Using Bayesian Decision for Ontology Mapping

Jie Tang*, Juanzi Li, Bangyong Liang, Xiaotong Huang, Yi Li, and Kehong Wang
(Department of Computer Science and Technology, Tsinghua University, P.R.China, 100084)
{j-tang02, liangby97, yi-li}@mails.tsinghua.edu.cn, {ljz, x.huang, wkh}@keg.cs.tsinghua.edu.cn

Abstract. Ontology mapping is the key point to reach interoperability over ontologies. In semantic web environment, ontologies are usually distributed and heterogeneous and thus it is necessary to find the mapping between them before processing across them. Many efforts have been conducted to automate the discovery of ontology mapping. However, some problems are still evident. In this paper, ontology mapping is formalized as a problem of decision making. In this way, discovery of optimal mapping is cast as finding the decision with minimal risk. An approach called RiMOM (Risk Minimization based Ontology Mapping) is proposed, which automates the process of discoveries on 1:1, n:1, 1:null and null:1 mappings. Based on the techniques of normalization and NLP, the problem of instance heterogeneity in ontology mapping is resolved to a certain extent. To deal with the problem of name conflict in mapping process, we use thesaurus and statistical technique. Experimental results indicate that the proposed method can significantly outperform the baseline methods, and also obtains improvement over the existing methods.

Keywords. Ontology Mapping, Semantic Web, Bayesian Decision, Ontology Interoperability

Contact information of Corresponding Author
Name: Tang Jie
E-mail: j-tang02@mails.tsinghua.edu.cn
Mail-Address: 12#109, Tsinghua University, Beijing, P.R. China 100084
Telephone: +86-10-62781461
Fax: +86-10-62789831

[1]Using Bayesian Decision for Ontology Mapping

Jie Tang*, Juanzi Li, Bangyong Liang, Xiaotong Huang, Yi Li, and Kehong Wang
(Department of Computer Science and Technology, Tsinghua University, P.R.China, 100084)
{j-tang02, liangby97, yi-li}@mails.tsinghua.edu.cn, {ljz, x.huang, wkh}@keg.cs.tsinghua.edu.cn

Abstract. Ontology mapping is the key point to reach interoperability over ontologies. In semantic web environment, ontologies are usually distributed and heterogeneous and thus it is necessary to find the mapping between them before processing across them. Many efforts have been conducted to automate the discovery of ontology mapping. However, some problems are still evident. In this paper, ontology mapping is formalized as a problem of decision making. In this way, discovery of optimal mapping is cast as finding the decision with minimal risk. An approach called RiMOM (Risk Minimization based Ontology Mapping) is proposed, which automates the process of discoveries on 1:1, n:1, 1:null and null:1 mappings. Based on the techniques of normalization and NLP, the problem of instance heterogeneity in ontology mapping is resolved to a certain extent. To deal with the problem of name conflict in mapping process, we use thesaurus and statistical technique. Experimental results indicate that the proposed method can significantly outperform the baseline methods, and also obtains improvement over the existing methods.

Keywords. Ontology Mapping, Semantic Web, Bayesian Decision, Ontology Interoperability

1 Introduction

Ontologies, as the means for conceptualizing domain knowledge, have become the backbone to enable the fulfillment of the Semantic Web vision [3]. Many ontologies have been defined to make data sharable, for example, Cyc Ontology [17], Enterprise Ontology [38], Bibliographic-data Ontology [14], Biological and Chemical Ontology (BAO) [25], and Bio-Ontologies [43]. See [45] for more ontologies.

Unfortunately, ontologies themselves are distributed and heterogeneous. Ontologies have two kinds of heterogeneities: *metadata heterogeneity* and *instance heterogeneity* [4, 16]. Specifically, entities (entity represents concept, relation, or instance) with the same meaning in different ontologies may have different label names and the same label name may be used for entities that have different intentional meanings; instances in different ontologies may have different representations; and different ontologies may have different taxonomy structures.

In order to achieve semantic interoperability over ontologies, it is necessary to discover

[1] Supported by the National Natural Science Foundation of China under Grant No. 60443002

* Corresponding author
E-mail addresses: j-tang02@mails.tsinghua.edu.cn, liangby97@mails.tsinghua.edu.cn, ljz@keg.cs.tsinghua.edu.cn, x.huang@keg.cs.tsinghua.edu.cn, yi-li@mails.tsinghua.edu.cn, wkh@keg.cs.tsinghua.edu.cn

ontology mapping at the first step. This is exactly the problem addressed in this paper.

Many efforts have been conducted to deal with the problem. However, the following problems still exist. First, the number of cardinalities that can be processed is limited. Most of the work focuses on only 1:1 mapping [9, 10, 15, 18, 22, 23, 26, 29] despite of the fact that approximately 22%-50% of mappings are beyond this cardinality by statistic on real-world examples [11, 32]. Secondly, Ontology mapping work has been done mainly on metadata heterogeneity, not on instance heterogeneity. In natural language processing, text normalization has been studied [34]. But before adapting the methodology to problem of instance heterogeneity, many efforts are still required. The existing methodologies proposed in the previous work can be used in ontology mapping. However, they are not sufficient for solving all the problems.

At present, questions arise for ontology mapping: (1) how to formalize the problem so that it can describe different kinds of mapping cardinalities and heterogeneities, (2) how to solve the problem in a principled approach, and (3) how to make an implementation.

In this paper, we tried to solve the above problems and have done the following work:

(1) We formalize ontology mapping as that of decision making. Specifically, discovery of optimal mapping is cast as finding the decision with minimal risk.

(2) We propose an approach called RiMOM (Risk Minimization based Ontology Mapping) to conduct ontology mapping by running several passes of processing: first multi-strategy execution in which multiple decisions find the mapping independently; and then strategy combination in which the mappings output by the independent decisions are combined; thirdly mapping discovery in which some mechanisms are used to discover the mapping in terms of the combined results. Mapping process can take place iteratively until no new mappings are discovered. In each iteration, user interaction is supported to refine the obtained mappings.

(3) We make an implementation for the proposed approach. For each available clue in ontologies, we propose an independent decision for finding the mappings. We also make use of the representation normalization and NLP techniques in the mapping process. We combine the results of independent decisions by a composite method.

We tried to collect heterogeneous ontologies from different sources. In total, 28 ontologies from five different sources were gathered. Five data sets were created with the 28 ontologies. Our experimental results indicate that the proposed method performs significantly better than the baseline methods for mapping discovery. We also present comparisons with existing methods. Experimental results indicate improvements over them.

The rest of the paper is organized as follows. Section 2 describes the terminologies used throughout the paper. In section 3, we formalize the problem of ontology mapping and describe our approach to the problem. Section 4 explains one possible implementation. The evaluation and experiments are presented in Section 5. Finally, before conclude the paper with a discussion, we introduce related work.

2 Terminology

This section introduces the basic definitions in the mapping process and familiarizes the reader with the notations and terminologies used throughout the paper.

2.1 Ontology

The underlying data models in our process are ontologies. To facilitate further description, we briefly summarize their major primitives and introduce some shorthand notations. The main components of an ontology are concepts, relations, instances and axioms [7, 37].

A concept represents a set or class of entities or 'things' within a domain. The concepts can be organized into a hierarchy.

Relations describe the interactions between concepts or properties of a concept. Relations fall into two broad types: *Taxonomies* that organize concepts into sub- or super-concept hierarchy, and *Associative relationships* that relate concepts beyond the hierarchy. The relations, like concepts, can also be organized into a hierarchy structure. Relations also have properties that can describe the characteristics of the properties. For example, the cardinality of the relationship, and whether the relationship is transitive.

Instances are the "things" represented by a concept. Strictly speaking, an ontology should not contain any instances, because it is supposed to be a conceptualization of the domain. The combination of an ontology with associated instances is what is known as a knowledge base. However, deciding whether something is a concept or an instance is difficult, and often depends on the application. For example, "Course" is a concept and "Linguistics" is an instance of that concept. It could be argued that "Linguistics" is a concept representing different instances of Linguistics courses such as "French Linguistics Course" and "Spanish Linguistics Course". This is a well known and open question in knowledge management research.

Finally, axioms are used to constrain values for classes or instances. In this sense the properties of relations are kinds of axioms. Axioms also, however, include more general rules, such as a course has at least one teacher.

For facilitating the description, we denote a concept by c and a set of concepts by C ($c \in C$), respectively. We use r to denote relation and use R to denote a set of relations ($r \in R$). We also respectively denote instance and a set of instances by i and I ($i \in I$). Axioms are denoted by A^o.

2.2 Heterogeneity of Ontology

In order to reach interoperability over heterogeneous ontologies, two problems must be dealt with: *metadata heterogeneity* and *instance heterogeneity* [4, 16]. Metadata heterogeneity concerns the intended meaning of described information. There are two kinds of conflicts in metadata heterogeneity: structure conflict and name conflict. Structure conflict means that ontologies defined for the same domain may have different taxonomies. Name conflict means that concepts with the same intended meaning may use different names and the same name may be used to define different concepts.

Figure 1 shows an example of metadata heterogeneity. Two ontologies O_1 and O_2 respectively represent college courses at Washington University and Cornell University[2]. The dashed line in the figure represents a reasonable mapping between them. Table 1 lists the mappings.

In the example, the concept "Asian_Studies" in Ontology O_1 has the same meaning as concept "Asian_Lanugages_and_Literature" in ontology O_2. But they have different names. On the other hand, the concept "Linguistics" is defined in both O_1 and O_2. However they represent slightly

[2] http://anhai.cs.uiuc.edu/archive/summary.type.html

different meanings. In ontology O_1, “Linguistics” denotes a linguistics course which focuses on the basic analytic methods of several subfields of linguistics such as phonetics, phonology, morphology, syntax, semantics, and psycholinguistics. While in ontology O_2, “Linguistics” is referred to as a taxonomy of linguistic courses, including four sub-classes: French linguistic course, Romance linguistic course, Spanish linguistic course, and Linguistics course.

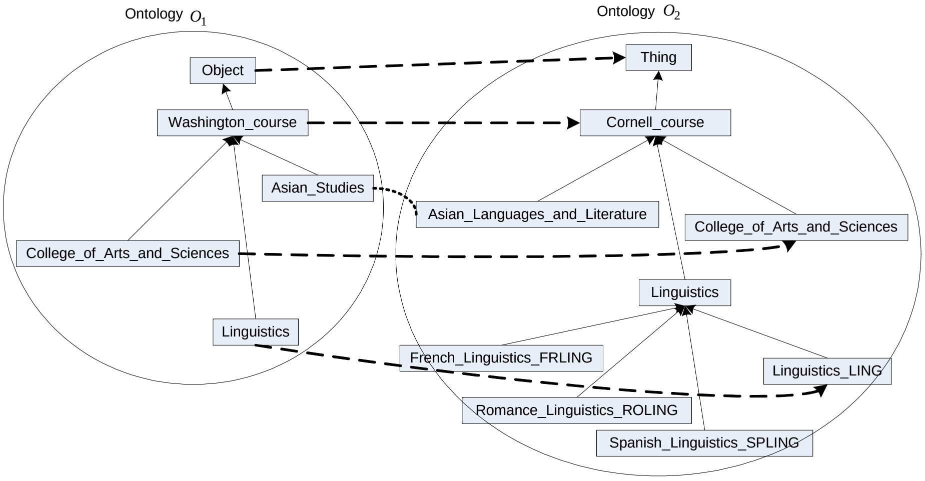

Ontology O_1
Ontology O_2
Object
Washington_course
Asian_Studies
College_of_Arts_and_Sciences
Linguistics
Thing
Cornell_course
Asian_Languages_and_Literature
College_of_Arts_and_Sciences
Linguistics
French_Linguistics_FRLING
Linguistics_LING
Romance_Linguistics_ROLING
Spanish_Linguistics_SPLING

Figure 1. Example of two heterogeneous ontologies and their mappings

Table 1. Mappings from O_1 to O_2

Ontology O_1	**Ontology O_2**
Object	Thing
Washington_course	Cornell_course
Asian_Studies	Asian_Languages_and_Literature
College_of_Arts_and_Sciences	College_of_Arts_and_Sciences
Linguistics	Linguistics_LING

Instance heterogeneity concerns the different representations of instances. Information described by the same ontology can be represented in different ways. This is also called representation conflict. For example, a date can be represented as “2004/2/27” and also can be represented as “Feb, 27, 2004”; person name can be represented as “Jackson Michael” or “Michael, Jackson”, etc. Instance heterogeneity makes it necessary for normalization before ontology interoperation.

Many efforts have been placed on the problem of metadata heterogeneity and few works focus on instance heterogeneity, to the best of our knowledge. Moreover, most of the existing work was focusing on 1:1 mapping.

2.3 Ontology Mapping

Ontology mapping takes two ontologies as input and creates a semantic correspondence between the entities[3] in the two ontologies [32].

In this paper, we define ontology mapping as a directional one. Given a mapping from ontology O_1 to O_2, we call ontology O_1 as source ontology and O_2 as target ontology. We call the process of finding the mapping from O_1 to O_2 as (Ontology) mapping discovery or mapping prediction.

[3] In this paper, to facilitate the description, we use entities to denote concepts, properties and relations.

Formally, ontology mapping function *Map* can be written in the following way:

$$Map(\{e_{i1}\}, O_1, O_2) = \{e_{i2}\}$$

with $e_{i1} \in O_1$, $e_{i2} \in O_2$: $\{e_{i1}\} \xrightarrow{Map} \{e_{i2}\}$. $\{e_{i1}\}$ or $\{e_{i2}\}$ denotes a collection of entities, and $e_{i1} \in C_{O1} \cup R_{O1}$. The target entity collection can contain one entity, multiple entities or null. Here null means that there is no mapping for $\{e_{i1}\}$ in O_1.

To facilitate the description, we usually leave out O_1 and O_2 and write the function as $Map(\{e_{i1}\})=\{e_{i2}\}$. Moreover, we use the notation $Map(O_1,O_2)$ to denote all entity mappings from O_1 to O_2.

There are six kinds of mapping cardinalities: 1:1, 1:n, n:1, 1:null, null:1, and n:m. Table 2 shows examples of the cardinalities.

Table. 2. Mapping cardinality examples

Cardinality	***O*₁**	***O*₂**	**Mapping Expression**
1:1	Faculty	Academic staff	O_1.Faculty= O_2.Academic staff
1:n	Name	First name, Last name	O_1.Name= O_2.First name+O$_2$.Last name
n:1	Cost, Tax ratio	Price	O_1.Cost*(1+ $O_{1.}$Tax ratio)= O_2.Price
1:null	AI		
null:1		AI	
n:m	BookTitle, BookaNo, PublisherNo, PublisherName	Book, Publisher	O_1.BookTitle + O_1.BookaNo + O_1.PublisherNo + O_1.PublisherName = O_2.Book + O_2.Publisher

Among these kinds of cardinalities, existing mapping methods was mainly focusing on only 1:1 mapping. This paper has investigated the problem of mappings with 1:1, n:1, 1:null, and null:1. The kind of n:m mapping is more complicated and is not the focus of this paper. For 1:n mapping, we consider it in a bidirectional mapping discovery process, that is, we find 1:n mapping by making use of both the mapping from O_1 to O_2 and the mapping from O_2 to O_1. In this paper, we confine ourselves to the one directional mapping and focus on the 1:1, n:1, 1:null and null:1 mappings.

Once a mapping $Map(\{e_{i1}\},\{e_{i2}\})$ between two ontologies O_1 to O_2 is discovered, we say that "entities $\{e_{i1}\}$ is mapped onto entities $\{e_{i2}\}$". For each pair of entity sets ($\{e_{i1}\}$, $\{e_{i2}\}$), we call it a *candidate mapping*. We make the assumption that an entity in the source ontology can only participant into at most one mapping.

3 Ontology Mapping Modeling

In this section, we first briefly introduce the Bayesian decision theory and its use in RiMOM, and then describe the mapping process, finally illustrate the sub decisions that are exploited to determine the mappings.

3.1 Bayesian Decision Theory

Bayesian decision theory provides a solid theoretical foundation for thinking about problems of action and inference under uncertainty [2]. In Bayesian decision theory, the observations are a set of samples *X*, in which each sample is denoted as *x*. Let $y \in Y$ be a 'class'. Each sample *x* can be classified into one class. Let $p(y|x)$ denote the conditional probability of the sample *x* belonging to class *y*. Let $A = \{a_1, a_2, \cdots, a_n\}$ be a set of possible decisions (actions). Actions are defined according to the specific application. For each action a_i, Bayesian decision theory associate a loss function $L(a_i, y)$ to indicate the loss of classifying the sample *x* to class *y*.

Given *Y* and *A*, the Bayesian risk of each sample *x* is defined by:

$$R(a_i \mid x) = \int_y L(a_i, y) p(y \mid x) dy$$

The solution to the Bayesian problem is to find an action a_i which minimizes the risk.

$$a^* = \arg_a \min R(a \mid x)$$

Classification is a special case of Bayesian decision problem where the set of action *A* and classes *Y* coincide with each other. An action then means to classify sample *x* to class *y*. For example, in Naïve Bayes classification, to find the action a_i with minimal risk means to classify the sample *x* to class *y* with highest probability (inversely minimal loss).

3.2 RiMOM (Risk Minimization based Ontology Mapping)

In terms of Bayesian decision theory, we formalize the ontology mapping problem as that of decision making. This section presents an ontology mapping model, called Risk Minimization based Ontology Mapping (RiMOM).

In our case, our observations are all entities in the two ontologies O_1 and O_2. Entities $\{e_{i1}\}$ in O_1 are viewed as samples and entities $\{e_{i2}\}$ in O_2 are viewed as classes. Each entity e_{i1} can be classified to one ‘class’ e_{i2}. This also means that entity e_{i1} is mapped onto entity e_{i2}. We use $p(e_{i2}| e_{i1})$ to denote the conditional probability of the entity e_{i1} being mapped onto entity e_{i2}. We then define actions as all possible mappings (i.e. all candidate mappings). In this way, finding the optimal mapping is formalized as finding the action with minimal risk.

We denote the loss function as $L(a_i, e_y, O_1, O_2, e_x)$. For entity e_x in O_1, the Bayesian risk is given by

$$R(a_i \mid e_x, O_1, O_2) = \int_{e_y} L(a, e_y, O_1, O_2, e_x) p(e_y \mid e_x, O_1, O_2) d(e_y), \qquad e_x \in O_1$$

We include O_1 and O_2 in the conditional probability $p(e_{i2}| e_{i1},O_1,O_2)$, which means that not only the information of e_x and e_y themselves but also the global information in O_1 and O_2 will be considered for calculating the mapping risk.

We employ a commonly used loss function, Log loss function, which is defined as:

$$L(a_i, e_y, O_1, O_2, e_x) = \log(p(e_y \mid e_x, O_1, O_2))$$

Finally, based on the Bayesian decision theory, the sufficient and necessary condition for minimal Bayesian risk is to find minimal risk for each sample. Thus the risk of mapping from O_1 to O_2 is defined as:

$$R = \int_{e_x} R(a_i \mid e_x, O_1, O_2) d(e_x), \qquad e_x \in O_1 \qquad \dots (1)$$

3.3 Process

Equation (1) is a general formula to view ontology mapping as a decision problem. There are many methods to implement it. In mapping discovery, different information can be exploited, e.g.

instance, entity name, entity description, taxonomy structure, and constraint. We designed a sub-decision for each of the available clues. Every sub decision can be used independently to discover the mappings from O_1 to O_2. The discovered mappings by these sub-decisions are then combined into the final mappings. In this paper, we also call the implementation of each decision as strategy.

Figure 2 illustrates the mapping process in RiMOM with two input ontologies, one of which is going to be mapped onto the other. It consists of five phases:

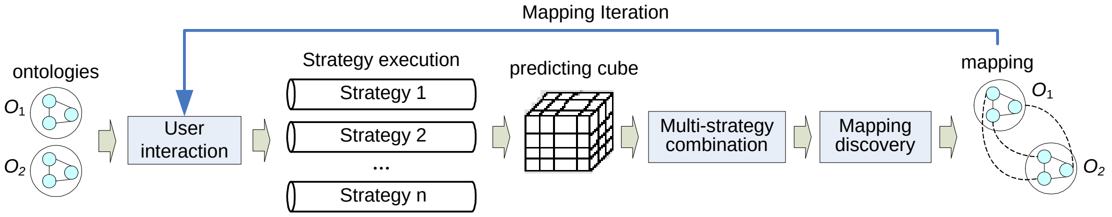


Figure 2. Mapping process in RiMOM

1. User Interaction (optional). RiMOM supports an optional user interaction to capture information provided by the user. The information can be rectified mapping or newly created mapping. The targeted user interaction can be used to improve the mapping accuracy.
2. Multi-strategy execution. The crucial process in mapping iteration is the execution of multiple independent mapping strategies. Every strategy determines a predicting value between 0 and 1 for each possible candidate mapping. The output of the mapping execution phase with k strategies, m entities in O_1 and n entities in O_2 is a $k*m*n$ cube of predicting values, which is stored for later strategy combination.
3. Strategy combination. In general, there may be several predicting values for a pair of entities, e.g. one is the prediction by their name and another one is by their instances. This phase is to derive the combined mapping results from the individual decision results stored in the predicting cube. For each candidate mapping, the strategy-specific predicting values are aggregated into a combined predicting value.
4. Mapping discovery. This phase uses the individual or combined predicting values to derive mappings between entities from O_1 to O_2. In existing literature, mechanisms include using thresholds or maximum values for mappings prediction [26], performing relaxation labeling [11], or combining structural criteria with similarity criteria.
5. Iteration. Mapping process taking place in one or more iterations depends on whether an automatic or interactive determination of mapping is to be performed. In interactive mode, the user can interact with RiMOM in each iteration to specify the mapping strategies (selection of mapping strategies), to correct mistake mappings, to create new mappings, or to accept/reject mappings from the previous iteration. In automatic mode, the strategies perform iteration over the whole process. Outputs of the iteration can be used in the next iteration. Each iteration contains two parts: one is to discover concept mappings and the other is to discover relation mappings. Iteration stops until no new mappings are discovered.

Eventually, the output is a mapping table. The table includes multiple entries, each of which corresponds to a mapping. An entry in the mapping table contains two entity sets. One set is the source entity set in O_1 and the other is the target entity set in O_2. Table 1 shows an example of the mapping table.

3.4 Multiple Decisions in RiMOM

In this section, we first present the sub decisions for each available clue. Then we combine the results from these sub decisions.

1. Name based decision

The most intuitive method may be that of exploiting entity name to discover the mapping. Several approaches have been proposed to conduct the mapping discovery by making use of the entity name. For example, Madhavan et al use VSM (Vector Similarity Model) by casting the problem as that of information retrieval [19]; Bouquest et al propose to employ Edit Distance to compute the similarity of entity names [4]; and Doan et al utilize machine learning methods to make prediction [11]. However, all the approaches have some troubles. Specifically, information retrieval methods usually result in unsatisfactory results. Edit Distance define the strings similarity by the minimum number of insertions, deletions, and substitutions required to transform one string into the other. It ignores that two entity name with similar meaning might be absolutely differently spelled. Moreover, classifier usually is effective on long text content, but not effective on short text content. Entity name is often represented by short text.

We propose to conduct name based decision by combining thesaurus method with statistical technique. Formally, we can define the similarity between words w_1 and w_2 as:

$$sim(w_1, w_2) = (sim_d(w_1, w_2) + sim_s(w_1, w_2))/2$$

where, $sim_d(w_1,w_2)$ denotes the similarity between w_1 and w_2 according to thesaurus. As the thesaurus, we use Wordnet, one of the most popular thesauruses. $sim_s(w_1,w_2)$ is the statistical similarity which will be described later.

Wordnet has a semantic network of word senses, in which each node is a synset. A synset contains words with same sense and a word can occur in different synsets indicating that the word has multiple senses. Lin defines the similarity between two senses in Wordnet [30] as:

$$sim_d(s_1, s_2) = \frac{2 \times \log p(s)}{\log p(s_1) + \log p(s_2)}$$

where, $p(s) = count(s)/total$, is the probability of a randomly selected word occurring in synset s or any synsets below it. $total$ is the number of word in Wordnet and $count(s)$ is the word count in s and synsets below it. The synset s is the common hypernym of synsets s_1 and s_2 in WordNet.

Let $s(w_1) = \{s_{1i} \mid i=1,2,\ldots,m\}$ and $s(w_2) = \{s_{2i} \mid i=1,2,\ldots,n\}$ denotes the senses of w_1 and w_2 respectively. We define the similarity of two words by the maximum similarity between their senses. It is written as:

$$sim_d(w_1, w_2) = \max(sim_d(s_{1i}, s_{2j})) \qquad s_{1i} \in s(w_1), s_{2j} \in s(w_2)$$

For statistical similarity calculation, we use a statistical similarity dictionary. Lin constructs a thesaurus, in which similarities between words are calculated based on their distribution in the documents [30]. We obtain the value of $sim_s(w_1,w_2)$ by directly looking up the dictionary.

It is necessary to do pre-processing before calculating the name similarity. The pre-processing includes: text tokenization for deriving a bag of tokens, e.g. “Earth-and-Atmospheric-Sciences”→{earth, and, atmospheric, sciences}, and expansion of abbreviations and acronyms, e.g. “CS”→{Computer, Science}.

The name based strategy then computes similarity matrix for the two words sets. Each value in the matrix denotes the similarity of a pairwise of words. Specifically, for two entity names, $name_1$

and $name_2$, they are pre-processed into two token sets $\{w1_i\}$ and $\{w2_j\}$. Then for each $w1_i$, we select the highest similarity $sim(w1_i,w2_j)$ as the similarity between $w1_i$ and $name_2$, i.e. $sim(w1_i, name_2)$. Finally, the similarity of $name_1$ and $name_2$ is defined as

$$sim(name_1, name_2) = \sum_{i=1\ldots n} sim(w1_i, name_2)/n$$

where n is the word count in $name_1$.

By comparison of existing methods, this method works well not only on similar names, but also on different names with semantic relationship.

2. Instance based decision

This strategy makes use of text classification techniques to find entity mappings. The inputs are all entities and their instances in the two ontologies.

An entity can have many instances. An instance typically has a name and a set of properties together with their values. We treat all of them as the textual content of the instance. We also take the documents that related to the instance as a kind of source to its textual content. For example, in the Course ontology of AnHai's data^2, we can take the web pages that related to the instance as its text content. In this way, we create a 'document' for each instance and a 'document' set for each enetity.

This strategy exploits the word frequencies in the textual content of the instance to discover mappings. It formulates ontology mapping as a classification problem. Given two ontologies O_1 and O_2 with a set of entities $\{e_{i1}\}$ and $\{e_{i2}\}$ respectively, and each entity e_{i1} with a set of instances $I_{i1}=\{i_{i1k}\}$, the decision takes $\{e_{i2}\}$ as classes, instances in O_2 as training samples and instances in O_1 as test samples, so that the mapping can be automatically discovered by predicting the class of the test samples. The textual content of each instance is processed into a bag of words, which are generated by word tokenizing, stop-word removing and word stemming. Let $i_{i1k} =\{w\}$ be the content of an input instance where w is a word.

We employ Naïve Bayesian (NB) classifier. NB tries to generate a model from training samples. The model can be applied to classify test samples. Given test instances I_{i1}, NB predicts its class by $\arg\max_{e_{i2}} p(e_{i2} \mid I_{i1})$. The posterior probability $p(e_{i2}|I_{i1})$ is calculated by:

$$p(e_{i2} \mid I_{i1}) = p(I_{i1} \mid e_{i2})p(e_{i2})/p(I_{i1})$$

In the equation, $p(I_{i1})$ can be ignored because it is just a constant. $p(e_{i2})$ is estimated as the probability of training instances that belong to e_{i2}. To compute $p(I_{i1}|e_{i2})$, we make the assumption that words appear in instances I_{i1} independently of each other for the given e_{i2}. Thus $p(I_{i1}|e_{i2})$ can be computed by $p(I_{i1} \mid e_{i2}) = \prod_{w \in I_{i1}} p(w \mid e_{i1})$. Finally, we are able to rewrite $p(e_{i2}|I_{i1})$ as:

$$p(e_{i2} \mid I_{i1}) = \prod_{w \in I_{i1}} p(w \mid e_{i2}) \bullet p(e_{i2}) \quad \ldots (2)$$

where $p(w|e_{i2})$ is estimated by $n(w,e_{i2})/n(e_{i2})$. $n(e_{i2})$ is the total number of words in the instances of e_{i2}, and $n(w,e_{i2})$ is the number of times that word w appears in the instances of e_{i2}.

For each possible candidate mapping of e_{i1}, the strategy computes the probability $p(e_{i2}|I_{i1})$, and predicts the mapping by $\arg\max_{e_{i2}} p(e_{i2} \mid I_{i1})$.

Instance based decision works well on long contents. It seems less effective on short contents.

3. Description based decision

Entity usually has comment or description (for short, we use description hereafter) and description is often expressed by natural language and is also one kind of valuable information for ontology mapping. Typically, it reflects more semantic of the entity than entity name itself.

We use text classification method to find mapping by using the entity description. Specifically, we use word frequencies in entity descriptions of the target ontology and construct a Bayesian classifier. Then we exploit words in entity descriptions of the source ontology for prediction. The principle of this decision is similar to that of instance based decision except that in instance based decision the words are from instance textual content while in description based decision the words are from the entity description.

4. Taxonomy context based decision

Taxonomy structure describes the taxonomy context for the entity. The strategy is derived from the intuition that entities occurring in the similar contexts tend to be matchable, e.g. “two concepts match if their sub-classes match”. A concept’s taxonomy context includes its super class, sub-classes, properties and relations. A relation’s taxonomy context includes its subject, object, super relation, sub relations, and constraints. Thus, the taxonomy context similarity of two entities can be defined by aggregating similarities of the respective entities in their contexts. The similarities are obtained from the other strategies such as name based decision and instance based decision. In the current version, we only consider the entities in the immediate context[4].

5. Constraints based decision

Constraints are often used to restrict concepts and properties in ontology. They are also useful for mapping discovery.

We utilized the constraints by defining heuristic rules for refining the learned mappings. Examples of such rules are:

- *datatypeproperty* with *range* “Date” can only be mapped to the *datatypeproperty* with *range* “Date”->confidence: 1.0.

- *datatypeproperty* with *range* “float” may be mapped to one with *range* “string”->confidence: 0.6. The rules are also defined similarly for “nonNegativeInteger”, “boolean”, etc.

- concepts that have the same properties but the properties have different cardinalities may not be mapped to each other-> confidence: 0.3. Here, for the same properties, we mean two properties that are proposed as a mapping by the other decisions. The rules are also defined similarly for “maxCardinality” and “minCardinality”.

- concepts that have the same number of properties tends to be mapped to each other->confidence: 0.3.

Each constraint is assigned with a confidence (e.g. 1.0 and 0.6) to extend the traditional Boolean constrain (i.e. yes or no). The confidences are specified manually. By far, we totally define 12 rules according to the constraints in ontology language and domain knowledge.

6. Using NLP to improve the decision

Information processing on plain text usually meets the problem of data sparseness. Data sparseness makes the classifier over-fitting the training examples, thus affects its effectiveness on unseen cases. In the processing of mapping discovery, we also observed such problem: lack of common instances. For example, instances of concept “telephone number” in two ontologies can have few common ones. The problem depresses the performance of instance based decision and description based decision. We propose to deal with the problem by making use of NLP technique.

Existing NLP techniques can be used to associate additional linguistic knowledge to each word.

[4] However, indirectly related entities will be considered in the future work.

The NLP techniques include: morphological analyzer, POS tagging, name entity recognizer, user-defined dictionary, etc. We employ POS (Part of Speech) and Name entity recognition results as the additional knowledge. An example is shown in table 3 (We conduct the NLP analysis by using GATE [3]).

Table 3. Instances of concept *Address* with NLP knowledge

Instance with NLP Knowledge			
Index	**Word**	**POS(Part of Speech)**	**Name Entity**
1	Knowledge	Noun	Organization
2	Engineering	Noun	
3	Group	Noun	
4	Tsinghua	Noun	University
5	University	Noun	
6	China	Noun	Country
7	100084	Number	Zipcode

With the additional knowledge, Bayesian classifier can learn the model not only by the bag of words but also by their POSs and name entities. Then, equation (2) becomes:

$$p(e_{i2} \mid I_{i1}) \propto \frac{(a_1 \prod_{w\in I_{i1}} p(w \mid e_{i2}) + a_2 \prod_{POS\in I_{i1}} p(POS \mid e_{i2}) + a_3 \prod_{ne\in I_{i1}} p(ne \mid e_{i2})) \bullet p(e_{i2})}{a_1 + a_2 + a_3} \quad \dots (3)$$

where $p(POS|e_{i2})$ is the conditional probability of *POS* given entity e_{i2}; $p(ne|e_{i2})$ is the conditional probability of name entity *ne* given entity e_{i2}. Parameters a_1, a_2, and a_3 are weights to tune the preferences to word, POS and name entity, respectively.

7. Combination of multi-decision

Outputs of the strategies need to be combined. There are two most popular approaches for combination: the *hybrid* or *composite* approach [9, 11]. Hybrid method is usually used when multiple algorithms are integrated into a single algorithm. Composite method is used when multiple algorithms results need to combination. We employed the composite method and combine the strategies by:

$$Map(e_{i1}, e_{i2}) = \sum_{k=1\ldots n} w_k \sigma(Map_k(e_{i1}, e_{i2})) \Big/ \sum_{k=1\ldots n} w_k$$

where w_k is the weight for individual strategy, and σ is a sigmoid function. Sigmoid function makes the combination emphasize high individual predicting values and de-emphasize low individual predicting values. Function σ is defined as:

$$\sigma(x) = 1/(1 + e^{-5(x-\alpha)})$$

where x is a individual predicting value. We tentatively set α as 0.5. The general shape of the sigmoid function is shown in figure 3.

4 Implementation

In this section, we consider one implementation of RiMOM. We focus on two phases in ontology mapping: Preprocessing and Discovery. We will not focus on mapping representation. In this paper, it is expressed by XML (section 4.2 will give an example). See [20] and [33] for details about mapping representation.

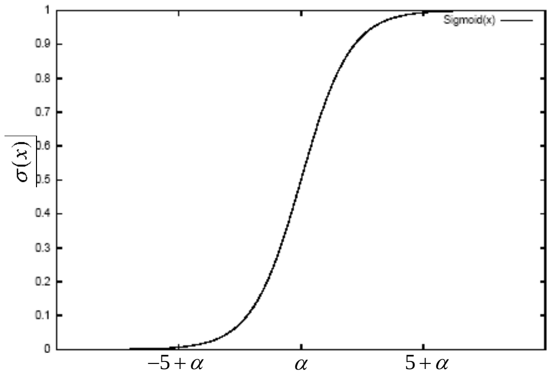


Figure 3. The Sigmoid function

4.1 Preprocessing

Before mapping process, textual contents of instances, entity names and entity descriptions need to be preprocessed. The preprocessing includes tokenization, stop word removing, word stemming, POS tagging, name entity recognition, and normalization. In our implementation, we use a general toolkit (viz. GATE [6]) to perform the preprocessing. GATE integrates many tools for NLP, including morphological analyzer, POS tagger, user-defined dictionary, and name entity recognition (recognition of person name, dates, number, organization names, etc). We process the textual content of each instance and store the result for later processing.

The same instance may have various expressions (also called instance expression conflict). In natural language processing, Sproat et al have investigated normalization of non-standard words in text processing [34]. They define a taxonomy of non-standard words and apply n-gram language models, decision trees, and weighted finite-state transducers to the normalization. But in ontology, the instance expression may not be in natural language, and thus the n-gram based method may not work well.

We formalize the problem as that of instance normalization. We conduct the normalization as follows. We first use GATE to identify the name entities as candidates for normalization (including: Time, Date, Year, Percentage, Money, Person Name, etc). We then defined hard-rules for normalizing Time, Date, Year, Percentage, Money, and Person Name. For example, for Date we transform the format to a unique form: year-month-day, e.g. “2004-3-1” and “March 1, 2004” are both transformed into the format “2004 March 1”; for Person Name, we normalize its format into “firstname lastname”, e.g. “Jackson Michael” and “Michael, Jackson” are both normalized into “Jackson Michael”[5]. We also merge the person names like “J. Michael” and “Jackson Michael”. Rules for other type of name entities are also defined in this way. We omit the details due to space limitation.

It seems reasonable to conduct the normalization for instances in this way. The rules defined work well in most of the cases. By a preliminary analysis on the 28 ontologies, we have found that more than 85.5% of the instance expressions conflict comes from Time, Date, Year, Percentage, Money, and Person Name.

Other task in this phase is to normalize the entity name for facilitating the name based decision. For example, given a concept’s name “company_information”, we need to tokenize it into

[5] GATE can recognize the first name and last name by name entity recognizer and by a user defined dictionary.

{company, information}. For relation name “hasEmployee”, we tokenize it into {has, Employee}. Moreover, we expand the abbreviations and acronyms by using a predefined dictionary such as “CS”→{Computer, Science}.

4.2 Discovery

Discovery consists of four stages: entity mapping, mapping combination, mapping discovery, and mapping refinement. First, concept mapping and relation mapping are performed independently by multiple decisions as described in section 3. Secondly, we combine the results from the multiple decisions and obtain a composite result. After that, we employ several strategies to determine the 1:1, n:1, 1:null, and null:1 mappings. Finally, we refine the generated mappings.

Multiple decisions and the combination algorithm are presented in section 3. In this section, we mainly discuss the discovery process and the refinement method.

1. Mapping discovery process

```
Input: onto1, onto2
Output: Mapping table

  //using techniques of normalization and NLP to preprocess
  preprocess(onto1);
  preprocess(onto2);
  //search for concept mapping
  foreach(concept_i in onto1)
      foreach(concept_j in onto2)
          //compute all sub-decisions
          NamePrediction(concept_i,concept_j);
          InstancebasedPrediction(concept_i,concept_j);
          DescriptionPrediction(concept_i,concept_j);
          TaxonomyContextPrediction(concept_i,concept_j);
          ConstraintDecision(concept_i,concept_j);
          DecisionCombination(concept_i);
      PreConceptDecision();
  ConceptMappingDecision();

  //prune concept mapping
  PruneConceptMapping();

  //search for property mapping
  foreach(property_i in onto1)
      foreach(property_j in onto2)
          //compute all sub-decisions
          NamePrediction(property_i,property_j);
          InstancebasedPrediction(property_i,property_j);
          DescriptionPrediction(property_i,property_j);
          TaxonomyContextPrediction(property_i,property_j);
          ConstraintDecision(property_i,property_j);
          DecisionCombination(property_i);
      PrePropertyDecision();
  PropertyMappingDecision();

  //prune concept mapping
  PrunePropertyMapping();

 //refine the generated mapping.
 MappingRefinement();
```

Figure 4. The flow in mapping discovery

In mapping discovery process, RiMOM computes the Bayesian risk for each possible mapping, and then searches the whole space to find the mapping with minimal risk. The algorithm of mapping discovery is shown in figure 4. *preprocess*() is the preprocessing procedure as described in section 4.1. *NamePrediction*(), *InstancebasedPrediction*(), *DescriptionPrediction*(), *TaxonomyContextPrediction*() and *ConstraintDecision*() are five sub-decisions. *DecisionCombination*() is the function to combine the results of the multiple decisions. For each concept, *PreConceptDecision*() first outputs top ranked three mappings. And then for all concepts, *ConceptMappingDecision*() determines the final concept mappings by using the outputs of *PreConceptDecision*(). *PruneConceptMapping*() uses domain knowledge to prune the error mappings and to discover 1:null mappings. We employ the same procedure to find mappings of properties. After that, we conducted a mapping refinement procedure. In this procedure, we refine the concept mapping and property mapping by making use of their results for each other. We should also take into consideration of other kinds of mappings. For example, since it is not necessary disjoint for concepts, properties, and instances, there should also include mappings of concept to instance, instance to concept, property to concept, etc. In this paper, we confine ourselves to the mapping of concept to concept and property to property. Because we have observed few other mapping types available in our data.

1:1 mapping is the simplest and also most common cardinality. The task of finding 1:1 mapping is accomplished by selecting the corresponding entity with minimal risk from O_2 for each entity in O_1. The selection is determined by the combination of decisions described in section 3.

n:1 mapping

n:1 may exist when multiple entities in O_1 are mapped to one entity in O_2. The discovery of n:1 mapping consists of two steps: mapping entities discovery and mapping expression discovery. In mapping entities discovery, we are aimed at finding whether there are multiple source entities mapped onto one target entity. In mapping expression discovery, we try to search for a function for combining the source entities so that the source entities can be best matched by the target entity. For example, the source entities are *firstname* and *lastname* and the target entity name is *person name*, then the expression function can simply be concatenation of the two source entities: *concat*(*firstname*, *lastname*) (also written as *firstname* + *lastname*).

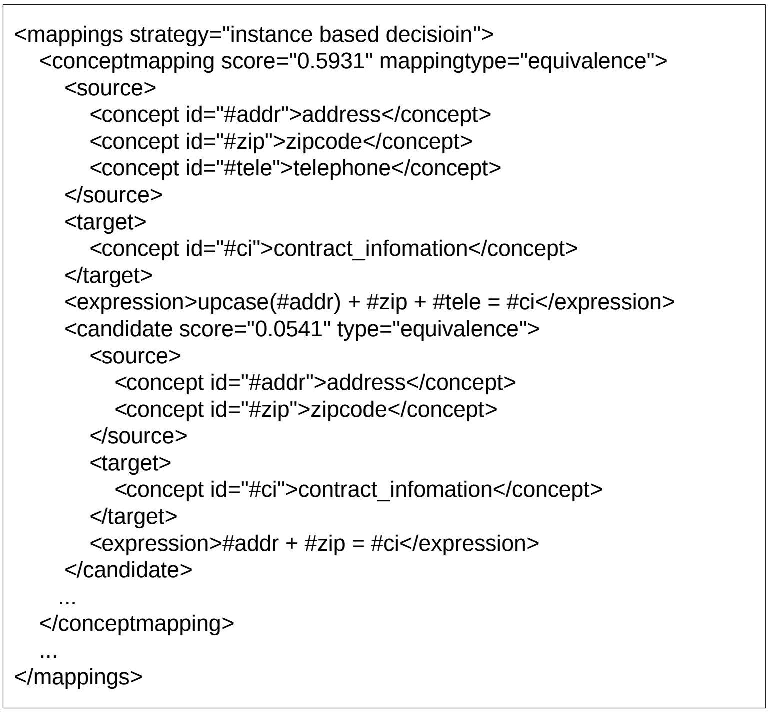
After predicting mapping for each entity of the source ontology, RiMOM search all the mappings to see whether there exist multiple source entities mapped onto the same target entity. If exist, RiMOM triggers a combination process, which automatically searches for the expression function. Now, we use an example to illustrate the process.

For example, when three concepts *Address*, *Zipcode* and *telephone* are all mapped onto one concept *contract_infomation*, RiMOM triggers a special function to search for the possible mapping expression. By mapping expression, we mean how the concepts from the source ontology should be organized so that they can be exactly mapped onto the target concept. Formal description of the mapping expression is

$$F(f(e_{Address}), f(e_{Zipcode}), f(e_{telephone})) = f(e_{contract\_\inf ormation})$$

where *f*(*e*) is a function of *e* such as *left*(*e*, *length*), *lowercase*(*e*). Function *F* is a composition function of the input parameters. Currently, for both function *F* and *f*, we only take the type of string into consideration. For function *f*, we define 5 functions including: left, right, mid, lowercase, uppercase, and capitalize. For function *F*, we define the function as string concatenation by different orders of the input parameters.

Figure 5 shows an output of n:1 mapping by using only instance based decision. This is a concept mapping with the source concepts “address”, “zipcode” and “telephone” and the target concept “contract_information”. Each concept is assigned with an id (e.g. #addr), which is used in the expression “upcase(#addr) + #zip + #tele = #ci”. The expression means that the concatenation of uppercase form of “address” and original form of “zipcode” and “telephone” is mapped onto “contact_infomation”. Each candidate mapping is labeled with a score. The highest scored one is proposed as the mapping and the other two top scored are followed as candidates.



```
<mappings strategy="instance based decisioin">
  <conceptmapping score="0.5931" mappingtype="equivalence">
    <source>
      <concept id="#addr">address</concept>
      <concept id="#zip">zipcode</concept>
      <concept id="#tele">telephone</concept>
    </source>
    <target>
      <concept id="#ci">contract_infomation</concept>
    </target>
    <expression>upcase(#addr) + #zip + #tele = #ci</expression>
    <candidate score="0.0541" type="equivalence">
      <source>
        <concept id="#addr">address</concept>
        <concept id="#zip">zipcode</concept>
      </source>
      <target>
        <concept id="#ci">contract_infomation</concept>
      </target>
      <expression>#addr + #zip = #ci</expression>
    </candidate>
    ...
  </conceptmapping>
  ...
</mappings>
```

Figure 5. An example of output by n:1 mapping

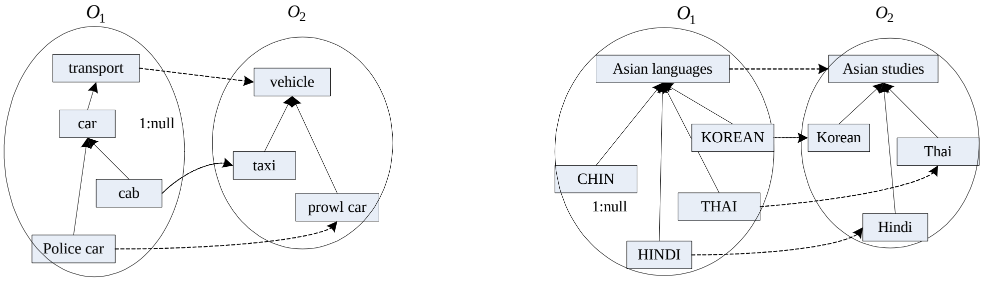
1:null mapping

1:null is a special case. We perform 1:null mapping discovery by using heuristic rules. Table 4 shows some examples of the rules.

Table 4. Examples of rules for 1:null mappings

Categorization	Examples
Threshold	For e_{i1}, if none of its candidate mappings has the predicting value exceeding threshold μ, then we infer that entity e_{i1} has a 1:null mapping. In our experiments, μ is assigned as 0.2.
	For e_{i1}, if all sub-decisions propose different mappings, i.e. the top ranked mappings of them are different, and none of them has the predicting value exceeding threshold λ (we tentatively set it as 0.3), then we infer that entity e_{i1} has a 1:null mapping.
Taxonomy	For e_{i1}, if both its super entity and sub entities can be mapped to the corresponding entities in O_2, and in O_2 there is no entity between the target super entity and target sub entities, then we can infer that e_{i1} has a 1:null mapping. See figure 6(a) for an example, for the concept “car”, its super concept “transport” and sub concepts “cab” and “police car” have mapping concepts in O_2. But in O_2, there is no concept between the concept “vehicle” and concepts “taxi” and “prowl car”. Then we say that concept “car” has a 1:null mapping.

	If entity e_{i1} has a corresponding entity e_{i2} in O_2, and the number of sub concepts of e_{i1} is greater than that of e_{i2}, then we infer that there might be 1:null mappings for sub concepts of e_{i1}. See figure 6(b) for an example, concept “Asian languages” has a mapping to “Asian studies”, and “Asian languages” has four sub concepts but “Asian studies” only has three sub concepts, then there might be one sub concept of “Asian languages” has 1:null mapping. We use the combined predicting value as the metric to judge which one entity has a 1:null mapping. The lower predicting value the entity has, the higher probability it has a 1:null mapping.



(a) (b)

Figure 6. Examples of 1:null mappings

null:1 mapping

The discovery of null:1 mapping is straightforward. When there is no entities mapped to e_{i2}, we say that for entity e_{i2}, there is a null:1 mapping.

2. Mapping refinement

In mapping refinement, we focus on refine the generated mapping by utilizing rules.

In this step, we aim to remove the top ranked but ‘unreasonable’ mappings. We also tried to highlight the mappings that are low ranked but seem ‘reasonable’ mappings. We use following four example cases to explain how we refine the generated mappings.

Case 1: Concept e_{i1} has a mapping to concept e_{i2}, and both its super concept e_{i1}^{p} and sub concept e_{i1}^{s} have mappings to the super concept e_{i2}^{p} of e_{i2}. The three mappings are contradictive. There might exist a mistake mapping. We define the rule in this case that the mapping e_{i1}^{s} to e_{i2}^{p} is a mistake mapping.

Case 2: For concept-pair e_{i1} and e_{i2}, its super concept e_{i1}^{p} and sub concept e_{i1}^{s} respectively has a mapping to the super concept e_{i2}^{p} and sub concept e_{i2}^{p} of concept e_{i2}. But e_{i1} does not have a top ranked mapping to e_{i2}. It has a mapping to concept e_{k2} that is scored higher than the mapping to e_{i2}. Then if the difference of their scores is slight and is under a threshold, we switch to propose the mapping of e_{i1} to e_{i2} rather than e_{i1} to $e_{k2.}$

Case 3: We make use of property mapping to refine concept mapping. For each generated concept mapping e_{i1} to e_{i2}, we checks mappings of their properties. The idea is to give a penalty for those concept mappings when their properties do not have mappings respectively. We calculate a score that indicates the percentage of correspondingly mapped properties in all of their properties. After that, we multiply the combined predicting value of mapping e_{i1} to e_{i2} by the score. Finally, we re-rank mappings for each concept.

Case 4: We make use of concept mappings to refine property mappings. For each property mapping e_{i1} to e_{i2}, we check its “domain” and “range”. We check whether their “domain” is coincide or whether there is a concept mapping between their “domain” concepts (in most cases, domain of property is concept). We also check whether their “range” have the same type (such as data type or object type). For data type, we again check whether they are the same data type. For object type, we again check whether their objects are concepts, and then check whether the concepts have a mapping. Next, we calculate a score after checking and multiply the combined predicting value of mappings e_{i1} to e_{i2} by the score. Finally, we re-rank mappings for each property.

We also exploit the rules defined for constraint based decision. Details of the rules defined for constraint based decision can refer to section 3.4.

5 Experiments and Evaluation

In this section, we first present our experiment design. Next, we give the experimental results on five data sets. After that, we compare RiMOM with existing methods. The implementation of RiMOM was coded in Java.

5.1 Experiment design

1. Evaluation measures

In the experiments of mapping, we conducted evaluations in terms of precision and recall. The measures are defined as follows:

Precision(*P*): It is the percentage of correct discovered mappings in discovered mappings.

Recall(*R*): It is the percentage of correct discovered mappings in correct mappings.

$$P = \frac{|m_a \cap m_m|}{|m_a|}, \quad R = \frac{|m_m \cap m_a|}{|m_m|}$$

where, m_a are mappings discovered by RiMOM and m_m are mappings assigned manually (we view the manually assigned mappings as correct mappings).

However, we note that it is difficult to directly port them to our scene, because n:1 mapping should not be judged by only correct or incorrect. Therefore, allowing for n:1 mapping, we extend the precision and recall as:

$$P = \frac{\sum_i m_i * f_p}{m_a}, m_i \in (m_a \cap m_m), \quad R = \frac{\sum_i m_i * f_c}{m_m}, m_i \in (m_a \cap m_m)$$

where, $f_p = m_{ai} \cap m_{mi} / m_{ai}$ is the proportion of correct items in the discovered mapping m_{ai}. $f_c = m_{ai} \cap m_{mi} / m_{mi}$ denotes the proportion of correctly discovered items in correct mapping m_{mi}. For example, the correct mapping is “Location+Zipcode+Email”→“Address” and the discovered mapping is “Location+Department+Phone+Email”→“Address”. Then we obtain f_{p}=2/4=0.5, f_{c}=2/3=0.667.

2. Data Sets

We tried to collect heterogeneous ontologies from different sources. Totally, we collected five data sets.

The Course Catalog ontology Ⅰ. It describes courses at Cornell University and Washington University. The ontologies of Course Catalog Ⅰ have 34-39 concepts, and are similar to each other.

The company Profile. It uses ontologies from Yahoo.com and The Standard.com and describes the business of the two companies.

The Employee Ontology. It describes employee information. Instances of the two ontologies have little overlap data.

The Sales Ontology. It describes sales information. Instances of the two ontologies have some overlap data.

EON. It includes 19 ontologies. The ontologies are about domain of Bibliographic reference.

Course Catalog Ⅰ and Company Profile are designed by Doan [11], and they were downloaded from http://anhai.cs.uiuc.edu/archive/summary.type.html. EON is from the 2004 Evaluation of Ontology-based Tools workshop at http://co4.inrialpes.fr/align/Contest/. We also created two data sets from real world databases: Employee Ontology and Sales Ontology. For each database, we created two heterogeneous ontologies according to the schema, and then translate records from the database into instances of the two ontologies.

Except for EON data set, the other four data sets respectively contain two heterogeneous ontologies, and thus the task is to map them onto each other. In EON, there are 26 ontologies used for the evaluations in the 2004 Evaluation of Ontology-based Tools workshop. One of the ontologies is chosen as target ontology (also called reference ontology in the EON workshop). The task is to map all the other 25 ontologies onto the reference one. In the final evaluation, however, only 19 mapping tasks are tested. On one ontology we met the problem of parsing error. Then we left it out from the data set. Finally, we included the 18 source ontologies and the target ontology in EON data set.

The entity names defined in the two ontologies of Course Catalog Ⅰ are similar to each other and those in Company Profile are not. The two data sets are used to test the effectiveness of name based decision. Instances of the two ontologies in Employee Ontology have little overlap data and those in Sales Ontology have some overlap. The two data sets are used to test the effectiveness of instance based decision. EON has 19 ontologies and 18 mapping tasks. It is designed to test many different kinds of mapping tasks. See [44] for details.

We manually created mapping for the Employee Ontology and Sale Ontology. Course Catalog Ⅰ, Company Profile, and EON include the 'correct' mappings in the data sets.

Table 5 shows the statistics on the data sets. The columns represent respectively data set, ontologies in the data sets, number of concepts, properties, manual mapping, and instances in the ontologies. Course Catalog Ⅰ, Company Profile, and EON are designed only for 1:1 mapping evaluation. So, our evaluation and comparison focus on the 1:1 mapping on them.

We see that in the first four data sets, the concept numbers of the two ontologies are significant different, in particular in the Company Profile: 333:115. Furthermore, the attribute numbers of the two ontologies are different. The big difference means the different nature of these ontologies and also means that it might be difficult to predict the 'correct' mapping between them.

3. Experiments setup

We used name based decision and instance based decision as baseline methods to test RiMOM. We also evaluated the effect of user interaction. User interaction is expressed by initial points, which means that several mappings are assigned before running the mapping discovery process (about 2-5 mappings are assigned). We performed the four kinds of processes on each data set.

- **Name based decision**. It only uses entity names as information to determine the mappings.
- **Instance based decision**. It only uses instances as information to discover mapping.
- **RiMOM**. It exploits the proposed approach in this paper to discover mapping.
- **RiMOM with initial points**. It introduces the user interaction into RiMOM by assigning several

mappings before running the mapping process.

For each method, we evaluated the performance of 1:1, n:1 and overall.

Table 5. Statistics on data sets (%)

Data set	Ontology	Concept	Property	Manual	Instance
Course Catalog Ⅰ	Cornell	34	0	34	1526
	Washington	39	0	37	1912
Company Profiles	Standard.com	333	0	236	13634
	Yahoo.com	115	0	104	9504
Employee Ontology	Ontology 1	51	218	47	5000
	Ontology 2	45	186	45	5000
Sales Ontology	Ontology 1	44	126	44	3000
	Ontology 2	59	163	52	3000
EON	Reference	33	59	--	76
	101	33	61	91	111
	103	33	61	91	111
	104	33	61	91	111
	201	34	62	91	111
	202	34	62	91	111
	204	33	61	91	111
	205	34	61	91	111
	221	34	61	91	111
	222	29	61	91	111
	223	68	61	91	111
	224	33	59	91	0
	225	33	61	91	111
	228	33	0	33	55
	230	25	54	75	83
	301	15	40	61	0
	302	15	31	48	0
	303	54	72	49	0
	304	39	49	76	0

5.2 Experimental results

1. Experiments

We evaluated the performance of our methods and effectiveness of user interaction on the five data sets. For short, we use Cornell and Wash to denote the course ontology of Cornell University and Washington University; Standard and Yahoo to denote company ontology of Standard.com and Yahoo.com; E1 and E2 to denote employee ontology 1 and employee ontology 2; Sale1 and Sale2 to denote Sales ontology 1 and Sales ontology 2; and Ref to denote the Reference Ontology in EON. Table 6-8 show the results of name based decision, instance based decision, and RiMOM on the five data sets respectively. Table 9 shows the results of RiMOM with initial points on the first four data sets. We did not evaluate RiMOM with initial points on EON. There are two reasons: the baseline methods and RiMOM already achieve high accuracy on several mapping tasks in EON and the number of entity in ontologies of EON is small compared to the other data sets.

Table 6. Precision and recall of name based decision (%)

Data set	Mapping	1:1		n:1		Overall	
		Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Course	Cornell to Wash	85.29	85.29	-	-	85.29	85.29
	Wash to Cornell	79.49	83.78	-	-	79.49	83.78
Company	Standard to Yahoo	64.00	72.40	-	-	64.00	72.40
	Yahoo to Standard	67.38	73.26	-	-	67.38	73.26
Employee	E1 to E2	85.60	78.00	50.50	57.00	69.49	64.30
	E2 to E1	76.83	83.89	47.30	62.56	66.57	72.78
Sales	Sale1 to Sale2	76.30	70.50	58.30	59.00	68.50	62.50
	Sale2 to Sale1	81.88	76.17	63.20	71.12	79.44	75.07
EON	101 to Ref	97.00	100.00	-	-	97.00	100.00
	103 to Ref	97.00	100.00	-	-	97.00	100.00
	104 to Ref	97.00	100.00	-	-	97.00	100.00
	201 to Ref	2.00	2.00	-	-	2.00	2.00
	202 to Ref	2.00	2.00	-	-	2.00	2.00
	204 to Ref	93.00	96.00	-	-	93.00	96.00
	205 to Ref	45.00	46.00	-	-	45.00	46.00
	221 to Ref	97.00	100.00	-	-	97.00	100.00
	222 to Ref	91.00	95.00	-	-	91.00	95.00
	223 to Ref	93.00	96.00	-	-	93.00	96.00
	224 to Ref	97.00	100.00	-	-	97.00	100.00
	225 to Ref	97.00	100.00	-	-	97.00	100.00
	228 to Ref	100.00	100.00	-	-	100.00	100.00
	230 to Ref	79.00	99.00	-	-	79.00	99.00
	301 to Ref	52.00	80.00	-	-	52.00	80.00
	302 to Ref	34.00	67.00	-	-	34.00	67.00
	303 to Ref	40.00	79.00	-	-	40.00	79.00
	304 to Ref	77.00	95.00	-	-	77.00	95.00

Table 7. Precision and recall of instance based decision (%)

Data set	Mapping	1:1		n:1		Overall	
		Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Course	Cornell to Wash	75.00	61.76	-	-	75.00	61.76
	Wash to Cornell	93.55	78.38	-	-	93.55	78.38
Company	Standard to Yahoo	80.00	87.50	-	-	80.00	87.50
	Yahoo to Standard	71.40	88.90	-	-	71.40	88.90
Employee	E1 to E2	55.00	43.50	40.50	66.50	52.50	50.00
	E2 to E1	64.50	56.38	54.68	63.49	61.27	59.64
Sales	Sale1 to Sale2	88.50	79.00	78.50	65.00	84.50	74.80
	Sale2 to Sale1	84.76	73.32	81.09	70.5	82.49	71.24
EON	101 to Ref	97.00	100.00	-	-	97.00	100.00
	103 to Ref	97.00	100.00	-	-	97.00	100.00
	104 to Ref	97.00	100.00	-	-	97.00	100.00
	201 to Ref	90.00	93.00	-	-	90.00	93.00
	202 to Ref	46.00	43.00	-	-	46.00	43.00
	204 to Ref	95.00	98.00	-	-	95.00	98.00
	205 to Ref	70.00	68.00	-	-	70.00	68.00
	221 to Ref	97.00	100.00	-	-	97.00	100.00

	222 to Ref	90.00	93.00	-	-	90.00	93.00
	223 to Ref	95.00	98.00	-	-	95.00	98.00
	224 to Ref	84.00	87.00	-	-	84.00	87.00
	225 to Ref	96.00	99.00	-	-	96.00	99.00
	228 to Ref	91.00	91.00	-	-	91.00	91.00
	230 to Ref	78.00	97.00	-	-	78.00	97.00
	301 to Ref	36.00	54.00	-	-	36.00	54.00
	302 to Ref	28.00	46.00	-	-	28.00	46.00
	303 to Ref	30.00	50.00	-	-	30.00	50.00
	304 to Ref	58.00	70.00	-	-	58.00	70.00

Table 8. Precision and recall of RiMOM (%)

Data set	Mapping	1:1		n:1		Overall	
		Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Course	Cornell to Wash	91.18	91.18	-	-	91.18	91.18
	Wash to Cornell	88.89	86.49	-	-	88.89	86.49
Company	Standard to Yahoo	81.00	89.30	-	-	81.00	89.30
	Yahoo to Standard	73.12	89.74	-	-	73.12	89.74
Employee	E1 to E2	86.56	84.0	71.66	90.50	82.61	85.89
	E2 to E1	78.38	84.43	63.21	67.39	73.00	78.59
Sales	Sale1 to Sale2	94.00	91.50	88.60	93.00	91.60	92.00
	Sale2 to Sale1	89.52	86.46	73.63	71.17	86.37	83.44
EON	101 to Ref	97.00	100.00	-	-	97.00	100.00
	103 to Ref	97.00	100.00	-	-	97.00	100.00
	104 to Ref	97.00	100.00	-	-	97.00	100.00
	201 to Ref	88.00	90.00	-	-	88.00	90.00
	202 to Ref	41.00	41.00	-	-	41.00	41.00
	204 to Ref	94.00	98.00	-	-	94.00	98.00
	205 to Ref	62.00	64.00	-	-	62.00	64.00
	221 to Ref	97.00	100.00	-	-	97.00	100.00
	222 to Ref	91.00	95.00	-	-	91.00	95.00
	223 to Ref	93.00	96.00	-	-	93.00	96.00
	224 to Ref	96.00	99.00	-	-	96.00	99.00
	225 to Ref	97.00	100.00	-	-	97.00	100.00
	228 to Ref	100.00	100.00	-	-	100.00	100.00
	230 to Ref	76.00	95.00	-	-	76.00	95.00
	301 to Ref	92.00	77.00	-	-	92.00	77.00
	302 to Ref	79.00	54.00	-	-	79.00	54.00
	303 to Ref	78.00	75.00	-	-	78.00	75.00
	304 to Ref	96.00	95.00	-	-	96.00	95.00

Table 9. Precision and recall of RiMOM with initial points (3 random non-leaf points) (%)

Data set	Mapping	1:1		n:1		Overall	
		Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Course	Cornell to Wash	94.12	94.12	-	-	94.12	94.12
	Wash to Cornell	94.74	97.30	-	-	94.74	97.3
Company	Standard to Yahoo	83.50	90.50	-	-	83.50	90.50
	Yahoo to Standard	73.46	90.38	-	-	73.46	90.38
Employee	E1 to E2	88.50	86.30	76.50	84.00	85.00	85.40
	E2 to E1	81.48	85.51	67.82	64.90	77.16	79.20

Sales	Sale1 to Sale2	95.80	94.80	92.50	91.00	94.30	93.09
	Sale2 to Sale1	91.06	88.24	80.36	78.92	88.48	85.72

On Sales Ontology, we respectively give the performances of concept mapping and property mapping. Figure 7 shows the experiment results.

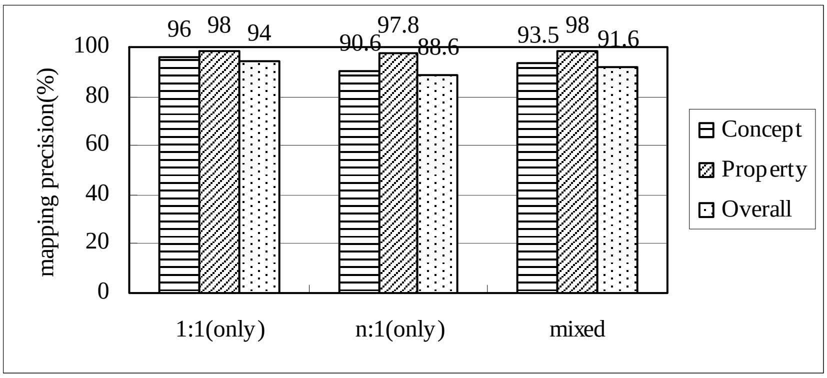
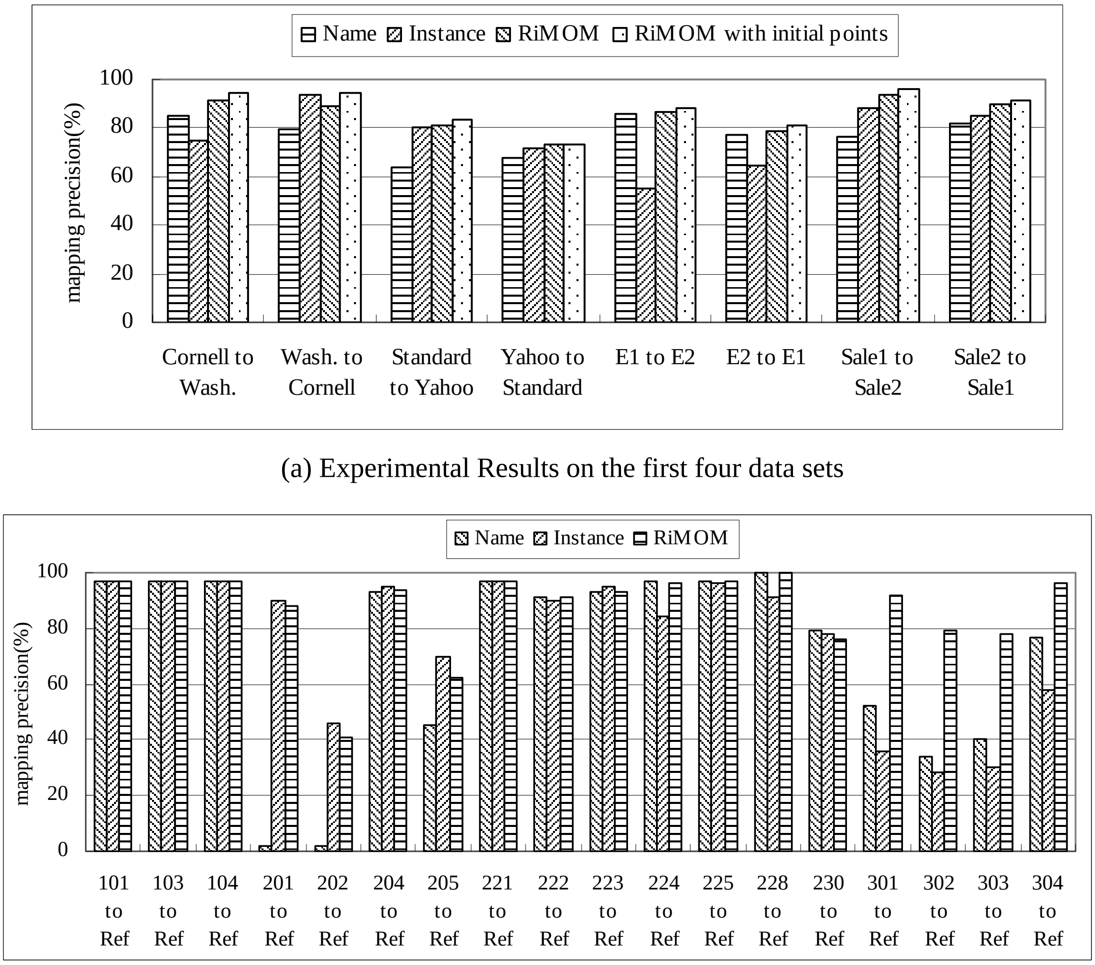


Figure 7. Precision of concept/property/mixed mapping on Sales Ontology

Figure 8 shows the comparison of the four methods on the five data sets. Specifically, figure 8(a) shows the comparison of the four methods on the first four data sets and figure 8(b) shows the comparison of name based decision, instance based decision, and RiMOM on EON.



(a) Experimental Results on the first four data sets

(b) Experimental results on EON

Figure 8. Experimental results

The four bars in figure 8(a) on each data set (from left to right) respectively represent the precisions produced by name based decision, instance based decision, RiMOM, and RiMOM with initial points. And the three bars in figure 8(b) denote precisions produced by name based decision, instance based decision, and RiMOM.

We see that RiMOM can achieve high performance in all the tasks. In most tasks, our method significantly outperforms the baseline methods. We conducted sign tests on the results. The *p* values are significantly smaller than 0.01, indicating that the improvements are statistically significant.

2. Discussions

We here focus on the analysis of the experimental results. Since the mapping tasks in the first four data sets are quite different from those in EON, we conduct the analysis for the first data sets and EON separately.

(1) **High performance**. In mapping on Course Catalog Ⅰ, Company Profile, Employee Ontology, and Sale Ontology, precisions range from 73% to 91.6% and recalls range from 83.44% to 92%. It seems that the proposed method is effective for ontology mapping. In EON, for most of the mapping tasks, we obtained good results. By average, the precision and recall are 87.28% and 87.72%.

(2) **Contribution of Instances**. On Course Catalog Ⅰ, Company Profile, and Sale Ontology, instance based decision outperforms name based decision (from +3% to +16% on precision except for Cornell to Wash). But we also note that there may be a lower performance by instance based decision when instances of the two ontologies have few common ones. Employee Ontology has exactly the problem. By using instance based decision, the precisions of E1 to E2 and E2 to E1 are only 52.5% and 61.27% in terms of precision, respectively. In EON, Instance based decision averagely outperforms the name based decision by +6.59% in terms of precision and +2.06% in terms of recall.

(3) **Improvement over baseline methods.** Compared to name based decision, RiMOM obtains a significant improvement (ranging from +6.91% to +33.72% with average +15.60% in terms of precision and ranging from +3.23% to +47.2% with average +19.49% in terms of recall). By the comparison with instance based decision, the improvement is also clear (ranging from +1.25% to +57.35% with average +15.02% in terms of precision and ranging from +0.94% to +47.64% with average +26.79% in terms of recall) except for the mapping from Wash to Cornell. The biggest problem of name based decision and instance based decision is that they strongly depend on only one kind of information attached to entities. This makes them sensitive to data set. For example, name based decision can reach 85.29% (mapping from Cornell to Wash) when entity names are similar. But it drops to 64% (mapping from Standard to Yahoo) when names are not similar. The same problem also occurs in instance based decision. RiMOM smoothes such bias by integrating all kinds of information. In EON, RiMOM outperforms both name based decision (averagely by +14.25% in terms of precision and +6.19% in terms of recall) and instance based decision (averagely by +21.78% in terms of precision and +8.37% in terms of recall).

(4) **Effectiveness of user interaction.** Since ontology is the foundation of the semantic web, the quality of ontology mapping is very important for interoperability. Therefore, targeted user interaction is also necessary. Many proposed techniques could be applied to the interaction: user feedback, specific constraints, and initial points. We adopt the method of initial points in our experiments. The average improvement by initial points is +3.56% in terms of precision and

+2.74% in terms of recall.

(5) **Error analysis.** We conducted error analysis on the results.

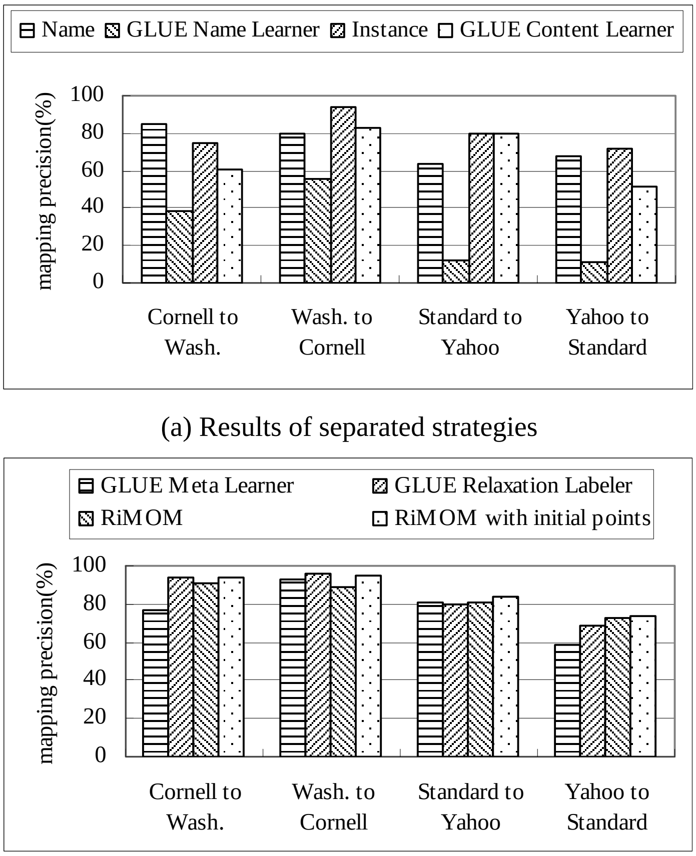
For 1:1 mapping, there are mainly three types of errors. More than 36% of the errors were due to mappings that the source entity was mapped to the super entity of the target entity. About 17.65% of the errors occurred when names of the source and target entities were absolutely different and common instances of the entities were very few either. Furthermore, 11.26% of the errors were results of the incorrect filtering by the constraint rules that are used in the mapping process.

For n:1 mapping, about 33% of the errors were due to missing one or two source entities. About 25% of the errors were results of including one mistake entity in the source entities. 18% of the errors were failures of finding the correct mapping expression.

5.3 Comparison with Existing Methods

1. Comparison with GLUE

We conducted the comparison with GLUE. Results of GLUE are from [11], where possibly we use the same data sets and evaluation metrics.



(a) Results of separated strategies

(b) Results of GLUE and RiMOM

Figure 9. Comparison between GLUE and RiMOM

Name — Name based decision in RiMOM, Instance — Instance based decision in RiMOM

GLUE Name Learner — predict mapping using entity name in GLUE

GLUE Content Learner — predict mapping using instances in GLUE

GLUE Meta Learner, GLUE Relaxation Labeler – two strategies to determine mappings in GLUE

Given two ontologies, GLUE tries to find the most similar concept in target ontology for each concept in the source ontology. GLUE contains two base learners: Content Learner and Name Learner, that respectively corresponds to instance based decision and name based decision in RiMOM. It uses a strategy, called Meta Learner, to linearly combine the base learners. Finally, GLUE provides a Relaxation Labeler to search for the mappings that best satisfy the given domain constraints and heuristic knowledge.

Figure 9 shows the comparison of GLUE and RiMOM. In figure 9(a), we compared individual strategies in GLUE and RiMOM, i.e. Name based decision vs. Name Learner and Instance based decision vs. Content Learner. In figure 9(b) we compared the final mapping results of Meta Learner, Relaxation Labeler, RiMOM, and RiMOM with initial points.

We see that name based decision in RiMOM significantly outperforms Name Learner in GLUE and instance based decision also reaches higher precision than Content Learner. In most cases, RiMOM and RiMOM with initial points both outperform Meta Learner. By comparison with Relaxation Labeler, RiMOM obtains better performance on the Company ontologies and is competitive on Course ontologies.

2. Comparison with EON results

We also conducted the comparison with the results of 2004 EON. We compared our results with those produced by “karlsruhe2”, “umontreal”, “fujitsu”, and “stanford”. The results are from http://co4.inrialpes.fr/align/Contest/results/. See also [44] for details. (RiMOM did not actually participate in EON2004. We just use the data set for evaluation and the results for comparison.)

Table 10 shows the comparison between results of 2004 EON and the results of RiMOM. In the table, we give the precisions and recalls. Notation “n/a” means that there is no result for evaluation.

Table 10. Comparison between results of 2004 EON and the results of RiMOM (%)

Algorithm	**karlsruhe2**		**umontreal**		**fujitsu**		**stanford**		**RiMOM**	
Mapping	**Prec.**	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**
101 to Ref	n/a	n/a	59.00	97.00	99.00	100.00	99.00	100.00	97.00	100.00
103 to Ref	n/a	n/a	55.00	90.00	99.00	100.00	99.00	100.00	97.00	100.00
104 to Ref	n/a	n/a	56.00	91.00	99.00	100.00	99.00	100.00	97.00	100.00
201 to Ref	43.00	51.00	44.00	71.00	98.00	92.00	100.00	11.00	88.00	90.00
202 to Ref	n/a	n/a	38.00	63.00	95.00	42.00	100.00	11.00	41.00	41.00
204 to Ref	62.00	100.00	55.00	90.00	95.00	91.00	99.00	100.00	94.00	98.00
205 to Ref	47.00	60.00	49.00	80.00	79.00	63.00	95.00	43.00	62.00	64.00
221 to Ref	n/a	n/a	61.00	100.00	98.00	88.00	99.00	100.00	97.00	100.00
222 to Ref	n/a	n/a	55.00	90.00	99.00	92.00	98.00	95.00	91.00	95.00
223 to Ref	59.00	96.00	59.00	97.00	95.00	87.00	95.00	96.00	93.00	96.00
224 to Ref	97.00	97.00	97.00	100.00	99.00	100.00	99.00	100.00	96.00	99.00
225 to Ref	n/a	n/a	59.00	97.00	99.00	100.00	99.00	100.00	97.00	100.00
228 to Ref	n/a	n/a	38.00	100.00	91.00	97.00	100.00	100.00	100.00	100.00
230 to Ref	60.00	95.00	46.00	92.00	97.00	95.00	99.00	93.00	76.00	95.00
301 to Ref	85.00	36.00	49.00	61.00	89.00	66.00	93.00	44.00	92.00	77.00
302 to Ref	100.00	23.00	23.00	50.00	39.00	60.00	94.00	65.00	79.00	54.00
303 to Ref	85.00	73.00	31.00	50.00	51.00	50.00	85.00	81.00	78.00	75.00
304 to Ref	91.00	92.00	44.00	62.00	85.00	92.00	97.00	97.00	96.00	95.00
Average	72.90	72.30	51.00	82.28	89.22	84.17	91.78	79.78	87.28	87.72

We see that RiMOM significantly outperforms karlsruhe2 and umontreal and is competitive with fujitsu and stanford.

3. Discussions

(1) By the comparison of individual strategies in GLUE and RiMOM, name based decision or instance based decision clearly outperforms Name Learner and Content Learner (from +41.95% to +512.55% by name based decision and about +15% by instance based decision). We believe that the improvement on name based decision lies in the usage of thesaurus and statistic dictionary. GLUE uses classifier to compute the similarity between entity names. However, classifier may not be effective on short text content. For the instance based decision in RiMOM, its advantage comes from the normalization and additional knowledge from NLP.

(2) RiMOM outperforms Meta Learner of GLUE on two mapping tasks: Cornell to Wash (by +18.42%) and Yahoo to Standard (by +23.93%). RiMOM underperforms Meta Learner on Wash to Cornell (by -4.42%). We think the advantage of RiMOM relies on that of our individual strategies. Experiments also show that RiMOM is only competitive with Relaxation Labeler. The reason may be that Relaxation Labeler makes full use of the domain constraints and heuristic knowledge, which effectively improve the mapping precision. It also means that the combination strategy and the heuristic rules used in RiMOM are not sufficient. That is also one of our future works.

(3) On EON, RiMOM significantly outperforms karlsruhe2 (+19.72% on precision and +21.33% on recall by average) and umontreal (+71.13% on precision and +6.12% on recall by average). Compared to fujitsu, RiMOM averagely outperforms it in terms of recall by +4.22%, but underperforms in terms of precision by -2.18%. By the comparison of stanford, RiMOM averagely outperforms it in terms of recall by +9.96%, but underperforms in terms of precision by -4.90%. The comparison indicates that RiMOM still needs to improve its precision.

6 Related works

In this section, we review the research efforts that are related to this paper. We clarify the related works from four aspects: Schema Matching, Ontology Mapping, Complex Matching and Efficient Mapping. There are a number of available systems that address schema matching or ontology mapping. A complete review of this subject is therefore outside the scope of this paper. We present some of them through their principles and availabilities.

6.1 Schema Matching

Many works have addressed the schema matching problem (e.g. [9, 10, 15, 16, 18, 19, 23], see also [32] for a survey). Some researches define a unified schema, and then employ a centralized approach to map all data sources onto the unified one. The approach may not be flexible enough to scale up to the Semantic Web, because ontology mapping is a more dynamic knowledge sharing or interoperability problem.

For example, COMA is a generic schema matching tool supporting different applications and multiple schema types [9]. It provides an extensible library of matching algorithms, a framework for combining match algorithms in a flexible way, and a platform for evaluating the effectiveness of different algorithms. Another feature of COMA tool is the capability to perform iterations in matching process.

Rondo is an environment for model (e.g. database schema) engineering which provides many unit primitives for manipulating models (extract, restrict, delete) and the way to compose them [24]. It converts schemas (SQL DDL, XML) into directed graphs whose nodes are candidate mapping pairs and arcs are shared properties. Arcs are weighted by their relevance between nodes. Rondo mainly uses entity names and taxonomy structure to determine the mappings.

Cupid has implemented a generic schema matching algorithm by combining linguistic and structural schema matching techniques. It computes normalized similarity with the assistance of a precompiled thesaurus. Input schemas are encoded as graphs. Nodes represent schema entities and are traversed in a combined bottom-up and top-down manner. In comparison with the other hybrid matchers e.g. DIKE [29], Cupid performs better in the sense of mapping quality.

Ontology mapping is different from schema matching [20, 39]. First, by the comparison of database schemas, ontology provides higher flexibility and more explicit semantics for defining data. Secondly, database schemas are not sharable or reusable, and usually are defined for a specific database, whereas ontology is by nature reusable and sharable. Thirdly, ontology development is becoming a more and more decentralized procedure. Finally, schema matching should take into account the effects of each change on the data (addition of a new class); while in ontology, the number of the knowledge representation primitives is much larger and more complex: cardinality constraints, inverse properties, transitive properties, disjoint classes, type-checking constraints, etc.

Although there are significant differences between schema matching and ontology mapping, many of the methods and technologies developed for schema matching can be applied or adapted to ontology mapping. Actually, some of the systems presented above are making such adoption.

6.2 Ontology Mapping

Ad-hoc rules is used in most previous works to map ontologies (as surveyed in [13, 39]). This approach allows limited flexibility in ontology integration, but mostly does not provide automatic mapping. We present some of these systems that provide automatic ontology mapping.

In the research area of knowledge engineering, a number of ontology integration methods and tools are proposed and have been developed. Among them, Anchor-PROMPT [26, 27] and Chimaera [22] are the few which have working prototypes [14].

Anchor-PROMPT is a tool for ontology merging and mapping [26, 27]. It contains a sophisticated prompt mechanism for possible mapping entities. The Anchor-PROMPT mapping algorithm takes as input two ontologies and a set of anchor-pairs of related entities, which are identified with the help of name based decision or defined by the user (similar to the initial points method). Then it refines them based on the ontology structures and user feedback. Their focus lies on ontology merging i.e. how to create a new ontology out of two.

Chimaera is an environment for merging and testing large ontologies [22]. Mapping in the system is performed as one of the major subtasks of merging. Chimaera searches for merging candidates as pairs of mapping entities, by using the information of entity names, entity definition, possible acronym, and expanded forms. It also has techniques to identify entities that should be related by subsumption, disjointness, etc. Chimaera does not make full use of the taxonomy structure, constraints and instances to refine the mappings, which may limit its potential applications.

The other category of work for ontology interoperability finds the mapping by employing machine learning methods. Each concept in ontology being regarded as a class, this method uses instances in target ontology as training samples to train a classifier and then uses instances of the source ontology as test samples to predict their correspondences.

For example, GLUE aims to automatically find ontology mapping for data integration [10, 11]. It uses machine learning techniques to discover mappings. It first applies statistical analysis to available data (joint probability distribution computation), and then generates a similarity matrix, based on the probability distributions. After that, it uses "constraint relaxation" to obtain a mapping from similarity matrix. RiMOM is similar to GLUE but with different concentration. First, RiMOM uses different methods to determine the optimal mappings with the available clues and constraints. GLUE uses Relaxation Labeling to handle wide variety of constraints and RiMOM uses risk minimization to search for the optimal mappings from the results of multiple strategies. Secondly, they exploit different methods in the mapping process. For example, on entity name, RiMOM exploits WordNet and statistical technique and GLUE exploits text classification methods; on data instances, RiMOM preprocesses them by normalization and NLP techniques, while GLUE does not; moreover GLUE does not provide an interface for user interaction; finally, Relaxation Labeler in GLUE seems more effective than multi-decision combination in RiMOM.

Some other methods exploit text categorization to automatically assign documents to the concept in the ontology and use the documents to calculate the similarities between concepts in ontologies [35]. Zhang et al make use of Support Vector Machines for finding mapping between web taxonomies [42]. They exploit the availability of two taxonomies to build classifier by transductive learning. They also propose a method, called cluster shrinkage, to enhance the classification. These two methods, however, do not efficiently exploit other information, such as entity name, constraints and taxonomy context.

Some other research efforts also include: Calvanese et al propose an ontology integration framework [5]. They provide semantics for ontology integration by defining sound and complete semantic conditions for each mapping rule. They focus on the mapping representation. Park et al have extended Protégé to support mapping two domain ontologies [31]. In this method, a valuable set of desiderata and mapping dimensions are defined. MAFRA [20] and RDFT [28] are two representation initiatives for mappings. Both of them have similar logic to represent the mappings. And both of them define a meta-ontology for mapping.

Bouquet et al formulate the problem of ontology heterogeneity as that of discovering, expressing and using ontology mapping [4]. They aim to provide a common framework for the future work in this research area and give the definition of many of the terms used in the area.

So far, existing research efforts focus on various aspects that are concerned with ontology integration (merging, mapping, translation and representation). In ontology mapping, different systems may exploit different information or different methods. Comparing with them, three features make RiMOM different: (1) RiMOM can combine almost all kinds of information in ontology. RiMOM is a general framework, which make it easily to incorporate new mapping algorithms. (2) RiMOM exploits NLP techniques and normalization in the preprocessing of mapping. These two strategies improve the performance of RiMOM. (3) RiMOM finds mappings of multiple kinds of cardinalities and most of existing systems take only 1:1 mapping into account.

6.3 Complex Matching

The other kind of work related to us includes: multi-matcher system, complex matcher discovery [8, 41]. They both focus on the complex mapping discovery between database schemas.

For example, iMAP formulates schema matching as a search in a very large or infinite match space and then makes the search efficient by employing a set of searchers. Each of the searchers is designed to discover a specific type of complex matches. iMAP exploits beam search and equation discovery to mine the complex text mapping and numeric function mapping [8]. By concerning with the discovery of complex mapping cardinality, RiMOM is also similar to iMAP. iMAP emphasizes particularly on complex expression and function discovery while RiMOM focuses on entity mapping itself (n:1).

6.4 Efficient Mapping

QOM considers the quality of the mapping results as well as the run-time complexity [12]. The hypothesis is that the mapping algorithms may be streamlined so that the loss of quality (compared to a standard based line) is marginal, but the improvement of efficiency is so tremendous that it allows for the ad-hoc mapping of large-size, light-weight ontologies. The evaluation was promising. QOM can reach high quality mapping quickly. But QOM focuses on only simple mapping such as 1:1 mapping and concept level mapping.

7 Conclusions

In this paper, we have investigated the problem of ontology mapping. In terms of Bayesian decision theory, we have formulated the problem as that of decision making. We have proposed an approach called RiMOM to perform the task. Using multiple decisions, we have been able to make an implementation of the approach. Furthermore, RiMOM support automatic discovery of mapping with different cardinalities including n:1, 1:null, null:1, and 1:1. Experimental results show that our approach can significantly outperform baseline methods for ontology mapping. By the comparison with GLUE, we observed an improvement on mapping accuracy. By the comparison with EON results, we see that RiMOM significantly outperforms karlsruhe2 and umontreal, and is competitive with fujitsu and stanford.

As the future work, we plan to make further improvement on the mapping accuracy. We also want to apply the proposed method to applications of semantic interoperability. Apart from that, several challenges for ontology mapping, also being our research interests, include: (1) Mapping representation. A standard language for representing the mapping results is necessary for further using the mapping by different systems. (2) Practical system. A practical system is also required not only to drive the research to its next step but also for the fulfillment of the semantic web vision. (3) Discovery of more sophisticated mappings. The challenge is to discover more sophisticated mappings between ontologies (such as n:m mappings) by exploiting more of the constraints that are expressed in the ontologies (via attributes and relationships, and constraints on them).

Acknowledgement

This work is funded by the National Natural Science Foundation of China under Grant No. 60443002. We owe special thanks to Professor Lu Hong-Jun for his constructive advices. Thanks to anonymous reviewers for their valuable comments.

References

[1] R. Benjamins and J. Contreras. White Paper Six Challenges for the Semantic Web. Intelligent Software Components. Intelligent Software for the Networked Economy (isoco). April, 2002.

[2] J. Berger. Statistical Decision Theory and Bayesian Analysis. Springer-Verlag. 1985

[3] T. Berners-Lee, M. Fischetti, and M. L. Dertouzos. Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web. 1999.

[4] P. Bouquet, J. Euzenat, E. Franconi, L. Serafini, G. Stamou, and S. Tessaris. Specification of A Common Framework for Characterizing Alignment. http://www.inrialpes.fr/exmo/cooperation/kweb/heterogeneity/deli/kweb-221.pdf. 2004

[5] D. Calvanese, G. De Giacomo, and M. Lenzerini. A Framework for Ontology Integration. The Emerging Semantic Web. IOS Press. 2002, 201–214.

[6] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, 2002.

[7] M. Dean, G. Schreiber, S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. Andrea Stein. OWL Web Ontology Language Reference. W3C Recommendation. Available at http://www.w3.org/TR/owl-ref/. 10 February 2004.

[8] R. Dhamankar, Y. Lee, A.H. Doan, A. Halevy, and P. Domingos. iMAP: Discovering Complex Semantic Matches between Database Schemas. In Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. Paris, France: ACM Press. 2004.

[9] H. Do and E. Rahm. Coma: A System for Flexible Combination of Schema Matching Approaches. In Proceedings of 28th International Conference on Very Large Data Bases. Hong Kong, China. 2002.

[10] A.H. Doan, P. Domingos, and A. Halevy. Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. In Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data. 2001.

[11] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to Map Between Ontologies on the Semantic Web. In Proceedings of the 11th World-Wide Web Conference. 2002. 662–673

[12] M. Ehrig and S. Staab. QOM – Quick Ontology Mapping. In Proceedings of the 4th International Semantic Web Conference. Japan. 2004

[13] J. Euzenat. State of the Art on Ontology Alignment. http://www.inrialpes.fr/exmo/ cooperation/kweb/ heterogeneity/deli/. August, 2004.

[14] T. Gruber. Introduction to the Bibliographic Data Ontology. http://www-ksl.stanford.edu/knowledge-sharing/ontologies/html/bibliographic-data.text.html.

[15] J. Kang and J. Naughton. On Schema Matching with Opaque Column Names and Data Values. In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. 2003.

[16] W. Kim and J. Seo. Classifying Schematic and Data Heterogeneity in Multi-Database Systems. *IEEE* Computer, 1991, 24(12):12-18

[17] D. B. Lenat. Cyc: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM, 1995, 38(11). 32-38.

[18] J. Madhavan, P. Bernstein, and E. Rahm. Generic Schema Matching with Cupid. In Proceedings of 27th International Conference on Very Large Data Bases. 2001. 48-58.

[19] J. Madhavan, P. Bernstein, K. Chen, A. Halevy, and P. Shenoy. Corpus Based Schema Matching. In Proceedings of the IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03), 2003.

[20] A. Maedche, B. Moltik, N. Silva, and R. Volz. MAFRA-An Ontology MApping FRAmework in the Context of the Semantic Web. In Proceeding of the EKAW 2002. Siguenza, Spain. 2002.

[21] A. Maedche and S. Staab. Ontology Learning for the Semantic Web. IEEE Intelligent Systems. 2001, 16(2): 72-79.

[22] D. McGuinness, R. Fikes, J. Rice, and S. Wilder. An Environment for Merging and Testing Large Ontologies. In Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning. Colorado. USA. 2000. 483-493.

[23] S. Melnik, H. Molina-Garcia, and E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm. In Proceedings of the 18th International Conference on Data Engineering. 2002.

[24] S. Melnik, E. Rahm, and P. Bernstein. Rondo: A Programming Platform for Model Management. In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. San Diego (CA US), 2003.

[25] Z. B. Miled, Y. W. Webster, N. Li, O. Bukhres, A. K. Nayar, J. Martin, and R. Oppelt. BAO, A Biological and Chemical Ontology for Information Integration. Online Journal of Bioinformatics. 2002, VOL 1. 60-73.

[26] N. F. Noy and M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In Proceedings of the 2000 National Conference on Artificial Intelligence. 2000. 450–455.

[27] N. Noy and M. Musen. Anchor-PROMPT: Using Non-local Context for Semantic Matching. In Proceedings of IJCAI 2001 Workshop on Ontology and Information Sharing. 2001. 63-70

[28] B. Omelayenko. RDFT: A Mapping Meta-Ontology for Business Integration. In Proceedings of Workshop on Knowledge Transformation for the Semantic Web (KTSW 2002) at ECAI'2002. Lyon, France. 2002. 76-83

[29] L. Palopoli, G. Terracina, and D. Ursino. The System DIKE: Towards the Semi-Automatic Synthesis of Cooperative Information Systems and Data Warehouses. In Proceedings of 2000 ADBIS-DASFAA Symposium on Advances in Databases and Information Systems. 2000. 108-117.

[30] P. Pantel and D. Lin. Discovering Word Senses from Text. In Proceedings of 2002 ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2002. 613-619.

[31] J. Y. Park, J. H. Gennari, and M. A. Musen. Mappings for Reuse in Knowledge-based Systems. In Proceedings of the 11th Workshop on Knowledge Acquisition, Modelling and Management (KAW 98). Banff, Canada. 1998.

[32] E. Rahm and P. A. Bernstein. A Survey of Approaches to Automatic Schema Matching. The VLDB Journal, 2001, 10. 334–350.

[33] N. Silva and J. Rocha. Semantic Web Complex Ontology Mapping. In Proceedings of 2003 IEEE/WIC International Conference on Web Intelligence. Halifax, Canada. 2003. 82-100

[34] R. Sproat, A. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards. Normalization of Non-Standard Words: WS'99 Final Report. http://www.clsp.jhu.edu/ws99/.

[35] X. Su. A Text Categorization Perspective for Ontology Mapping. Technical report. 2002.

[36] J. Tang, B. Y. Liang, J. Z. Li, and K. H. Wang. Risk Minimization based Ontology Mapping. 2004 Advanced Workshop on Content Computing (AWCC). Springer-Verlag, LNCS/LNAI. 2004.

[37] K. M. Ting and I. H. Witten. Issues in Stacked Generalization. Journal of Artificial Intelligence Research, 1999, 10. 271-289.

[38] M. Uschold, M. King, S. Moralee, and Y. Zorgios. The Enterprise Ontology. The Knowledge Engineering Review, Special Issue on Putting Ontologies to Use, 1998, 13(1). 31-89.

[39] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Huebner. Ontology-Based Integration of Information – A Survey of Existing Approaches. In Proceedings of IJCAI 2001 Workshop on Ontologies and Information Sharing. 2001.

[40] F. Wiesman, N. Roos, and P. Vogt. Automatic Ontology Mapping for Agent Communication. Technical report. 2001.

[41] L. Xu and D. Embley. Using Domain Ontologies to Discover Direct and Indirect Matches for Schema Elements. In Proceedings of the Semantic Integration Workshop at ISWC-2003. 2003.

[42] D. Zhang and W. S. Lee. Web Taxonomy Integration using Support Vector Machines. In Proceedings of the World-Wide Web Conference (WWW-2004). ACM Press. New York, USA. 2004. 472–481.

[43] Bio-Ontologies. http://www.cs.man.ac.uk/~stevensr/ontology.html. 10 April 2005.

[44] EON2004. http://km.aifb.uni-karlsruhe.de/ws/eon2004/. 8th November 2004.

[45] SchemaWeb. http://www.schemaweb.info/schema/BrowseSchema.aspx. 10 April 2005.

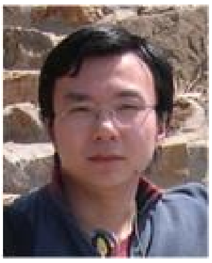
Biography

Jie Tang: born in 1977, Ph. D. candidate. His main research interests include Semantic Web Annotation, Ontology Interoperability, Machine Learning, Text Mining, Information Extraction and Information Retrieval.

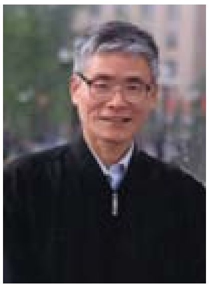
Juanzi Li (associate professor): She got doctor degree from Tsinghua University in 2000. Main research directions include Semantic web，natural language processing and knowledge discovery on internet.

Bangyong Liang: Ph. D. candidate. Research Areas cover Knowledge Base System, Data Mining, XML Technology, Agent Technology and Semantic Web.

Xiaotong Huang: Her research areas cover Knowledge Base System, XML Technology and Semantic Web.



Yi Li: His research interests include Ontology Mapping, Semantic Integration.



Kehong Wang (professor): Before 1992, his research interests include knowledge engineering and distributed knowledge processing. In recent years, he is engaging his research on network computing and knowledge processing. He has published many papers and academic books including <knowledge engineering and knowledge processing system>, <Java technology series books> and so on.