system, and then point out the semantic relationships between the query words and analyze the semantic meaning that the users want to express with the help of semantic relationship reasoning module. Now, the query will

---

Algorithm 1 // the algorithm briefly describes how to compute the similarity between the concepts
of the ontologies, which is the core of the semantic retrieval as mentioned above

---

Input: two concepts C_1 and C_2 of the ontologies.

Output: the similarity between C_1 and C_2

1. Read in the ontologies in the ontology editor tool Jena;
2. Get the OntClass (sets of superclasses) of the classes C_1 and C_2 by using the function getOntClass() of OntModel in Jena;
3. Get the lists of superclasses of the classes C_1 and C_2 by the function ListSuperClasses() of OntClass;
4. Add the lengths of two lists of superclasses and get the total length N;
5. Compute the number L of the same superclasses in the two lists of superclasses;
6. Compute the semantic similarity between C_1 and C_2 by $2L/N$ according to the formula (1) which will be presented in the following section.

be transformed into the query language of ontology, i.e., the SPARQL language [18]. The query language will be submitted to semantic query analysis module for the query of domain ontology. When the domain ontology offers query results, the semantic query results will be returned to the user by semantic query analysis module.

In more detail, the main modules in the framework in Figure 1 are introduced as follows:

- User query module: the module serves to provide the users and the system with an interface. In this module, the technology of JSP will be adopted, through which a dynamic, interactive and efficient Web server application program will be set with the help of HTML Web.

- Retrieval word analysis module: the module will carry out a preliminary treatment of the retrieval words. A user's query is first sent to the module, and then the analytical result will be passed on to the semantic query analysis module. The notional words of retrieval importance will be selected and labelled, which is the preparation of the later reasoning process of semantic relationships. In the module, the Chinese lexical analysis system ICTCLAS proposed by the Calculation of Chinese Academy of Science will be used. This tool can classify, delete, and label the retrieval words. In this way, the key words will be recognized. For example, if the user inputs a sentence: "The computer teachers of the college of math and computer science teach the basic and professional courses of computer science", the analytical result will be: "The college of math and computer science/nt 's/udel computer/n teacher/n teach/v computer science/n basic courses/n and/cc computer science/n professional courses/n".

- Semantic reasoning module: the module is responsible for creating the computer understanding upon the users' query by judging the semantic relationship of retrieval words. The Java technique and the existing free and open source framework Jena [10] will be used in this module. The query of ontology depends on SPARQL language, which is not able to analyze the information, but can search the model for those data that is directly related to each other. The aim of setting a semantic reasoning module is to infer those data with implied semantic association and get all data with related relationships, which will serve to construct the query base of SPARQL language. With the help of ontology information and related axiom descriptions, this module can get extra assertions from the basic descriptions of RDF and all semantic closures of RDF digraph through reasoning. When the user submits his query information and gets data from RDF data model, not only can he get those data contained in data model, but also the contain knowledge data from the reasoning mechanism.

In this module, the general rule reasoning mechanism of Jena package [10] is adapted to carrying out the reasoning process, and Generic Rule Reasoner Factory is introduced. Accordingly, formal rule library files are introduced to analyze the education management ontology base. At the same time, all the semantic properties of ontology base (the most important are object properties) will be selected and a semantic relationship triad will be set in accordance with the type as (Domain, Property name, Range). The semantic relationship of those labelled retrieval words will be identified according to the semantic relationship triad. The form of ontology triad is (subject, predicate, object). The users' query usually includes subject and object. All the semantic closures of RDF digraph have been acquired through the above semantic reasoning. Therefore, two triads, at most, are enough to get the retrieval result if there is retrieval result of the users' query in ontology base. Here, the Vector Space Model (VSM) is applied to construct the semantic relationships of those retrieval words. Those labelled retrieval words and the semantic properties of the semantic relationship triad are regarded as vectors. Cosine Function is used to match and judge the information. For each pair of retrieval vector and semantic property vector, the semantic similarity between C_i and C_j can be computed as the following formula (1):

$$sim\left(d_i, d_j\right) = \frac{\sum\limits_{k=1}^{n} W_{ik} * W_{jk}}{\sqrt{\left(\sum\limits_{k=1}^{n} W_{ik}^2\right)\left(\sum\limits_{k=1}^{n} W_{jk}^2\right)}} \quad (1)$$

In the formula, n stands for characteristic dimension of text vector, W_{ik} stands for the weight of corresponding feature item.

- Semantic query analysis module: SPARQL language satisfies certain paradigm and is basically made up of the sub-sentences of SELECT, FROM, WHERE, and USING. The detailed syntax of the SPARQL language can be found in [18]. For example, the following case is to find the institution where the teachers work from the given RDF diagram.

<http://www.uclab.org/Education_Management.owl/School/the College of math and computer science>
<http://www.uclab.org/Education_Management.owl/Teacher/computer teacher>"TANG"

The query includes two parts, SELECT sub-sentences and WHERE sub-sentences. SELECT identifies the variable item of the query result, there is a three sectional structure mode in WHERE sub-sentences:

SELECT ? Computer teacher
WHERE{<http://www.uclab.org/ Education_Management.owl/School/the college of math and computer science>
    <http://www.uclab.org/ Education_Management.owl/Teacher/computer teacher>
? computer teacher}

This query sentence will be delivered to ontology model. If there is triad, that is in line with semantic relationship, the property value will be returned. If not, "NULL" is returned.

- Ontology management module: the module serves to manage the domain ontology base. The education management ontology is stored as OWL files, which is made up of classes, instances and properties. To achieve ontology query, it is necessary to analyze the OWL ontology and the theory of ModelFactory is adopted to set the ontology model. The analysis work will be well done by operating the model. The following are the analysis methods:

- Analysis of classes: The classes in ontology will be recognized with the method of listHierarchyRootClasses. Then, whether there is sub-class or not will be identified

and if the answer is "yes", all sub-classes will be reached by Recursive method.

- Analysis of instances: The instances of those classes that have been got in the first step will be obtained by listsubClasses method. Then, all the names of the instances will be received.

- Analysis of properties: All the attributes will be checked to find whether they are data properties and object properties by DatatypeProperty and ObjectProperty methods. A further analysis will be carried out with the help of different kinds of their characters. If the result shows Datatype, the attribute value is Literal type, otherwise, if the result shows ObjectType, the attribute value is Resource type.

In the real applications, we need to first construct a domain ontology by extracting knowledge from an education management system as the approach proposed in Section 3, and then annotate the constructed ontology. The construction process uses the Java JDBC/ODBC API to connect and parse the database of the education management system and stores the parsed information as Java ArrayList classes, then transforms the parsed information into an ontology by using the approach proposed in Section 3. In brief, the transformations perform two kinds of operations, i.e., the transformations from symbols of the education management system (e.g., concepts, attributes, etc.) to OWL identifiers (e.g., classes, properties, etc.) and the transformations from relations and restrictions of the education management system to OWL axioms. Also, a user query interface need to be developed with the Java and JSP techniques as shown in Figure 1. Finally, a system will integrate these modules in Figure 1 to carry out the ontology-based semantic retrieval for the education management system. Moreover, as we know, currently there is no widely accepted or standard dataset of education management systems, but some common concepts and structures are contained in many education management systems. For example, many education management systems include some common concepts such as Colleges, Departments, Faculties, Administrative Staffs, Students, Undergraduate Students, Graduate Students, Courses, and so on. Also, some concepts are grouped together to form the common hierarchy relations such

as Undergraduate Students is contained in Students and Graduate Students is also contained in Students. In the following we provide an example from the real education management system in our university to further show and demonstrate that the proposed approach is feasible and the ontology-based semantic retrieval algorithm is efficient. Then a complex education management system can be handled by jointly using our approach. In detail, in the education management system of our university, some concepts are organized as tree structures such as the concept University includes many Colleges and Departments, each College includes several Research Institutions, and so on. Also, there are some relationships such as the relationship Choose between concepts Students and Courses, the relationship Teach between concepts Faculties and Courses, and so on. Each concept is a table stored in the database, and the concepts and their relationships and tree structures are implied in the forms of attributes, primary keys, foreign keys, and referential integrity constraints in the database. Based on the real education management system in our university, we constructed a brief education management ontology according to the proposed approach and framework as introduced above, which mainly includes 10 kinds of resources according to the types of resources,

and 27 classifications according to the contents of courses. For example, the media content in the ontology covers teaching video, audio, classroom records, courseware, and so on. In detail, we search the system according to the types of resources, contents of courses and texts, and a part of results are shown in Table 1. On this basis, we retrieve the ontology with the ontology-based semantic retrieval mechanism, and retrieve the original system with the keyword retrieval mechanism, and then we compare the experiment results as shown in Table 2.

As shown in Table 2, the ontology-based semantic retrieval can find more results related to the user's query, since many resources do not have standard keyword names. But at the process of constructing semantic ontology from these resources, they will be renamed and annotated. In this case, the ontology-based semantic retrieval can return more results, and expand the probability of information the user needs. Further, the retrieval effectiveness is tested based on the standard information retrieval metrics [25], i.e., *Precision* and *Recall*, and they are defined as follows. Figure 2 and Figure 3 show the comparison of *Precision* and *Recall* between ontology-based semantic retrieval and keyword retrieval.

Types of contents	Extracted contents	Extended content
Courses evaluation	Courses, Evaluation	Courses, Excellent courses, Evaluation indicators, Evaluation
Data Structure	Data Structure	Data Structure, Linear list, Stack, Tree, . . .
Programme	Programme	Programme, Code, Program design
Computer-related courses	Computer, Courses	Computer, OS, Database, Data Structure, C, Java, . . .

Table 1. The ontology-based semantic retrieval framework for education management systems.

Contents	Keyword retrieval	The correct results using keyword retrieval	Ontology-based semantic retrieval	The correct results using semantic retrieval
Courses Evaluation	16	15	34	29
Data Structure	26	24	47	43
Programme	31	31	58	55
Computer-related courses	78	69	137	132

Table 2. Comparison between ontology-based semantic retrieval and keyword retrieval.
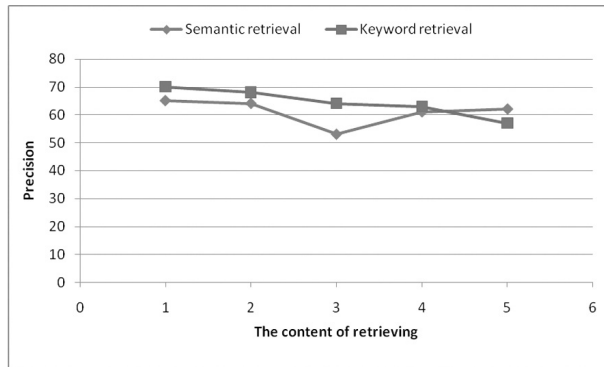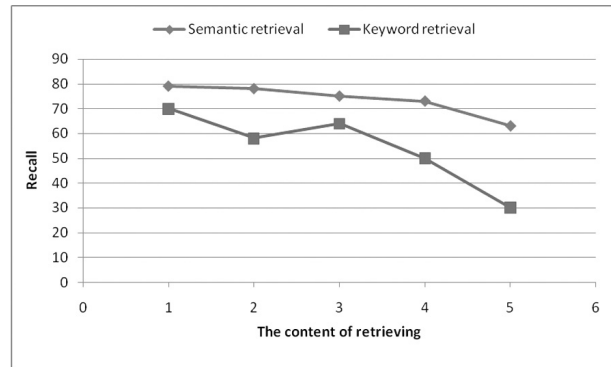
Figure 2. Comparison of precision.

Figure 3. Comparison of recall.

$$\text{Precision} = \frac{\#correct\_identified\_documents}{\#identified\_documents} \times 100\%$$

$$(2)$$

$$\text{Recall} = \frac{\#correct\_identified\_documents}{\#all\_need\_identified\_documents} \times 100\%$$

$$(3)$$

Based on the approach and framework proposed in this paper, it shows that the ontology-based semantic retrieval mechanism (including the ontology construction, annotation, semantic retrieval, and the framework) can return more useful results related to the needs of users. In particular, when users' query conditions are not the precise and single word, the recall of the ontology-based semantic retrieval will rise and well satisfy the needs of the users.

5. Conclusion and Future Work

In this paper we proposed an ontology-based semantic retrieval approach and framework for education management system. We presented some rules for constructing domain ontology from the education management system. Also, the semantic annotation method of the constructed ontology was given. On this basis, we provided a brief ontology-based semantic retrieval algorithm. Finally, a framework was developed, and the conducted experiments showed that the ontology-based semantic retrieval model could obtain comparable and better performance results than the traditional information retrieval technology for education management system.

It should be pointed out that there are some problems to be resolved in the future. One research direction involves further test and evaluation of our approach and framework with more complex real education management systems. On this basis, more comprehensive technical details and results of the framework will be provided. Besides, efficient management and usage of the ontology technology in the other real domains is highly demanded.

Acknowledgments

References

[1] T. BERNERS-LEE, J. HENDLER, O. LASSILA, The Semantic Web. *Scientific American*, **284**(5), 34–43, 2001.

[2] F. BAADER, D. CALVANESE, D. MCGUINNESS, D. NARDI, P. F. PATEL-SCHNEIDER, EDS., The Description Logic Handbook: Theory, Implementation, and Applications. *Cambridge University Press*, Cambridge, 2003.

[3] B. CHANDRASEKARAN, J. R. JOSEPHSON, B. V. RICHARD, What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems*, **14**(1), 20–26, 1999.

[4] O. CORCHO, M. FERNÁNDEZ-LÓPEZ, A. GÓMEZ-PÉREZ, Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering*, **46**, 41–64, 2003.

[5] M. FERNANDEZ, I. CANTANDOR, V. LOPEZ, D. VALLET, P. CASTELLS, E. MOTTA, Semantically enhanced information retrieval: An ontology-based approach. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web. Special Issue on Semantic Search*, **9**(4), 434–452, 2011.

[6] R. V. GUHA, R. MCCOOL, E. MILLER, Semantic search. *Proceedings of the 12th International World Wide Web Conference (WWW 2003)*, Budapest, Hungary, 700–709, 2003.

[7] I. HORROCKS, P. F. PATEL-SCHNEIDER, ET AL., From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *J. Web Semantics*, **1**(1), 7–26, 2003.

[8] S. HANDSCHUH, S. STAAB, R. VOLZ, On deep annotation. *Proc. of the 12th Int'l World Wide Web Conf.*, ACM Press, 431–438, 2003.

[9] V. JAIN, M. SINGH, Ontology Based Information Retrieval in Semantic Web: A Survey. *International Journal of Information Technology and Computer Science*, **5**(10), 62–69, 2013.

[10] JENA, http://jena.apache.org

[11] S. KARA, Ö. ALAN, O. SABUNCU, S. AKPNAR, N. K. CICEKLI, F. N. ALPASLAN, An ontology-based retrieval system using semantic indexing. *Information Systems*, **37**(4), 294–305, 2012.

[12] A. KIRYAKOV, B. POPOV, I. TERZIEV, D. MANOV, D. OGNYANOFF, Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, **2**(1), 49–79, 2004.

[13] M. LIAO, A. ABECKER, A. BERNARDI, K. HINKELMANN, M. SINTEK, Ontologies for knowledge retrieval in organizational memories. *Proceedings of the Workshop on Learning Software Organizations, Fraunhofer Institute for Experimental Software Engineering*, 11–25, 1999.

[14] C. D. MANNING, P. RAGHAVAN, H. SCHÜTZE, Introduction to information retrieval. *Cambridge University Press*, New York, NY, USA, 2008.

[15] H. M. MULLER, E. E. KENNY, P. W. STERNBERG, Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS Biology*, **2**(11), 2004.

[16] A. MAEDCHE, S. STAAB, N. STOJANOVIC, R. STUDER, Y. SURE, SEmantic portAL: the SEAL approach. *Spinning the Semantic Web*, 317–359, 2003.

[17] OWL: ONTOLOGY WEB LANGUAGE, http://www.w3.org/2004/OWL/

[18] SPARQL QUERY LANGUAGE FOR RDF, http://www.w3.org/TR/rdf-sparql-query

[19] G. SINGH, V. JAIN, Information Retrieval (IR) through Semantic Web (SW): An Overview. *Proceedings of CONFLUENCE 2012 – The Next Generation Information Technology Summit at Amity School of Engineering and Technology*, 23–27, 2012.

[20] R. STUDER, R. BENJAMINS, D. FENSEL, Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, **25**(1-2), 161–198, 1998.

[21] L. J. TANG, X. CHEN, The study of semantic retrieval based on the ontology of teaching management. *Proceedings of 2011 International Conference on Advanced in Control Engineering and Information Science, CEIS 2011*, 1555–1559, 2011.

[22] C. TSINARAKI, P. POLYDOROS, S. CHRISTODOULAKIS, Interoperability support between mpeg-7/21 and owl in ds-mirf. *IEEE Transactions on Knowledge and Data Engineering*, **19**, 219–232, 2007.

[23] T. Y. TAO, M. ZHAO, An Ontology-Based Information Retrieval Model for Vegetables E-Commerce. *Journal of Integrative Agriculture*, **11**(5), 800–807, 2012.

[24] R. VOLZ, S. HANDSCHUH, S. STAAB, ET AL., Unveiling the hidden bride: deep annotation for mapping and migrating legacy data to the Semantic Web. *Journal of Web Semantics*, **1**(2), 187–206, 2004.

[25] C. J. VAN RIJSBERGEN, A new theoretical framework for information retrieval. *Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pisa, Italy, 194–200, 1986.

Contact addresses:
Lijun Tang
School of Mathematics and Computer Science
Ningxia University
Yinchuan
750021
China
e-mail: tanglj@nxu.edu.cn

Xu Chen
Network Information Technology Center
Beifang University of Nationalities
Yinchuan
750021
China
e-mail: chenxu@nun.edu.cn

LIJUN TANG is currently working as a lecturer in School of Mathematics and Computer Science, Ningxia University, China. Her research interests include ontology and the Semantic Web.

XU CHEN is currently working in Network Information Technology Center, Beifang University of Nationalities, China. His research interests include spatio-temporal data management and the Semantic Web.