MetaData Pro: Ontology-Based Metadata Processing for Web Resources

Ting Wang[1], Ji Wang[2], Yang Yu, Rui Shen, Jinhong Liu, and Huowang Chen

National Laboratory for Parallel and Distributed Processing,
Changsha, Hunan, P.R.China 410073
[1] wonderwang70@hotmail.com, [2]jiwang@mail.edu.cn

Abstract. Metadata is the foundation of Semantic Web. The MetaData Pro project seeks to build a metadata processing platform for web resource. The system architecture has three key components-- Metadata Extraction, Ontology Management, and Metadata Retrieval. The system can automatically extract metadata about web resource: if the web resource itself contains metadata, extracts them; otherwise, automatically generates the metadata for the resource according to Dublin Core by applying automatic keyword extraction and text summarization techniques. To manage the metadata, MetaData Pro integrates Protégé to create domain ontology, makes use of HowNet to help ontology construction, and provides an ontology-based metadata retrieval.

1 Introduction

Semantic Web has complex hierarchy based on XML and RDF, in which metadata plays an important role. Metadata, the data about other data, most commonly refer to the descriptive information about Web resources in the web-age term. The web resource metadata can serve a variety of purposes, such as identifying a resource that meets a particular information need, evaluating resources suitability for use, intelligent browsing, Agent based web service, and so on. Metadata can provide the unstructured data with structures or semi-structures.

Currently many metadata languages have been developed for indexing web information resources with knowledge representations (logical statements) and storing them in web documents [1], for example, the Dublin Core[2], the PRISM[3], XMP[4], IMS Metadata[5], V-Card[6] and so on. The metadata in the web is tremendous and grows quickly. Some of them are contained in web resources such as in html, PDF, JEPG files, and some stand alone as presented in XML files. So it is important to manage them efficiently, which includes extracting and collecting them from various web resources, storing them in uniform representation, and identifying them for particular use.

The MetaData Pro is a platform developed for processing metadata in web resources. Its significant feature is to link the metadata extraction, Natural Language Processing (NLP) and ontology together to achieve continuous metadata information processing. The extraction tool searches online documents and extracts metadata contained in them, or automatically generates the metadata for the resource according to Dublin Core by applying automatic keyword extraction and text summarization

C. Bussler et al. (Eds.): WISE 2004 Workshops, LNCS 3307, pp. 34–45, 2004.

techniques. It stores various metadata in RDF [7] and provides the uniform accessing methods. To manage the metadata, MetaData Pro integrates Protégé to created domain ontology and enhanced the metadata retrieval with ontology-based term expansion mechanism. To facilitate the ontology creation, we build a tool to extract ontology from HowNet -- a Chinese-English bilingual knowledge dictionary [8].

In the rest of this paper, we will first give an overview on the architecture of MetaData Pro. After that, we will present its three key components -- Metadata Extraction, Ontology Management, and Metadata Retrieval. Finally, we will summarize the work and give the way ahead.

2 Architecture of MetaData Pro

MetaData Pro's architecture (see Fig. 1) comprises three key components: Semantic Metadata Extracting, Ontology Management, and Metadata Retrieval. At first, the web resources (including various file types: html, xml, PDF, JEPG…etc) are collected by a spider. The Semantic Metadata Extraction tool gleans metadata from the obtained resources and passes the information to the Metadata Base (MB) and stores the data in the RDF triple model. The Ontology Management tool provides ontology editing by using Protégé [9]. The ontologies outputted by Protégé are stored in the Knowledge Base (KB) in RDF model also. The Metadata Retrieval tool accepts user requests expressed in ontology vocabulary which is converted to a RDF query using a calculating engine on the Knowledge Base and retrieval the satisfied metadata in MB. All the tools are integrated into Protégé as plugins.

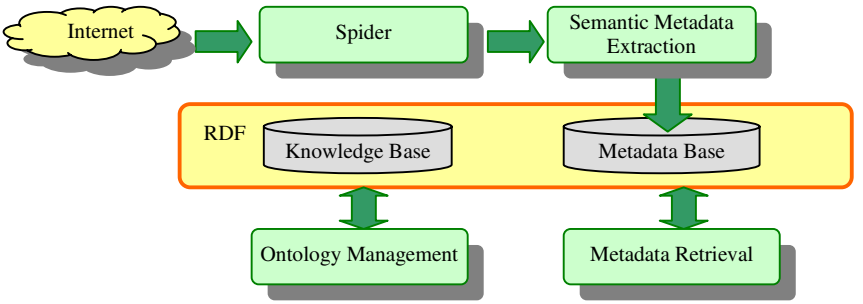

Fig. 1. Architecture of MetaData Pro

2.1 Metadata Representation

Although MetaData Pro is aimed to manage various widely used metadata such as Dublin Core, PRSIM and XMP, it in fact can extract more metadata other than these types. All the metadata satisfying one of the following conditions can be extracted automatically by MetaData Pro:

- The metadata is contained in html files and expressed as "<META>"tags.
- The metadata is contained in html files and expressed as embedded RDF content.
- The metadata is contained in xml files using RDF/XML.

- The metadata is contained in files (PDF, JEPG,…, etc) satisfying XMP standard.

In fact, Metadata Pro is sensitive to the expressing way of metadata rather than the metadata vocabulary itself. For a kind of metadata other than the above three types, if it is contained in web resources in the ways listed above, it can be fully processed by MetaData Pro. Moreover, even the expressing way changed, we can add a new extractor to capture it conveniently.

Metadata can be expressed fairly differently in various resources. A unified representation form is needed. RDF provides a framework for metadata. All the extracted metadata can be converted to RDF model and stored in database.

In MetaData Pro, both the metadata and knowledge (ontologies) are all expressed in RDF model. The storing of RDF data has two solutions: one is the RDF specified database, such as Guha's RDFDB [10]; the other is to store RDF in relational database. To achieve more scalability, we choose the later way to store both the MB and KB RDF data. To manipulate these RDF data, we apply Jena [11] to provide query language--RDQL, a query language for RDF.

2.2 Modules

The Semantic Metadata Extraction searches online documents and extracts metadata from Web resources in two ways: if the web resource contains metadata, it extracts them directly; if doesn't, it will automatically generate the metadata for the resource according to Dublin Core. When generates metadata, some are obtained with the help of HTTP and html parsing, such as Identifier, Format, Date and Title; some is obtained by applying automatic keyword extraction and text summarization techniques, such as Subject and Description.

MetaData Pro uses Protégé to create the domain ontologies. Protégé is a graphical tool for ontology editing and knowledge acquisition that we can adapt to enable conceptual modeling with new and evolving Semantic Web languages[9]. To facilitate the ontology creation, we build a tool to extract ontology from HowNet -- a Chinese-English bilingual knowledge dictionary. The concepts are converted from the lexicon entry and related according to its semantic definition and the sememe hierarchy. The ontologies are stored in KB in RDF model also.

To retrieve metadata, user requests can be expressed in the query language, which is defined as a logical expression augmented with ontology vocabulary and operators. The query expression will be converted to a RDF query using a naive calculating engine on the KB. The calculating engine provides an ontology-based term expansion mechanism. At last Jena will do the RDF query and return the satisfied metadata.

3 Metadata Extraction

Currently more and more metadata are now contained in the web resources to index web information resources with structured or semi-structured representations. There are many tools can help people to add metadata while creating web resources, for example, Adobe has integrated the XMP framework into Adobe Photoshop, Acrobat, FrameMaker, GoLive, InCopy, InDesign, Illustrator, and LiveMotion. So the resources produced by these tools, such as HTML, XML, PDF, JEPG files and so on,

will contain metadata to describe the resources itself. For the metadata contained in these web resources, MetaData Pro can extract them according to their corresponding expression methods.

However, as Semantic Web is still on its way, there are great amount of available web resources not annotated with metadata, so these resources will not be managed in the Semantic Web frame. There is a strong need to develop tools to automatically generate metadata for these resources. With the help of these tools, most of the web resources without annotated metadata can be shifted to Semantic Web frame, as well as the generated metadata will give the resources more structured descriptive information. Of course, these tools must be designed according to certain metadata standard. In MetaData Pro, Dublin Core is taken as the annotated standard for the generated metadata.

3.1 Extract Metadata from Web Resources

The metadata standards define the vocabulary used to describe the metadata, but its expression way may vary from file types to types. For each expression way, there should be a corresponding extractor.

- For the metadata contained in html files, if the metadata is expressed as "<META>" tags, the extractor will produce the RDF triple statements by adding the URL of the web document as the Subject and convert the Name-Content pairs contained in <META> tag to the Predicate and Object of RDF triple.
- For the metadata contained in html files, if the metadata is expressed as embedded RDF content, the extractor also produces the RDF triple statements by converting the value of rdf:about attribute to the Subject, and other attribute-value pairs to the Predicates and Objects of RDF triple.
- If the metadata is contained in xml files using RDF/XML, the extractor treats this case just as the above one.
- If the metadata is contained in web resource files, such as PDF, JEPG and so on, satisfying XMP standard, the extractor will first scan the document to locate the metadata according to the XML Packet format defined by XMP, and then convert the RDF/XML metadata to RDF triple statements.

All the extracted metadata is converted to RDF triple statements and stored in the MB through Jena.

3.2 Automatically Generate Metadata for Web Pages

As Semantic Web is still on its way, there is a strong need to develop tools to generate metadata for the annotated web resources. Metadata Pro provides the feature to generate metadata for the web pages (HTML, XML) not annotated with metadata, taking Dublin Core as the annotated standard for the generated metadata. Thus these resources can be shifted forward to Semantic Web frame.

Dublin Core is widely used as the vocabulary to describe the information about web resources. It defines a metadata element set consisted of 15 elements, some of which can be generated according to HTTP header, some can be got by HTML parsing, and some can be obtained by applying NLP techniques such as automatic

keyword extraction and text summarization. Table 1 shows the generated metadata element and how its value is generated.

Table 1. Generated metadata element and its value generating method

Element	Value generating method
Identifier	Using the URL of the page.
Format	Get from the content type in HTTP header.
Date	Set to the created or last modified date of the page, got from HTTP header.
Title	Set to the text extracted from the <title> tag in the html page.
Subject	Set to the text extracted from the page by keyword extraction.
Description	Set to the text extracted from the page by text summarization.
Creator	Set to "Metadata Pro" indicating it is generated by the software.

3.3 Automatic Keyword Extraction and Text Summarization

According to the specification of Dublin Core, the Subject element is used to describe the subject and keywords. Typically, a subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. The Description element may include but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content. Metadata Pro applies the NLP techniques to generate the values of the Subject and Description: using automatic keyword extraction to get the keywords from the document and using text summarization to get the summary of the document.

In Metadata Pro, automatic keyword extraction and text summarization are closely related. The former is the foundation of the later, as shown in Fig. 2.

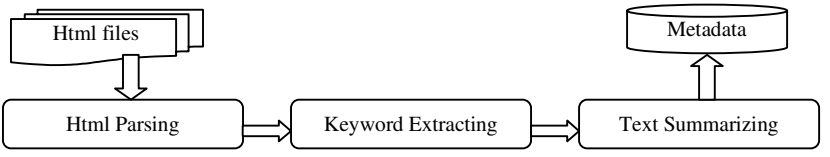

Fig. 2. The workflow of automatic keyword extraction and text summarization

The simplest way to extracting the keywords from a text is based on finding the most frequent words in the text. The basic intuition underlying this approach is that the most important concepts in the texts are likely to be referred to repeatedly, or, at least, more frequently than minor concepts [12]. Although this basic intuition is sensible, it is too simple to achieve our goal. In Metadata Pro, what we should do is to extract keywords from the html page, which is marked with tags. Many tags have certain semantic meaning which can give us the information about the structure of document. Moreover, the location of each word in the text also brings different implication for its contributes to the text [13].

To improve the extraction and summarization, five factors have been considered:

- Word Processing: Metadata Pro is aimed to process web pages in both English and Chinese, so the word processing is a little trouble. For English pages, stemming is requested when break the text into words, while for Chinese, the sentences should be segmented into words because there is no blank between the words as English do. Segmenting is an important technology for Chinese text processing.
- Stop words: the words appear in most document frequently should not be the candidates, that is to say the stop words should be filtered out.
- Frequency: the most frequent words in the text tend to be the keywords candidates; the sentences containing most keywords candidates tend to be in the summary.
- Location: the location of each word in the html document should be given different weight. For example, the text appear in TITLE tag and sub title tags (H1~H6) will certainly convey more important meaning of the document than general text. Also sentences in different position play different roles in the meaning expression. It is reported that about 85% topic sentences appear at the begin of the paragraph as well as about 7% topic sentences is the last one of the paragraph. Here we consider the following locations: title, sub title, begin of paragraph, end of paragraph, middle of paragraph, first paragraph, last paragraph and so on, each of these locations is associated with a weight w_l (where l indicates different location).
- Length of the document: we can set the number of keywords to, for example 5 or 10(5 is default in the system), but it is improper to fix the number of sentences in summary, because the lengths of web pages vary greatly. It is more rational to set the ratios of the summary compare to the total document. The ratio is user-defined, which is set as 10% in default.

Now we can give the keyword extraction and text summarization methods. Metadata Pro first parses the html files, scans the text to recognize the word. For English text, it stems each word; for Chinese text, it segments the string into words using Longest-Match method [14]. Then the stop words should be filtered out. Then we can calculate the weight for each word as follow: for each word i, its weight is

$$WordWeight_i = \sum_l w_l * N_{li} \tag{1}$$

where w_l is the associated word weight of the location l, N_{li} is the occurrence frequency of word i in the location l in the total document. So $WordWeight_i$ is the weight of word i to the document. The words with top 5 weights are taken as the keywords of the document.

After calculating the weight for each word in the document, we can produce the summary as follows: for each sentence i in the document, calculate its weight as,

$$SentWeight_i = \sum_j WordWeight_j + s_l \tag{2}$$

where $WordWeight_j$ is the weight of word j in the sentence calculated as formula (1), s_l is the associated weight of the sentence's location l, So $SentWeight_i$ is the weight of sentence i to the document. The sentences with top N weights are taken as the summary of the document, where N = the ratio of summary * (the number of sentences in the document). The selected sentences are arranged according to their original order of their occurrence in the document. Of course, the reduplicate sentences must be deleted.

4 Ontology Management

4.1 Using Protégé-2000

Protégé [9] is an extensible, platform-independent environment for creating and editing ontologies and knowledge bases. It is a tool which allows the user to construct the domain ontology, customize data entry forms and enter data. It is also a platform which can be extended with graphical widgets for tables, diagrams, animation components to access other knowledge-based systems embedded applications. Protégé can also be a library which multiple applications can share to access and display knowledge bases.

Metadata Pro fully takes these advantages of Protégé by applying it in two ways:

- Using it as a visual tool to construct domain ontology.
- Taking it as an integrated platform for metadata extraction and retrieval tools.

4.2 Extract Ontology from HowNet

Knowledge acquisition is always the bottleneck of AI applications. Before the Semantic Web becomes practical and useful, there is a critical barrier we must breakthrough -- large-scale ontology construction. Two main approaches can aid this work [15]. The first one facilitates manual ontology engineering by providing natural language processing tools, including editors, consistency checkers, mediators to support shared decisions, and ontology import tools, such as Protégé. The second approach relies on machine learning and automated language-processing techniques to extract concepts and ontological relations from structured and unstructured data or text such as OntoLearn. The later approach causes more and more research interest. There are many machine readable structured resources, for example, various lexicons or dictionaries, such as WordNet which used in many works related to ontology [15]. Metadata Pro makes use of HowNet, a Chinese-English bilingual knowledge dictionary to help ontology construction.

HowNet is an on-line common-sense knowledge base unveiling inter-conceptual relations and inter-attribute relations of concepts as connoting in lexicons of the Chinese and their English equivalents [8]. It has been widely used to support semantic analysis [16]. As a knowledge base, the knowledge structured by HowNet is a graph rather than a tree. It is devoted to demonstrate the general and specific properties of concepts. In HowNet, every concept is an entry, comprising four items which are all made up of two portions joined by the "=" sign. The left hand side of the "=" sign is the data field, while the right hand side is the data value. The items are arranged as:

W_X= word / phrase form

G_X = word / phrase syntactic class

E_X = example of usage

DEF = concept definition

Now the dictionary provide both Chinese and English knowledge, so here X can be either Chinese(C for short) or English (E for short). For example, Fig. 3 shows the entries for the concept *doctor*, *hospital*, *invalid* and *patient*:

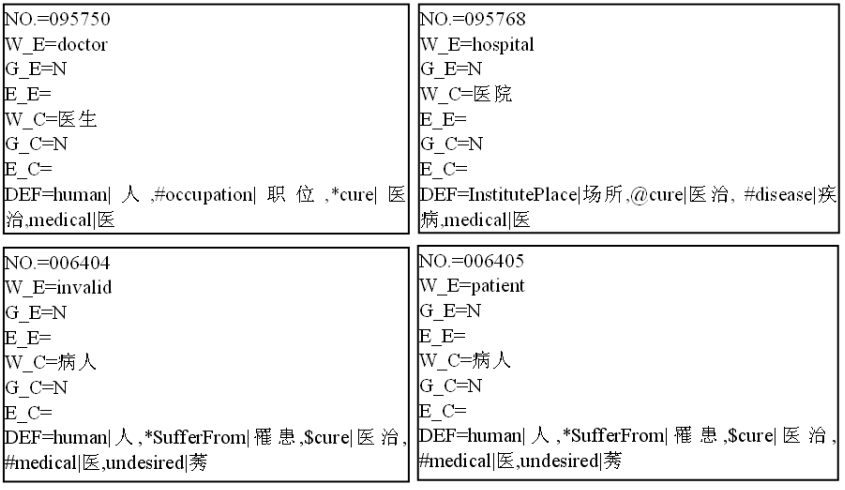

Fig. 3. Entries for the concept *doctor*, *hospital* , *invalid* and *patient* in HowNet

The entries numbered as 006404 and 006405 has the same DEF, both of them express the same concept which can be express in Chinese as word "病人" and "patient" or "invalid" in English. In both languages, its syntactic classes are all N which means noun. The most important part of every concept entry is DEF, which should include at least one feature expressed in sememe (the basic semantic unit in HowNet) and the first item in the DEF is the main feature. DEF also can express the relations between the concepts by specifying event role information. As shown in the example, expression "#medical|医" means the concept is co-relation to the concepts whose DEF contain the "medical|医" feature. Here, symbol *, $, @ and # are relation marks. The relations represented by them are shown in Table 2.

Table 2. Relations represented by the marks

Symbol	Relation
*	*agent-event*
$	*patient-event*
#	*concepts co-relation.*
@	*location-event*

The sememe system in HowNet is organized in Hypernym-Hyponym hierarchy. Fig. 4 gives a brief view of it. Together with the event role relationship expressed in DEF, there will be 16 relations between the concepts in HowNet, such as Hypernym-Hyponym, synonym, antonym, part-whole, agent-event, instrument-event and so on. So the concepts in the dictionary are connected as a network by those relations (this is why the dictionary is called HowNet). It should be note that, the relations between the concepts are not directly at concept level but established on sememe system. So if we want to extract ontology from the dictionary, we must infer from the DEF, using the sememe features and relation marks defined in it.

Fig. 4. A brief view of sememe hierarchy

To extract ontologies, there are two tasks:

- Extract concepts. In HowNet, different entries can represent the same concept, these entries has the same DEF, for example, the entries of invalid and patient in Fig. 3. We must first group the entries into concepts by the DEF.
- Extract the relations between concepts. The Hypernym-Hyponym relation is implied by main features of concepts. For example, the DEF of concept *tiger* is "beast|走兽", and the concept *mammal* is defined as "AnimalHuman|动物", from the sememe hierarch shown in Fig. 4, we can create a Hypernym-Hyponym relation between the *mammal* and *tiger*. The event role relationships expressed in DEF can also bring relations between concepts. As a more complex example, the concepts and their relations extracted from the entries shown in Fig. 3 are:

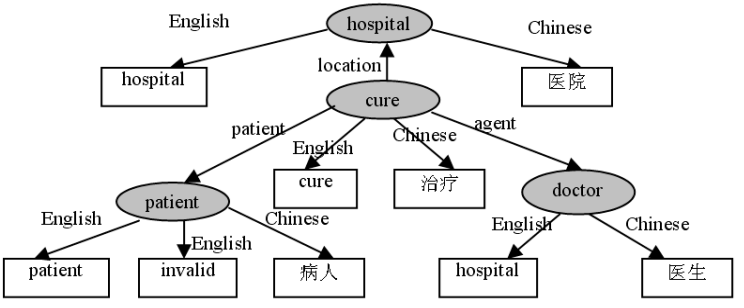

Fig. 5. The concepts and relations extracted from Fig. 3

In Fig. 5, the nodes (drawn as ovals) represent concepts and arcs represent named properties or predicate or relations. Nodes that represent string literals will be drawn as rectangles. The word in ovals nodes is just a label of the concept, while the words expressing the concept in English and Chinese are set as the value of corresponding properties (such as the *English* and *Chinese* relations).

The extracted ontologies are stored in KB in RDF model. The ontologies in KB can be used to express the metadata query, which is defined as a logical expression augmented with ontology vocabulary and operators.

5 Metadata Retrieval

5.1 Concept-Based Retrieval

Current information retrieval tools mostly are based on keyword search, which is unsatisfied because of its low precision and recall. However, if we consider the query words as concepts rather than literals, then we can retrieve relevant documents even if they do not contain the specific words used in the query. Recently concept-based information retrieval tools have been developed in both academic and industrial research environments, even some of them offer search facilities for the web [17].

In Metadata Pro, metadata itself is a retrieval object, for example, the value of the RDF statement is often been queried. In fact the metadata retrieval is quite like general text information retrieval for the metadata itself is text.

For concept-based information representation and retrieval, the key issue is how to represent the concepts. Most works treat concepts as thesaurus, some also use co-occurrences of words to present concept related words [17]. We think all these methods are reasonable but they need and could be used synthetically rather than individually. Ontology may be the integrated bed to represent the knowledge. As shown in Fig. 5, the words representing the same concept can be described as the properties of the concept. Moreover all different relations of words can be represented in the ontology by the means of concept-properties model. So in Metadata Pro, we implement concept-based retrieval by using ontology to express the user query request. A query language is designed based on the common logical expression augmented with ontology vocabulary and operators.

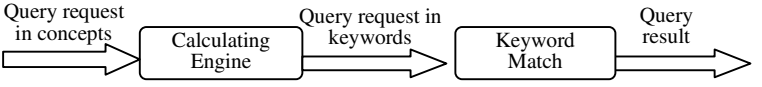

Fig. 6. The workflow of metadata query

5.2 Query Language

In Metadata Pro, the metadata retrieval tool consists of a query language and a calculating engine. With the query language, query request will be represented in the concepts or relations described in KB. This request will be converted to the final search request – Boolean expression on keywords, by the calculating engine using ontologies from KB. At last, the keyword match is run on metadata, and the related data will be fetched out (Fig. 6).

Because all the metadata is stored in RDF triples, the retrieval request can specify both the Predicate and Object, that is to say, user can specify both the Predicates to indicate the search scope and the criteria which the Object value should match:

Predicate = <Predicate List>

Object = <Criterion-expression>

where <Predicate List> specifies the Predicate set, and <Criterion-expression> is defined as:

<Criterion-expression > ::= < Item > | <Criterion-expression > or <Item >

<Item> ::= < Factor> | <Item > and <Factor >
<Factor> ::= <Criterion-expression>)|<Ontology-factor>| not <Criterion-expression >
<Ontology-factor> ::= <Concept> | literal | relation (<Ontology- factor >)
<Concept> ::= class | instance

The bone of the query language is based on the common logical expression, but the basic <Factor> has been augmented with ontology vocabulary and operators. The <Factor> can be concepts, literals (keywords) and relations.

When the query request in concept is converted to query request in keywords, the ontology vocabulary and operators in the query will be interpreted as below:

- First, calculating the *relation(<Ontology-factor>)* sub expression, it will be interpreted as the classes or instances which have the relation with the concept represented by *<Ontology-factor>*. Repeat the step until all the relations have been calculated. Now ontology vocabulary in the query is just classes, instances or literals (keywords).
- Secondly, all the classes and instances in the query will be replaced by the labels of the class and instance.
- Till now, the query is converted to common Boolean expression on literals (keywords), which can be processed by Jena's RDQL to retrieval RDF data.

Taking the ontology in Fig. 5 as example, if the query request is to find the metadata whose object value matches the class *patient*, the request can be refined and described in the query language as follow:

Predicate = *

Object = English ({ patient }) or Chinese ({ patient })

where { patient } means the concept patient, and the query means to search all the predicate types and find the triples whose Object contains words "patient" or "invalid" or "病人". This example demonstrates that with the query language, both thesaurus based and cross language search can be implemented on the ontology representation. Of course, more complex criterion expressions can be constructed, even related concepts retrieval can be implemented by using the relations between the concepts to specify the user query.

6 Conclusion

Metadata plays an important role in Semantic Web. The MetaData Pro is a platform for the processing of web resource metadata. It provides metadata extraction, ontology management, and metadata retrieval. The MetaData Pro project links the metadata extraction, NLP and ontology together to achieve continuous metadata information processing. The system has been implemented on the base of Protégé-2000. Protégé-2000 itself is used to construct ontologies. Both KB and MB are stored in RDF triples in relational database. A bridge has been developed to link the Protégé-2000 and RDF database by applying Jena. Both the metadata extraction and retrieval modules have been integrated into Protégé as its plugin tables.

We will develop more powerful and sophisticated technology. This problem can also lead to an important work--information structuralization. Metadata is normally understood to mean structured data about various resources that can be used to help support a wide range of operations, so it can be undertaken the responsibility to

structure web information. We plan to further investigate and develop tools to improve the structural extent of web information by converting them into high quality metadata.

Acknowledgement. This research is supported in part by the National Natural Science Foundation and the National High Technology Research and Development Program (2002AA116070) of China.

References

1. James Mayfield and Tim Finin: Information retrieval on the Semantic Web: Integrating inference and retrieval. SIGIR Workshop on the Semantic Web, Toronto, 1 Aug. (2003)
2. Dublin Core Metadata Element Set 1.1: Reference Description, http://dublincore.org/documents/dces/
3. PRISM Specification, http://xml.coverpages.org/prismv1b.pdf
4. ADOBE XMP. http://partners.adobe.com/asn/developer/xmp/pdf/MetadataFramework.pdf.
5. IMS Learning Resource Meta-data Specification. http://www.imsglobal.org/metadata/index.cfm
6. vCard: The Electronic Business Card, http://www.imc.org/pdi/vcard-21.txt.
7. Resource Description Framework Model and Syntax Specification, http://www.w3.org/TR/REC-rdf-syntax
8. HowNet. http://www.keenage.com/
9. Natalya F. Noy, Michael Sintek, Stefan Decker, Monica Crubézy, Ray W. Fergerson, and Mark A. Musen. Creating Semantic Web Contents with Protégé-2000. IEEE Intelligent Systems, March/April (2001)
10. R.V.Guha. rdfDB : An RDF Database. http://www.guha.com/rdfdb/
11. Jena 2 - A Semantic Web Framework. http://www.hpl.hp.com/semweb/jena.htm
12. Paolo Tonella, Filippo Ricca, Emanuele Pianta and Christian Girardi. Using Keyword Extraction for Web Site Clustering. Proc. of WSE 2003, 5th International Workshop on Web Site Evolution, Amsterdam, The Netherlands, September 22 (2003)
13. H.P.Edmundson. New Methods in Automatic Extracting. Journal of the ACM, Vol. 16(2) (1969)
14. Kaiying Liu. Chinese Text Segmenting and Tagging. Beijing, the Commercial Press (2000)
15. Roberto Navigli and Paola Velardi. Ontology Learning and Its Application to Automated Terminology Translation. IEEE Intelligent Systems, January/February (2003)
16. Xuan Qi, Ting Wang and Huowang Chen. Research on the Automatic Semantic Tagging Method. Journal of Chinese Information Processing, Vol.15, No.3 (2001) 9-15
17. H-M. Haav and J. F. Nilsson. Appr.ches to Concept Based Exploration of Information Resources. In: W. Abramowicz and J. Zurada (eds.): Knowledge Discovery for Business Information Systems. Kluwer Academic Publishers (2001) 89-109