

# Automated Semantic Annotation and Retrieval Based on Sharable Ontology and Case-based Learning Techniques

Von-Wun Soo, Chen-Yu Lee, Chung-Cheng Li, Shu Lei Chen and Ching-chih Chen\*

*Department of Computer Science, National Tsing Hua University, HsinChu, 30043, Taiwan*

*\*Graduate School of Library and Information Science, Simmons College, Boston, MA 02115, USA*

*{soo;leoli}@cs.nthu.edu.tw; {g916711; g916723}@oz.nthu.edu.tw; chen@simmons.edu*

## Abstract

*Effective information retrieval (IR) using domain knowledge and semantics is one of the major challenges in IR. In this paper we propose a framework that can facilitate image retrieval based on a sharable domain ontology and thesaurus. In particular, case-based learning (CBL) using a natural language phrase parser is proposed to convert a natural language query into resource description framework (RDF) format, a semantic-web standard of metadata description that supports machine readable semantic representation. This same parser also is extended to perform semantic annotation on the descriptive metadata of images and convert metadata automatically into the same RDF representation. The retrieval of images then can be conducted by matching the semantic and structural descriptions of the user query with those of the annotated descriptive metadata of images. We tested our problem domain by retrieving the historical and cultural images taken from Dr. Ching-chih Chen's "First Emperor of China" CD-ROM [25] as part of our productive international digital library collaboration. We have constructed and implemented the domain ontology, a Mandarin Chinese thesaurus, as well as the similarity match and retrieval algorithms in order to test our proposed framework. Our experiments have shown the feasibility and usability of these approaches.*

## 1. Introduction

Image retrieval research has been on-going for sometime. Two major paradigms are: Text-based metadata image retrieval and content-based image retrieval [1-5]. Text-based approaches, based on the keyword match of the text metadata description of images with the text queries, are usually relatively simple and easy to use but have their limitations when retrieving images that require subtle query expressions or domain knowledge. Content-based approaches, on the other hand, usually retrieve relevant images based on similar features of color, texture, shape, and spatial relations among image contexts. They often require advanced image processing and pattern recognition techniques. The low

retrieval precision and difficulty to formulate an exact feature query are the major drawbacks of this second approach. To overcome these drawbacks, semantic-sensitive content-based image retrieval techniques have been introduced [5]. In this paper, we focus on the text-based image retrieval paradigm. Traditional text-based information retrieval systems or search engines usually are based on keyword matching techniques [6-10]. Although widely used nowadays, they usually suffer from the so called "too many or nothing" problem for various reasons. One common reason is that the users might not have complete domain knowledge and often cannot specify appropriate and exact keywords for a valid query. The other is that the target documents are expressed in terms of plain-text format that is hard for the search engine to parse; thus it is difficult to understand the semantics of the documents during the retrieval process.

To address these problems, we propose the semantic annotation approach. In other words, we annotate documents to be retrieved with semantic tags that are defined and derived from a set of domain concepts or schemes called domain ontology and thesaurus so that the information retrieval can be conducted to some extent at the abstract "semantic" level instead of at the purely syntactic keyword matching level. However, this often leads to one or more of four major difficulties:

- 1) Different annotators or domain experts might use a different ontology and so end up with different annotation results,
- 2) Converting a user natural language query into semantic schema requires one to have a significant amount of domain knowledge for syntactic grammar analysis on the query language,
- 3) The manual annotation of a large amount of descriptive metadata of images is a laborious task, and
- 4) Matching a query instance with each annotated image description can be extremely inefficient and tedious particularly when the size of the images is large.

Our previous work [11] proposed a conceptual framework to remedy the first problem mentioned above. It adopted the sharable ontology concept of the semantic web [12] that enables the sharing of the domain knowledge on the web with a standard and uniform representation. WWW consortium (W3C) has

recommended several specifications and standards for web ontology languages based on XML (Extensible Markup Language) such as RDF/RDFS (Resource Description Framework/Schema) [13-20], DAML+OIL [21] and OWL [22,23], etc. These languages provide a well-defined set of relational terms essential for building domain concepts, and for serving as a website resource to be referred and shared by other domain ontology resources in terms of name spaces. RDF, a web standard for expressing metadata based on XML, provides interoperability among different platforms and allows knowledge exchange in machine-readable format on the Web. It represents semantic relations as an information resource in terms of a triple of Subject/Resource, Predicate/Property, and Object/Literal. For example: a sentence, “A general wears an armor,” can be described as a triple {General, Wear, Armor} where “General”, “Wear”, and “Armor” correspond to the Subject, Predicate and Object respectively in the RDF schema. By allowing domain ontology and thesauri to be shared by the annotators (content provider) and the users (content consumer), we have designed intelligent software agents [9-11,24] to retrieve images by matching the user’s query descriptions with the image descriptions using the same domain ontology expressed in terms of RDF instances and RDF schemas. Since the software agents use the same domain ontology and thesaurus as those used by the original annotators to interpret the annotations during the retrieval of images, it is more likely that we can avoid the mismatch problems of keywords and domain concepts. When compared with the traditional keyword match methods, we have found that the use of this method has enhanced both the recall and precision of image retrieval. Yet, this method has relied heavily on manual annotation of images using a visualization tool and a rough natural language parser to convert metadata descriptions of images and the query phrases into RDF-based annotated image descriptions and query schema respectively.

In order to overcome the other three difficulties mentioned above, we have further proposed a case-based learning approach and a conceptual clustering algorithm, to address specifically the following three main questions:

- 1) How to automatically convert a natural language query phrase into a RDF one?
- 2) How to perform automatic semantic annotation using the textual metadata descriptions of images and thus automatically convert them into RDF instances?
- 3) How to design a fast similarity matching method to match a query instance with a huge number of images in the image base in terms of RDF descriptions?

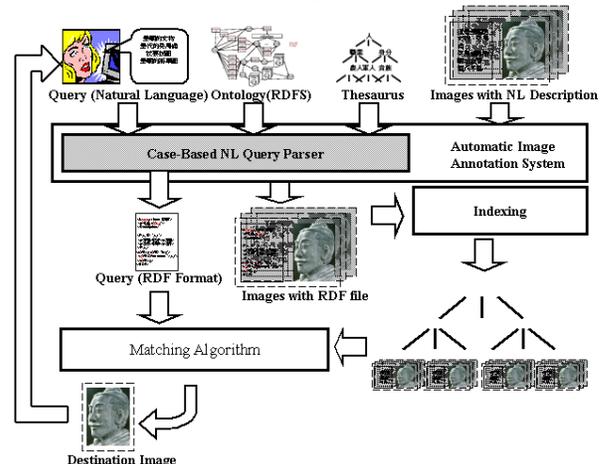
We have designed an automatic annotation technique using case-based learning to address the first and second issues and implement a conceptual clustering method to deal with the third one. Our test-bed images are the historical images of the terracotta soldiers taken from Prof.

Ching-chih Chen’s “First Emperor of China” CD-ROM [25]. She also has provided the simplified version of the metadata descriptions of these images. Before the experiments, we have translated these metadata descriptions together with additional verbal ones into Mandarin Chinese texts and then conducted automatic semantic annotations on these texts with the aid of a Chinese thesaurus and a domain ontology.

We shall provide a brief overview of our system in Section 2; and describe the case-based learning for a natural language query parser in Section 3, automatic semantic annotation approaches in Section 4, the automatic indexing and structural matching for semantic image retrieval in Section 5, and the experimental results in Section 6. We shall present our conclusions and discuss our potential future work in the final Section 7.

## 2. System overview

Our overall image retrieval system based on sharable ontology is illustrated in Figure 1. This sharable ontology, expressed in terms of RDF schema together with a sharable thesaurus, provides an aid to convert a user query into a sequence of semantic codes. The user’s natural language (NL) query, parsed by a case-based NL query parser, is converted into a RDF query instance. On the right hand side of Figure 1, the images are first fed into an automatic annotation system that converts the NL



**Figure 1. Sharable Ontology Image Retrieval System** descriptions of images into image RDF instances. They are then classified into clusters for later retrieval by automatic indexing based on the RDF triple descriptions of the images. A matching algorithm then matches the RDF query instance with the indexed image clusters and finds the most similar image in terms of RDF descriptions.

For the Mandarin Chinese Thesaurus we have augmented it with domain specific lexical items including the names of historical figures, articles, locations, countries, etc for the terracotta soldiers in Qin dynasty. The total size of the thesaurus is now more than 70,000

terms that is organized in a semantic hierarchy. These terms basically are common ones. For the purpose of this project, we have added some proper names into our thesaurus, including 225 names of ancient Chinese kings and heroes, and 178 historical locations although they are seldom used in the image retrieval domain. The hierarchy is divided into 4 levels with the first level the most abstracted layer consisting of words, such as Person, Article, Action, etc. and the fourth level consisting of the synonyms of words. There are 12 categories/concept words in the 1st level, 94 in the 2nd, and 1428 in the 3rd. For example, a word “general” coded as “AE1004” can be separated into four sub-codes as “A”, “E”, “10”, and “04”. Here “A” is the code for “Person”, “AE” stands for “Career” of a person, “AE10” associates with the “Ranks in Military”, and “AE1004” means a specific rank of a military officer such as “general” or “commander”. We also have developed a domain ontology specifically for describing historical images [26]. This domain ontology, in contrast to the thesaurus, defines domain concepts and schemas in terms of classes and properties. Currently there are 6 classes and 99 properties for the terracotta soldier domain that define the objects and relations among the objects and data.

### 3. Case-based learning for implementing a natural language query parser

A full-fledged NL parser for Mandarin Chinese needs a full set of grammar and semantic domain models. It needs also to assign thematic roles of constituents correctly in the parser tree in order to enable the conversion of the user’s query into a proper corresponding RDF instance. This is not an easy task. Case-based learning [26-28] is suitable for learning regularities where domain expert rules are difficult to express or acquire. Our hypothesis is that if we could have a case base created by collecting a set of query phrases with their corresponding RDF instances that have been correctly converted, then by looking for a similar query phrase in the case base, a new query phrase can be converted into a similar corresponding RDF instance. Thus, a case-based query parser consists of four functional modules: 1) a Pre-processor module, 2) a Similarity matching module, 3) a Prediction module and 4) a Memory update module. The process and dataflow of our prototype of the Case-based learning NL query parser is shown in Figure 2.

#### 3.1. The pre-processor module

With the aid of a thesaurus, a natural language query phrase is segmented into words. In other words, the query phrase can be divided into several meaningful word

segments that are attached with semantic codes provided by the thesaurus. For example, a specific phrase, “A Qin Dynasty general wore an armor” can be converted into four segments -- “Qin Dynasty” “a general”, “wore”, and “an armor”. The thesaurus will assign a semantic code for each word segment; thus “a general” will be assigned a semantic code “AE1004.” With the aid of the ontology, the subject in this case is a “person”, and the RDF triple expression is {“person”, “status”, “general”}. The rest of the RDF triples created are: {“person”, “wore”, “armor”}, {“person”, “period”, “Qin Dynasty”}. In this case, the “person” is common among all RDF triple expressions. The domain ontology about a “person” in terms of an RDF schema has the “status”, “wore”, that is a “person” and “period” attributes that have been constructed at the ontology implementation stage.

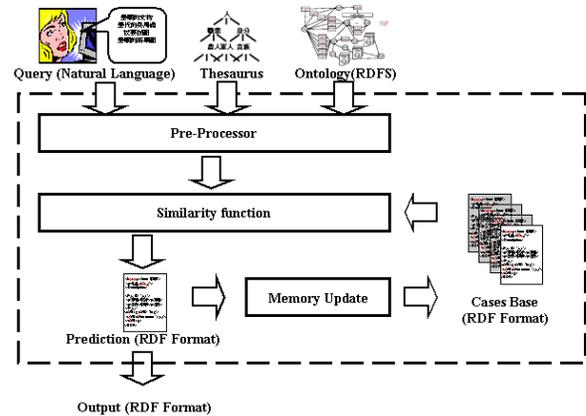


Figure 2. The CBL Natural Language Query Parser

#### 3.2. The similarity module

This module conducts the similarity match between a code sequence of a new incoming query and the code sequences of old query phrases in the case base. When we compare two similarity match algorithms -- a Most Common Subsequence Algorithm (MCS) and a Multiple Layer Recursive Matching (MLRM) algorithm, we found that the MLRM is more effective. We further elaborate on this in the following sections.

##### 3.2.1. The MCS algorithm

In the MCS algorithm, the inputs are two semantic code sequences: a query code sequence (QCS) with word length  $n$  and a case code sequence (CCS) with word length  $m$  that are denoted as:

$$QCS = \{C_0, C_1, \dots, C_n\} \text{ and } CCS = \{C'_0, C'_1, \dots, C'_m\}$$

where  $C_i, C'_j$  are the semantic codes of word  $i$  in QCS and word  $j$  in CCS, respectively.

The MCS algorithm, described in Algorithm 1, uses the

similarity scoring function to find a case code sequence (CCS) that is the most common between this sequence and the query code sequence (QCS). Two codes are considered as a match if their similarity is greater than a certain defined threshold. We define a similarity function  $Code\_S(C_i, C_j')$  to calculate the abstract similarity between the semantic codes  $C_i$  and  $C_j'$  in the thesaurus. In our Chinese thesaurus the semantic codes currently have four semantic levels as already stated. For example, for semantic codes “AE0205”, it belongs to four semantic categories -- “A”, “AE”, “AE02”, and “AE0205” -- from general to specific. Therefore two codes “AE0205” and “AE0304” are considered as the similar semantic codes if the threshold is set at 3 or 4 since they both have similar Level 1, “A” and Level 2, “AE”, but are not considered as similar if the threshold is set at 1 or 2. However, semantic codes “BE0204” and “AE0204” are not matchable because at the semantic category at Level 1, “A” and “B” are different. Likewise, in Algorithm 1 we also use a similarity function  $Seq\_S(QCS, CCS)$  to calculate the similarity scores between two code sequences QCS and CCS based on accumulation of individual pair of code comparisons.

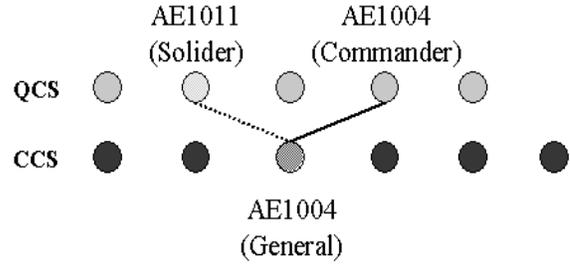
```

MCS(QCS[C0, C1, ..., Cn], CCS [C'0, C'1, ..., C'm], e)
{
  A=0, B=0;
  While (A ≤ n and B ≤ m)
  {
    For (i from A to n)
    For (j from B to m)
    If (Code_S(Ci, C'j') > e)
    {
      Seq_S(QCS, CCS) += Code_S(Ci, C'j');
      A = i + 1; B = j + 1; Break to While;
    }
  }
  return Seq_S(QCS, CCS);
}

```

**Algorithm 1. The MCS algorithm**

MCS is sensitive to the threshold. Yet, it is not easy to select a proper threshold. By decreasing the threshold, MCS might match with a high similarity code while miss the exact match code that follows. For example, as shown in Figure 3, the dotted line indicates that a matching pair with similarity greater than the threshold (the general matches with a solider) is detected before an exact match (a general matches with a commander) that is indicated by a solid line. On the other hand, if we increase the threshold too much, it might miss a pair that is a real match in the domain but cannot simply pass the threshold. It also causes the case-based learning parser to be rigid and unable to cope with various new queries that might match with an old case.



**Figure 3. Two mismatch problems in MCS algorithm**

**3.2.2. The MLRM algorithm**

The MLRM algorithm is described in Algorithm 2.

```

MLRM(QCS[C0, C1, ..., Cn], CCS [C'0, C'1, ..., C'm], ek)
{
  If (either QCS or CCS is empty or no more higher layer)
  return 0;
  A=0, B=0;
  While (A ≤ n and B ≤ m)
  {
    For (i from A to n)
    For (j from B to m)
    If (Code_S(Ci, C'j') > ek)
    {
      Seq_S(QCS, CCS) += Code_S(Ci, C'j') * Weightk;
      Seq_S(QCS, CCS) += MLRM(QCS[CA, ..., Ci-1],
        CCS[C'B, ..., C'j-1'], ek+1);
      A = i + 1; B = j + 1; Break to While;
    }
  }
  Seq_S(QCS, CCS) += MLRM(QCS[CA, ..., Cn],
    CCS[C'B, ..., C'm'], ek+1);
  return Seq_S(QCS, CCS);
}

```

**Algorithm 2. The MLRM algorithm**

Like MCS, MLRM accepts the same input QCS and CCS sequences, but the similarity match is recursively computed at multiple layers with different thresholds ( $e_k$ ) and similarity weights ( $Weight_k$ ). MLRM attempts to find a node pair from QCS and CCS that has the highest match (with initial threshold  $e_1$  at layer 1). The highest match pair essentially divides the sequences into two subsequences. And then MLRM continues the match process until one of the subsequences becomes empty by recursively matching each subsequence with a lower threshold at the next layer. The thresholds and weights at each layer are empirically and heuristically determined to reflect the relative importance of the similarity match between two code sequences. We tentatively set the thresholds  $e_1, e_2, e_3$  and  $e_4$  as 4, 3, 2, 1 at each layer respectively because we wish to control the semantic code in QCS to match all four semantic levels. The determination of the similarity weights at each layer is

also very intuitive. What we hope is to have the effect that each weight assigned at a lower layer should be higher than that at the higher layers while the weight sum at any two higher layers should be greater than the weight at a lower layer. As a result, this process will result in the decrease of the relative importance of any single similarity match, and will guarantee that the relative importance of any two similarity matches at higher layers is greater than that of a single similarity match at a lower layer. It also implies that the setup of the weight at each layer should satisfy mathematically: for all  $i$ ,  $w_k > w_{k+j}$  and  $w_i < w_{i+j} + w_{i+k}$  where  $j \geq 1$  and  $k \geq 1$ . Accordingly, we tentatively set the weights  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$  as 10, 9, 8, and 7 respectively in the experiment.

Figure 4 illustrates MLRM with an example. At Step 0, a pair of nodes is found (connected with a dark line) and divides the sequence into two subsequences. Each subsequence is recursively processed by MLRM as indicated in Step 1. In Step 1, two additional similarity pairs are found and they are further divided into finer subsequences to be processed recursively at Step 2. MLRM stops when no more similarity pairs can be found or no more nodes are left in the subsequences.

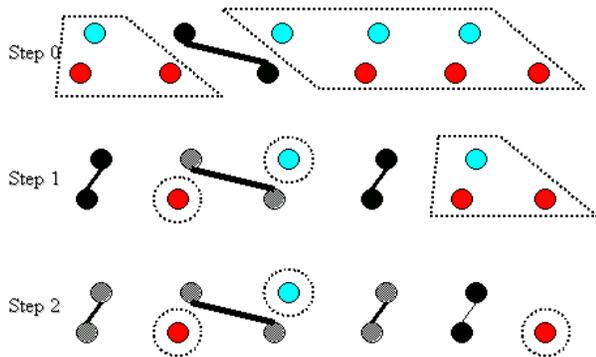


Figure 4. An example of recursive matching in MLRM

### 3.3. The prediction module

After finding the most similar case by the similarity match algorithm, a predicted RDF instance with descriptors associated with semantic codes provided by the thesaurus and ontology can be generated. For example: If a new query, “A chancellor wore a uniform” is most similar to the old query, “A Qin-dynasty general wore an armor” in the case base (the RDF description of the old query is {Person, Status, General}, {Person, Wear, Armor}, {Person, Period, Qin-dynasty}), the prediction module finds the match for this associated pair -- (Chancellor, General), (wear, wear) and (uniform, armor) among the words in the two queries. It then allows the roles -- {subject, predicate object} -- of the words in RDF

triple schema in the old query to be taken by the corresponding words in the new query. Thus, “chancellor” adopts the status of “person” (expressed as {Person, Status, Chancellor}), “uniform” adopts “object” and “wear” adopts “predicate”. The new query is predicted as {Person, Wear, Uniform} in an RDF triple.

### 3.4. The memory update module

When the system converts a NL query into its corresponding RDF instance, the user can add it as a new case to the case base or modify it if the result is not correct. Thus, it becomes a new entry in the case base.

## 4. Automatic semantic annotation on descriptive metadata of images

It is generally difficult for intelligent software agents to retrieve images based on the NL descriptive metadata of all images because of the inefficiency in parsing and analyzing the NL sentences during the retrieval process. One compromise is to conduct the semantic annotation on the descriptive metadata in advance. The idea of semantic annotation is to assign domain concepts in terms of semantic tags that are well defined in the domain ontology and thesauri to the word segments or phrases in the descriptive metadata so that it could facilitate the retrieval of the images based on the semantic tags. Yet, it is simply too laborious to manually conduct this kind of semantic annotation. Thus, the first essential step is to automate this process in order to facilitate the generation of annotated descriptive results in accordance with the RDF standards.

The descriptive metadata of each image may consist of several sentences, and each sentence may include several phrases. Since the case-based learning query parser can convert the query phrase into an RDF description, we can easily separate a sentence into several phrases and then pass the phrase to the case-based learning query parser to be converted into RDF descriptions one by one. All RDF descriptions can then be combined together at the end. This automatic semantic annotation system is illustrated in Figure 5. It consists of three major modules - a separation module, a case-based NL parser, and a combination module. We shall elaborate on these modules in the following sections.

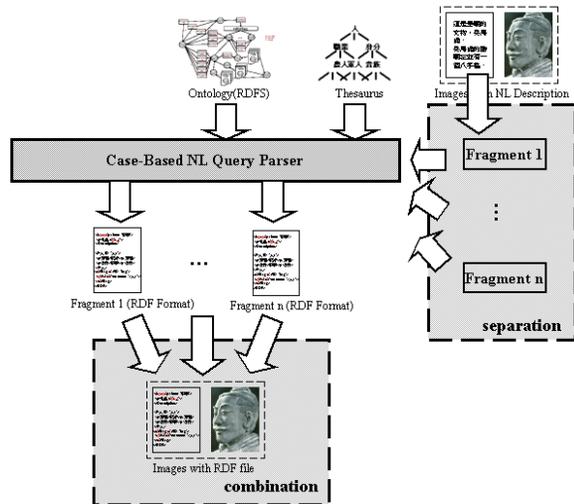


Figure 5. Automatic annotation of metadata descriptions of images

#### 4.1. The separation module

The separation module as shown in Figure 5 separates a sentence description into phrase fragments by punctuations, verbs, or some stop words. For instance, a simple description of a certain image is given as “A tall soldier in a silver armor holds a sharp sword. A leather saddle is on the white horse.” This description consists of two simple sentences and can be separated into five fragments as “A tall soldier”, “in a silver armor”, “holds a sharp sword”, “A white horse” and “a leather saddle.” It is possible that the semantic links between the phrases might be missing after the separation. In this example since the “soldier” from the first phrase fragment is the subject, it covers the following fragment without a subject (“soldier”). So the phrase fragment becomes “A tall soldier”, “soldier holds a sharp sword” and “soldier in a silver armor”. In the second sentence, “horse” is the subject of the first phrase; therefore it becomes the subject of the last phrase fragment as “horse with a leather saddle”.

**Example:** A description of Figure 6 could be as “金黃色的兵馬俑, 身分是將軍, 頭戴著頭盔, 身穿著盔甲, 有濃密的眉毛, 表情威武。” (translated to be “A golden yellow colored terracotta soldier, his status is a general, wearing a helmet on his head, wearing armor, has thick eyebrow, with brave expression” )

The separation module separates the full description into several phrase fragments (PF’s) as follow:

PF 1 “金黃色的兵馬俑” (*A golden-yellow colored terracotta soldier*)

PF 2 “身分是將軍” (*The status is a general*)

PF 3 “頭戴著頭盔” (*wear a helmet on his head*)

PF 4 “身穿著盔甲” (*wear armor*)

PF 5 “有濃密的眉毛” (*has thick eyebrows*)

PF 6 “表情威武” (*with brave expression*)



Figure 6. The general in armor

PFs 1-2 are phrases that have a “subject” (terracotta soldier and general), and PFs 3-6 are those without a “subject”. But in the nearest phrase PF2, a “person” is the “subject” in an ontology schema “a person whose status is a general”, so the separation module assigns “person” as the “subject” to those phrase fragments following the subject. Thus, PFs 3-6 are modified as follows:

PF 3 “頭戴著頭盔的人” (*the person wears a helmet on his head*)

PF 4 “身穿著盔甲的人” (*the person wears armor*)

PF 5 “有濃密的眉毛的人” (*the person has thick eyebrows*)

PF 6 “表情威武的人” (*the person has brave expression*)

After the processing of the separation module, the automatic semantic annotation system requests the CBL NL query parser to convert PFs 1-6 one by one to corresponding RDF descriptions one of which (PF 1) is shown in Figure 7.

```

金黃色的兵馬俑.rdf - 記事本
檔案(F) 編輯(E) 格式(O) 說明(H)
<?xml version='1.0'?>
<RDF xmlns='http://www.w3.org/1999/02/22-rdf-syntax#xsd'
xmlns:P='http://home.kimo.com.tw/cadip_jac'
>
  <Description about='金黃色的兵馬俑'>
    <p:包含物件爲 rdf:resource='#bag'/>
  </Description>

  <P:物品 rdf:ID='物品1'>
    <p:顏色>金黃</p:顏色>
  </P:物品>

  <rdf:Bag rdf:ID='bag'>
    <rdf:li rdf:resource='#物品1'/>
  </rdf:Bag>
</RDF>

```

Figure 7. The RDF instance for the phrase “A golden-yellow colored terracotta soldier”

## 4.2. The combination module

Since all phrase fragments are expressed in RDF format using the same domain ontology and thesaurus, the combination module can easily combine all fragments of RDF descriptions back into complete RDF instances. Using the same example in Section 4.1, the RDF instances of all phrase fragments are constructed. The combination module merges these RDF instances from PFs 1 to 6. The PF 1 is determined as belonging to a description of a piece of “Article/Items” (defined a priori in our domain ontology in terms of RDFS), and PFs 2-6 as belonging to the descriptions of a “Person” by the CBL NL query parser. This means that the descriptions consist of one article and one person. The combination module treats PF 1 as a single Subject/Resource (Article1) description, and merges PFs 2-6 into another single Subject/Resource (Person1) description. The PFs 2-6 are merged as:

```
<p:人 rdf:ID='人 1'>
  <p:身分>將軍</p:身分>
  <p:身體.穿著>盔甲</p:身體.穿著>
  <p:頭.盔飾>頭盔</p:頭.盔飾>
  <p:頭.臉部.眉毛>濃密</p:頭.臉部.眉毛>
  <p:頭.臉部.表情>威武</p:頭.臉部.表情>
</p:人>
```

The prefix “p:” is a name space pointer to the resource file of our domain ontology where the conceptual terms and schemas are defined. The English version of the above RDF description is:

```
<p:person rdf:ID='person1'>
  <p:status>general</p:status>
  <p:body.wear>armor</p:body.wear>
  <p:head.wear>helmet</p:head.wear>
  <p:head.face.expression>brave</p:head.face.expression>
  <p:head.face.eyebrows> thick</p: head.face.eyebrows >
</p:person>
```

After combining the two resource descriptions (Article1 and Person1) into one description, the final RDF description is shown in Figure 8.

## 5. Automatic indexing and structural matching

To retrieve an image from a large image base by description matching from a query can be very time consuming and inefficient if the images are not properly indexed. Since each image has been described in terms of semantic RDF descriptions, we use an automatic indexing

algorithm by grouping images according to their triple descriptions.

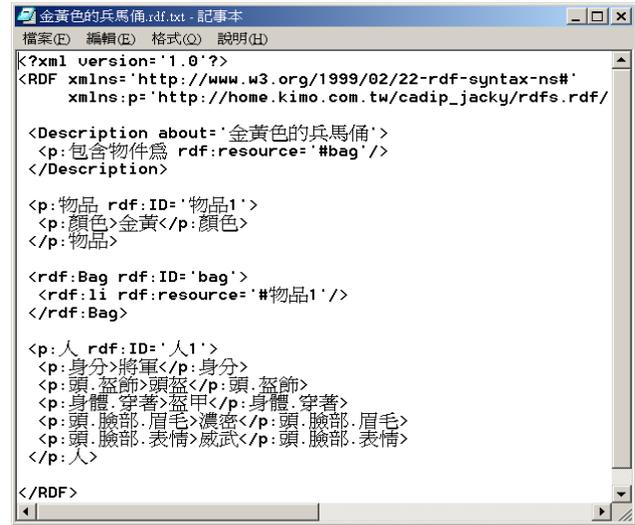


Figure 8. The full description of a RDF instance after combining the phrase fragments

The indexes are formed according to the triples (subject; predicate; object). This indexing technique is somewhat similar to the concept of inverse indexing. However, it is conducted on the conceptual structure level (e.g., a RDF triple structure) rather than on the keyword level. Figure 9 shows a simplified example of two images as the result of the automatic indexing based on the triples.

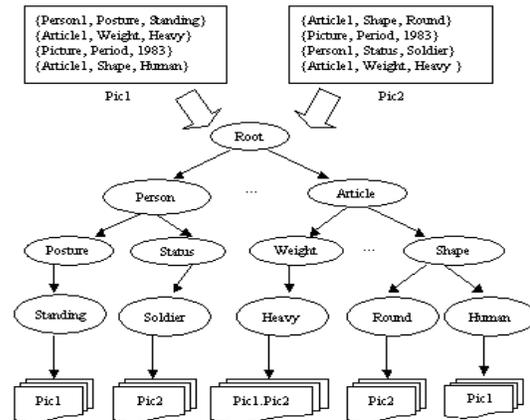


Figure 9. Automatic Indexing based on RDF triples

After the indexing scheme is formed, a simple structural matching algorithm is used to match between the query description and image descriptions in terms of RDF triples. It first splits the RDF query instance into a set of triples, using a scoring function to weigh the matching of each triple against the indexes of the RDF triples of descriptive metadata of images. It computes the

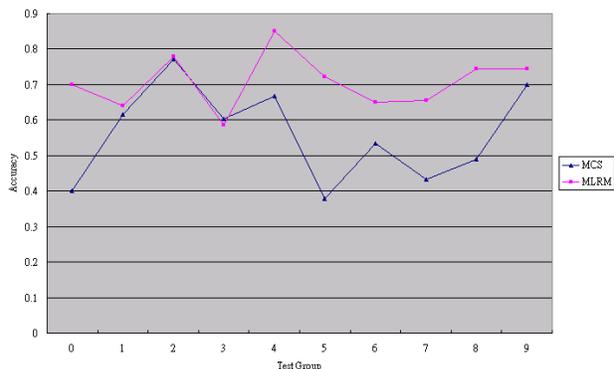
best match image by finding the one with the largest number of triples matched with the query. The major advantage of using this algorithm is the relative ease of matching the image description with the user query and the fast speed in doing it. In other words, the tasks of having to compare the query instance with all image instances one by one is greatly simplified while the computational complexity in time is constant.

## 6. Experimentation and analysis

### 6.1. Results of similarity match in CBL

Our case base consists of 100 NL query phrases with corresponding correct RDF descriptions. Five staff members developed the queries of the selected 50 images and manually converted these queries to RDF descriptions. We then used the statistical cross-validation leave-one-out method to compare the performances between the MCS and MLRM algorithms.

We first randomly divided 100 NL query phrases into 10 groups. We used 9 of the 10 groups of these query phrases as the training set and used the last (one) group as the test set. Using the similarity matching algorithms (MCS and MLRM), the test set is used to compare the accuracy of the case-based learning query parsers. The testing experiments were repeated 10 times using 10 different test sets. These experimental results are shown in Figure 10.



**Figure 10. The comparison of cross validation experiments of MCS and MLRM**

The accuracy was calculated by using the following formula:

$$\text{Accuracy} = \frac{\text{Score of correctness of triples predicted}}{\text{Total score of all triples to be correctly annotated}}$$

The correctness of a triple is scored as 1, 2, or 3 depending on the number of entries in the triple that are

correctly predicted. In other word, if all three entries in a triple are predicted all correctly, it is scored as 3.

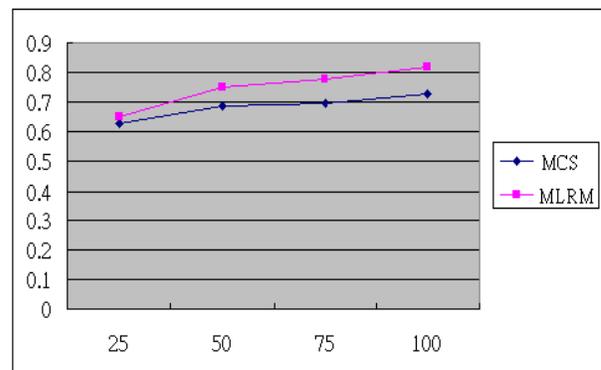
The experimental results show that the average accuracy of MCS is 0.559 with a variance of 0.0179 while the MLRM is 0.707 with a variance of 0.0059. Thus, we conclude that on the average MLRM significantly outperformed MCS.

### 6.2. Results of automatic semantic annotation

We further constructed 20 full descriptions (by 4 different persons) of images in NL in order to evaluate the performance of the automatic annotation system. Each description contains about 3 to 7 phrases, and the number of training cases for the CBL NL parser is 100. We calculated the accuracy of the automated semantic annotation according to the following formula:

$$\text{Annotation accuracy} = \frac{\text{Score of correctness of annotated triples}}{\text{Total score of all triples to be correctly annotated}}$$

The “total score of all triples to be correctly annotated” is computed from the final correctly generated RDF instance, while the “score of correctness of annotated triples” is computed from the triples that have been correctly generated by the automated annotation method. Figure 11 shows the annotation accuracy for each description. The average accuracy of annotation is 0.6 that is indicated as a horizontal line at the 0.6 accuracy level, while the performance of each individual automated semantic annotation varied, from the worst case at 0.3 for the fifth description, to the best case of perfect annotation at the 6th and 20th descriptions.



**Figure 11. The accuracy of automatic semantic annotation**

The automatic annotation module described above has two major potential weaknesses that may decrease the accuracy of annotation: 1) It is based on a CBL NL parser, so the performance of this parser could affect the results of the separation module. 2) Combining RDF instances is

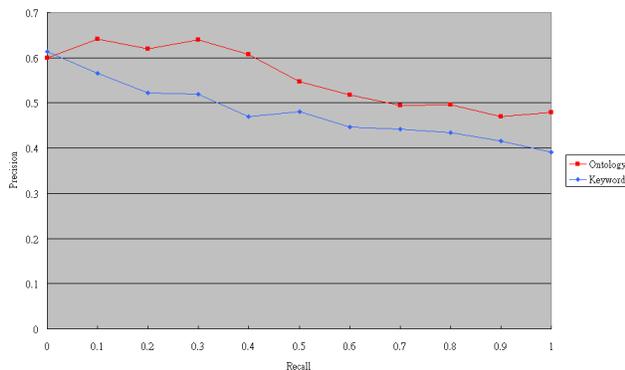
a complex task; therefore, our assumption that each description can only describe one instance for each subject is not adequate. For example, the descriptive metadata of a picture of 5 weapons would be treated by the system as that all “weapons” (subject) are of the same kind, thus all properties of these 5 weapons would be merged together. This is clearly not correct if these weapons are not of the same kind.

Despite the imperfect result of the CBL NL parser in this experiment due to the 1 number of available cases, it is worth noting that most subjects can be assigned even if the separation module did not do so.

### 6.3. Image retrieval results based on ontology

In this experiment, we selected 30 queries at random, and tested them on 49 images that were manually annotated into RDF instances. We then compared the results of image retrieval by using keyword-based and ontology-based retrieval methods. The keyword-based retrieval methods retrieved images with exact keywords specified both in the queries and the retrieved image descriptions. On the other hand, the ontology-based image retrieval converted the queries to RDF instances and retrieved the images based on the structural match described above.

Of the 30 queries, 14 valid ones did not yield any retrieved images in both methods because no images among the 49 tested met the query criteria. The precision versus recall of image retrieval based on the average of the remaining 16 queries is shown in Figure 12.



**Figure 12. The precision-recall curves of the keyword-based retrieval vs. the ontology-based retrieval**

From Figure 12, we observed that the ontology-based retrieval has an advantage over the keyword-based one in terms of precision at the same level of recall. In fact, the 11-point average precisions are 0.55 and 0.48 for the ontology-based retrieval and the keyword-based retrieval, respectively. The average precision gap is 7 %.

To demonstrate further the difference in retrieval accuracy between the keyword-based and the ontology-based methods, and how the ontology-based methods has an advantage over the keyword-based one, two cases are illustrated in the following:

*Case 1:* The first case used the query – “作戰的士兵” (“*Soldier at combat*”) and has yielded significantly different results between the two retrieval methods. This query needs domain ontology information about a “*person*” whose status is “*soldier*” and is in the state of “*combat*”. In the keyword-based retrieval, the relationship between “*combat*” and “*person*” cannot be established together. Yet it can be considered with the aid of the domain ontology. Figure 13 shows the result of the keyword-based retrieval method (5 images in the first row, ranked from left to right according to their match score) vs. the ontology-based retrieval method (5 images in the second row). Clearly, the first image on the left (an inner view of combat carriage) and the second image (another inner view of combat carriage) of the first row should be regarded as mismatches. Thus, results shown on the second row using the ontology-based method had much better retrieval results than those of the keyword-based method.



**Figure 13. The images retrieved using the query, “Soldier at combat”, by using the keyword-based retrieval method (1st row) vs. the ontology-based retrieval method (2nd row), respectively**

*Case 2:* The second case used the query – “左手臂朝左的士兵” (“*The soldier with his left hand directing toward left as viewed by the annotator*”). Again, we have found significantly different retrieval results between the two methods. The query requires to have the ontology describing a person whose status is a soldier and whose left hand directs toward the left. The keyword-based retrieval method cannot establish such a relationship. It can only match the keywords “左手臂” (*left hand*), “朝左” (*directing toward left*), “士兵” (*soldier*) with images, when the metadata descriptions of these images mention those keywords. Figure 14 shows the results of both retrieval methods.



**Figure 14. The images retrieved using the query, “The soldiers whose left hand directs toward left” by using the keyword-based retrieval method (1st row) vs. the ontology-based retrieval method (2nd row) respectively**

The keyword-based retrieval found 5 images shown in the 1st row, with the first image on the left receiving the highest score. The ontology-based retrieval found 5 images shown in the 2nd row. The correct images retrieved are the 1st and 2nd images from the left of the second row (using the ontology-based retrieval) while the keyword-based method resulted in finding the correct images in the 1st and the 4th positions of the first row. This means that the ontology-based retrieval has yielded more accurate search results, because it ranked more accurately on the relation of the “left hand” and “toward left” as well as the status of person as a “soldier”.

## 7. Conclusions and future work

In this work, we have established a framework for facilitating image retrieval using domain ontology. Building on the assumptions that if the descriptive metadata of the image resources were annotated before retrieval and both information resource descriptions and user queries could be converted into semantic web RDF (resource descriptive framework) format with the aid of a sharable domain ontology and thesaurus, then the information resources can be matched or retrieved to some extent with the user query at an abstract and semantic level. This paper contributes the methodologies for automating the query conversion, the semantic annotation, and the retrieval processes using case-based learning techniques. We have developed two similarity comparison algorithms -- MCS and MLRM -- over the NL phrases that can facilitate the retrieval mechanism for finding a most similar case from the case base. The results indicated that the MLRM performed better than the MCS algorithm. We also have extended the techniques of using a case-based learning phrase parser to address the problem of automatic semantic annotation of descriptive metadata of images. For dealing with large-scale

structural matching, we have developed methods for automatic indexing for the RDF instances. The experiments described in this paper have shown the feasibility of automating the semantic annotation of the descriptive metadata of images and automatic conversion of user queries into RDF instances using case-based learning techniques.

While the research results reported here have extended our earlier work as reported in 2001 [11], there are potentials for much more future research including those listed in the following:

1) The influences of ontology expressions on the performance of retrieval cannot be ignored. It affects the semantic interpretations of query instances, cases of CBL, as well as descriptive metadata of images. A finer ontology representation will enhance the performance. However, precise domain ontology construction is a tedious task that requires commitments of both domain experts and ontology engineers. Conducting logical inferences on the domain ontology can augment the performance further. In OWLIR [29], they showed by conducting additional logical inferences on the campus event ontology and OWLIR could increase the precision of information extraction by almost 20% against a RDF triple matching method.

2) Although our automatic semantic annotation can handle simple sentences and phrases well, yet it is not as accurate for complex sentences. Ways to enable annotators to write down descriptive metadata of images in terms of NL sentences rather than RDF structures should be further explored.

3) Merging ontology with more than one RDF instance is a complex but interesting problem. It can be required in many other domains and needs further investigation. How to resolve inconsistencies among many ontological instances is also a real challenge.

## 8. Acknowledgments

This research is supported in part by the MOE Program for Promoting Academic Excellence of Universities under grant number 89-E-FA04-1-4, and the IDLP project of NSC of Taiwan and is in collaboration with the US NSF/IDLP CMNet project led by Prof. Ching-chih Chen of Simmons College of USA under NSF grant number NSF/IIS-9905833.

## 9. References

- [1] Arnold W.M. Smeulders, "Content-Base Image Retrieval at the End of the Early Years", *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 22, No. 12, December, 2000.

- [2] Qasim Iqbal and J.K. Aggarwal, "Image Retrieval via Isotropic and Anisotropic Mapping", in *IAPR Workshop on Pattern Recognition in information system*, Setubal, Portugal, pp. 34-49, July 6-8, 2001.
- [3] "Blobworld Image retrieval using regions", Digital Library Project, University of California, Berkeley, <http://elib.cs.berkeley.edu/photos/blobworld/>.
- [4] Wolfgang Muller and Henning Muller, "MRML: A Communication Protocol for Content-Based Image Retrieval", *Tech. Rep.* 01.01, University of Geneva, 2001.
- [5] Ching-chih Chen, and James Z. Wang, "Large-scale Emperor Digital Library and Semantics-sensitive Region-based Retrieval", *Proceeding of the International Conference on Digital Library -- IT Opportunities and Challenges in the New Millennium*, National Library of China, Beijing, China, pp. 454-462, July 9-11, 2002.
- [6] "The Text REtrieval Conference (TREC), co-sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA)", <http://trec.nist.gov/>.
- [7] Gerard Salton, *The SMART Retrieval System--Experiments in Automatic Document Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1971.
- [8] Gerard Salton and Christopher Buckley, "Term-weighting Approaches in Automatic Text Retrieval. " *Information Processing and Management*, Vol. 24(5), pp.513--523, 1998.
- [9] Jyi-Shane Liu, Von-Wun Soo, Chia-Ning Chiang, Chen-Yu Lee and Chun-Yu Lin, "Gaz-Guide: Agent-Mediated Information Retrieval for Official Gazettes," *In AI Lecture Note Series, Springer*, in *Proc. of PRIMA*, pp. 154-167, 2001.
- [10] Von-Wun Soo, Jyi-Shane Liu and Chia-Ning Chiang, "Exploiting Contextual Knowledge for Information Retrieval in a Multiagent-based Digital Library Service System", *NIT 2001, Beijing. Global Digital Library Development in the New Millennium: Fertile Ground for Distributed Cross-Disciplinary Collaboration*. Ed. by Ching-chih Chen. Beijing: Tsinghua University Press, pp. 299-306, 2001.
- [11] Von-Wun Soo, Chen-Yu Lee, Chao-Chun Yeh and Ching-chih Chen, "Using Sharable Ontology to Retrieve Historical Images", in *Proc. of ACM/IEEE International Joint Conference of Digital Library*, pp.197-198, 2002.
- [12] James Hendler, Tim Berners-Lee, and Eric Miller, "Integrating Applications on the Semantic Web," *Journal of the Institute of Electrical Engineers of Japan*, Vol 122(10), October, 2002, pp. 676-680, <http://www.w3.org/2001/sw/>.
- [13] "Resource Description Framework (RDF) Model and Syntax Specification W3C Recommendation 22, February, 1999", <http://www.w3.org/RDF/>.
- [14] Boris Motik and Vlado Glavinic, "Enabling Agent Architecture through an RDF Query and Inference Engine", *10th Mediterranean Electro-technical Conference, MeleCon*, 2000.
- [15] Stephen CraneField, "Networked Knowledge Representation and Exchange using UML and RDF", *Journal of Digital Information*, Vol.1, Issue 8, 2001.
- [16] Stefan Decker, Frank Van Harmelen and Jeen Broekstra, "The semantic Web - on the respective Role of XML and RDF", <http://www.ontoknowledge.org/>.
- [17] B. Amann and I. Fundulaki, "Integrating Ontologies and Thesauri to Build RDF Schemas", *Third European Conference ECDL'99*, Paris, France, Springer Verlag, pp. 234-253, September, 1999.
- [18] Stefan Decker and Sergey Melnik, "The Semantic Web: The Roles of XML and RDF", *IEEE Internet Computing*, Vol. 4(5), pp. 63-74, October 2000.
- [19] Steffen Staaba and Michael Erdmann, "An Extensible Approach for Modeling Ontologies in RDF(S)", in *Proc. of ECDL Workshop on the Semantic Web*, pp.11-22, 2000.
- [20] Natalya F. Noy, Micheal Sintek and Stefan Decker, "Creating Semantic Web Contents with Protege-2000", *IEEE Intelligent Systems*, Vol.16 (2), pp.60-71, 2001.
- [21] "The DARPA Agent Markup Language Homepage", <http://www.daml.org/>.
- [22] Michael K. Smith, and Deborah McGuinness, "Web Ontology Language (OWL) Guide Version 1.0", <http://www.w3.org/TR/owl-guide/>.
- [23] Peter F. Patel-Schneider, and Patrick Hayes, "Web Ontology Language (OWL) Abstract Syntax and Semantics", <http://www.w3.org/TR/owl-semantics/>.
- [24] Yi-Jia Chen and Von-Wun Soo, "Ontology-based Information Gathering Agents", in *Proc. of Web Intelligence*, pp.423-427, 2001.
- [25] Ching-chih Chen, "The First Emperor of China", CD-ROM, Voyager, 1991.
- [26] David W. Aha. "Case-based Learning Algorithms", in *Proceedings of the DARPA Case--Based Reasoning Workshop*, Morgan Kaufmann, pp. 147-158. 1991.
- [27] C. Cardie, "Using Decision Trees to Improve Case-Based Learning", in *Proc. of the Tenth International Conference on Machine Learning*, Morgan Kaufmann, pp. 25-32, 1993.
- [28] Kirsti Racine and Qiang Yang. "Maintaining Unstructured Case Bases", *In Proc. of the 2nd International Conference on Case-Based Reasoning (ICCBR)*, SpringerVerlag, pp. 553-564, 1997.

[29] Urvi Shah, Tim Finin, Anupam Joshi, R. Scott Cost and James Mayfield, "Information Retrieval on the Semantic Web",

*in Proc. of Conference on Information and Knowledge Management*, Nov 4-9, 2002, McLean, VA., pp. 461-468.