FCA-Mapping: A Method for Ontology Mapping

Liya Fan, Tianyuan Xiao

National CIMS ERC, Department of Automation, Tsinghua University
Beijing 100084, P.R. China
fanfanfly@gmail.com

Abstract. Ontology mapping is a precondition to achieve interoperability between agents or services using different ontologies. Today, a series of methods for ontology mapping have been reported. However in the existing literature, the focus has so far been on computing similarity and establishing equal mapping relation. In practice, it is possible to have other mapping relations besides equal mapping, such as subclass mapping relation. In this paper a new method for ontology mapping, FCA-Mapping, is presented. FCA-Mapping is based on Formal Concept Analysis theory. By computing relation measures between entities of different ontologies, it can automatically establish equal and subclass mapping relations.

Keywords: ontology mapping; relation measure; similarity measure; inclusion measure

1 Introduction

In recent years, ontologies have been widely used for different purposes. As lots of ontologies are developed by different people or organizations, the heterogeneity between different ontologies is inevitable. In order to overcome the heterogeneity and establish interoperability between agents or services which are built on different ontologies, semantic mapping between ontologies is necessary.

Now, a series of methods for ontology mapping have been reported [1]. Some major characteristics of the best known methods are summarized in Table 1.

From Table 1, we can see that the focus of the previous works has been laid exclusively on computing similarity and establishing equal mapping relation. However in practice, there are also other mapping relations besides equal mapping relation, such as subclass mapping relation. In order to overcome the former limitation, we put forward a new method of ontology mapping, named FCA-Mapping, which can perform better in the process of establishing equal and subclass mapping relations through computing relation measures.

This paper is structured as follows: section 2 introduces the definitions of ontology, ontology mapping and relation measure. Section 3 recalls the basics of FCA theory. Section 4 presents a method for computing similarity measure and inclusion measure.

Section 5 puts forward our method for ontology mapping: FCA-Mapping. Section 6 evaluates FCA-Mapping. Section 7 summarizes our contributions.

Table 1. Comparison between existing methods

Name	Entities that can be mapped	Relation	Information that can be used	Major techniques
GLUE[2]	Concepts	equality	Instances, semantic relationship	Machine Learning, Relaxation Labeling
Leyun[3]	Concepts	equality	Instances, semantic relationship	Machine Learning
CAIMAN[4]	Concepts	equality	Instances, semantic relationship	Machine Learning
String Metric[5]	All entities	equality	Entities names	String Metric
IF-Map[6]	Concepts and properties	equality	Class and property names, instances, semantic relationship	information-flow theory
Marc Ehrig[7]	Concepts, properties and instances	equality	Rules suggested by experts	Rules
QOM[8]	Concepts, properties and instances	equality	Names of concepts, properties and instances, rules	Multidimensional scaling, rules
Euzenat[9]	All entities	equality	Entities name, semantic relationship	OL-graph
H-MATCH[10]	Concepts and properties	equality	Class and property names, semantic relationship	H-Model

2 Definitions

In this section, our understanding of ontology and ontology mapping is introduced shortly at first, and then we discuss the definitions of relation measure, similarity measure and inclusion measure which are the bases of establishing semantic mappings in this paper.

2.1 Ontology and Ontology Mapping

Based on the Karlsruhe Ontology Model [11], we adopt the following definition of ontology in our research.

Definition 1. An ontology is a structure

$$O = (C; T; R_C; R_T; I; V; \sigma_{RC}; \sigma_{RT}; \tau_C; \tau_T; \tau_{RC}; \tau_{RT})$$

consisting of a set of concepts C, a set of datatypes T, a set of relations R which include concept properties R_C and datatype properties R_T, a set of instances I, a set of data values V, the signature σ_{RC}: $R_C \rightarrow C\times C$, the signature σ_{RT}: $R_T \rightarrow C\times T$, the concept instantiation τ_C: $C \rightarrow \beta(I)$, the data value instantiation τ_T : $T \rightarrow \beta(V)$, the relation instantiation τ_{RC} : $R_C \rightarrow \beta(I^2)$, and the attribute instantiation τ_{RT} : $R_T \rightarrow \beta(I\times V)$.

In order to introduce the definition of relation measure, in this paper, we extend the definitions of ontology mapping in the existing literature [1, 7, 12, 13] as follows.

Definition 2. Given two ontologies O_1 and O_2, an ontology mapping is defined as a function, *map*, based on the correlative relation R_{map} and the entities of ontologies: E_1 and E_2, where $E_1 \in \{C_1, T_1, R_{C1}, R_{T1}, I_1, V_1\} \in O_1$, $E_2 \in \{C_2, T_2, R_{C2}, R_{T2}, I_2, V_2\} \in O_2$:

$$map: E_1 \xrightarrow{R_{map}} E_2 ,$$

$R_{map}(e_1, e_2)$ asserts that e_1 and e_2 can be mapped through the correlative relation R_{map}, $e_1 \in E_1$, $e_2 \in E_2$.

Definition 2 does not restrict the type of R_{map}. In our scenario, R_{map} not only includes equal relation, but also includes other relations, such as subclass relation. In practice, the type of R_{map} should be determined by the requirements of applications.

2.2 Relation Measure

In this paper, semantic mappings are established based on relation measures. Based on the definitions of similarity in references [8, 14], we first give the general definition of a relation measure, and then focus on two specific cases: the similarity measure and inclusion measure.

Definition 3. Given two ontologies O_1 and O_2, a relation measure is defined as a real-valued function as follows:

$$Relating: (E_1)\times(E_2) \rightarrow [0, 1]$$

where $E_1 \in \{C_1, T_1, R_{C1}, R_{T1}, I_1, V_1\} \in O_1$, $E_2 \in \{C_2, T_2, R_{C2}, R_{T2}, I_2, V_2\} \in O_2$, the value of *Relating*(e_1,e_2), $e_1 \in E_1$, $e_2 \in E_2$, indicates the probability of establishing mapping between e_1 and e_2 through a correlative relation R_{map}.

According to the type of correlative relation, relation measure can be classified into different types. In this paper, we pay more attention to similarity measure and inclusion measure whose corresponding correlative relations are equal relation and subclass relation, respectively.

Definition 4. Given two ontologies O_1 and O_2, a similarity measure is defined as a real-valued function:

$$Similarity: (E_1)\times(E_2) \rightarrow [0, 1]$$

where $E_1 \in \{C_1,T_1,R_{C1},R_{T1},I_1,V_1\} \in O_1$, $E_2 \in \{C_2,T_2,R_{C2},R_{T2},I_2,V_2\} \in O_2$, and E_1 and E_2 are of the same kind. The value of *Similarity*(e_1,e_2), $e_1 \in E_1$, $e_2 \in E_2$, indicates the probability of establishing mapping between e_1 and e_2 through the equal relation $Same_{map}$. $Same_{map}(e_1,e_2)$ asserts that e_1 and e_2 can be mapped through the equal relation $Same_{map}$, namely, e_1 equals to e_2.

The function of *Similarity* ought to be reflexive and symmetric [14], i.e.

$\forall x, y \in E$, it holds 1. *Similarity*$(x,x) = 1$ （reflexivity）

2. *Similarity*$(x,y) =$ *Similarity*(y,x) （symmetry）

Definition 5. Given two ontologies O_1 and O_2, an inclusion measure is defined as a real-valued function:

$$Sub：(E_1)\times(E_2)\rightarrow[0, 1]$$

where $E_1 \in \{C_1,T_1,R_{C1},R_{T1},I_1,V_1\} \in O_1$, $E_2 \in \{C_2,T_2,R_{C2},R_{T2},I_2,V_2\} \in O_2$, and E_1 and E_2 are of the same kind. The value of *Sub*(e_1,e_2), $e_1 \in E_1$, $e_2 \in E_2$, indicates the probability of establishing mapping between e_1 and e_2 through the subclass relation Sub_{map}. $Sub_{map}(e_1,e_2)$ asserts that e_1 and e_2 can be mapped through the subclass relation Sub_{map}, namely, e_1 is the subclass of e_2.

When computing relation measure between the entities of different ontologies, we should take into account the information on both syntax level and semantic level. The syntax level information is the syntax representation of these entities, such as the entities' name, and the semantic level information refers to the semantic relations between these entities, such as the affiliations between entities.

3 Formal Concept Analysis

FCA-Mapping is based on the Formal Concept Analysis (FCA) theory. In this section, we recall the basics of FCA theory as far as they are needed for this paper. A more extensive overview can be found in references [16, 17].

FCA was introduced as a mathematical theory modeling the concept of "concepts" in terms of lattice theory. To allow a mathematical description of extensions and intension, FCA starts with a formal context.

Definition 6. A formal context is a triple $K = (G, M, I)$, where G is a set whose elements are called object, M is a set whose elements are called attributes, and I is a binary relation between G and M (i.e. $I \subseteq G \times M$). $(g,m) \in I$ means "object g has attribute m".

A formal context can be represented by a cross table: a rectangular table with one row for each object and one column for each attribute, having a cross in the intersection of row g with column m if and only if $(g,m)\in I$. In a formal context, for any $A \subseteq G$, we let $A' = \{m \in M \mid \forall g \in A : (g,m) \in I\}$. Dually, for any $B \subseteq M$, we let $B' = \{g \in G \mid \forall m \in B : (g, m) \in I\}$. Now the definition of formal concept can be given as follows.

Definition 7. (A, B) is a formal concept of $K = (G,\ M,\ I)$ if and only if $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$. The set A is called the extent of the formal concept (A, B), and the set B is called its intent.

A formal context may have many formal concepts. the set of all formal concepts of $K = (G,\ M,\ I)$ is denoted by $\mathfrak{B}(K)$.

Definition 8. Let (A, B) and (C, D) be formal concepts of $K = (G,\ M,\ I)$. We say that (A, B) is a sub-concept of (C, D) (and, equivalently, that (C, D) is a super-concept of (A, B)), if and only if $A \subseteq C$ (equivalently, $D \subseteq B$). We use the mark " $\le$ " to express this relation and thus have

$$(A, B) \le (C, D) \Leftrightarrow A \subseteq C \Leftrightarrow D \subseteq B .$$

The set of all formal concepts of $K = (G,\ M,\ I)$, ordered by this relation, is denoted by $\underline{\mathfrak{B}}(K)$=($\mathfrak{B}(K)$, $\le$), and is called the concept lattice of the formal context $K = (G,\ M,\ I)$.

For a formal context, its concept lattice is unique. In this paper, we use existing algorithm [16] to generate concept lattice which is the basis of computing similarity measure and inclusion measure.

A possible confusion might arise from the different meanings of the word "concept" in FCA and in ontologies. In order to distinguish these two notions, we will always refer to the FCA concepts as "formal concepts". The concepts in ontologies are referred to just as "concepts" or "ontology concepts".

4 Method for Computing Similarity and Inclusion Measure

The primary goal of ontology mapping is to establish correlative relations between the entities of different ontologies. The characteristics of entities can indicate different correlative relations between entities. For example, if the properties of two concepts are equal, the concepts are also equal. In this statement, concepts can be viewed as entities, and properties can be viewed as characteristics.

FCA theory provides a method to identify the entities with the same characteristics, which can indicate equal relation $Same_{map}$ between these entities. Simultaneously, FCA also provides a method to order entities according to their characteristics, which can indicate subclass relation Sub_{map} between them. Here, entities correspond to the objects in definition 6, and characteristics correspond to the attributes in definition 6. Hence FCA theory can be used to compute the similarity measure and inclusion measure between entities of different ontologies.

In this paper, our method is based on FCA theory and includes four steps: (1) inputting information, (2) constructing formal context, (3) generating concept lattice, (4) computing similarity measure and inclusion measure.

4.1 Inputting Information

The input includes two ontologies O_1 and O_2, and the similarity matrix, *Matrix*, which can be computed by the method described in section 5.2 based on syntax level. The similarity matrix, *Matrix*, describes the similarities between the characteristics of E_1 and E_2, where $E_1 \in \{C_1, T_1, R_{C1}, R_{T1}, I_1, V_1\} \in O_1$, $E_2 \in \{C_2, T_2, R_{C2}, R_{T2}, I_2, V_2\} \in O_2$. The following example is given to explain the method used.

Example 1: Figure 1 depicts the ontologies O_1 and O_2. Nodes represent concepts which are denoted by the letter "*c*". The properties, denoted by the letter "*a*", are placed next to the concepts which they belong to. These properties belong to the *R* in definition 1. Arrow lines are used to indicate subclass relations between concepts, where the arrow heads point at the sub-concepts.

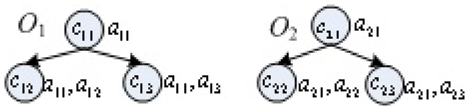

Fig. 1. Ontology O_1 and O_2

In this example, we have $E_1 = \{c_{11}, c_{12}, c_{13}\}$, $E_2 = \{c_{21}, c_{22}, c_{23}\}$, the characteristics of $E_1 = \{a_{11}, a_{12}, a_{13}\}$, and the characteristics of $E_2 = \{a_{21}, a_{22}, a_{23}\}$. The similarity matrix, *Matrix* of E_1 and E_2 is given in Table 2 as follows:

Table 2. Similarity matrix of E_1 and E_2

$O_1 \backslash O_2$	a_{21}	a_{22}	a_{23}
a_{11}	0.8	0.1	0.1
a_{12}	0.2	0.6	0.9
a_{13}	0	0.3	0.7

4.2 Constructing Formal Context

Different characteristics of entities can indicate different correlative relations between these entities, so we should construct different formal context for different entities and characteristics. The process of constructing formal context includes two steps:

1. Chose the set of entities that need to be mapped: $Entity1 \subseteq E_1$ and $Entity2 \subseteq E_2$, and the set of their characteristics A_1 and A_2 which will be analyzed, and then construct formal context.
2. Based on the similarity matrix of *Entity*1 and *Entity*2, *Matrix*, amend the formal context that has been constructed.

A more rigorous description goes as follows:

Step 1: Given two ontologies O_1 and O_2, we chose $Entity1 = \{e_{1i}, i \in \{1,...,n\}\}$, $Entity2 = \{e_{2i}, i \in \{1,...,m\}\}$, $A_1 = \{a_{1i}, i \in \{1,..., p\}\}$, and $A_2 = \{a_{2i}, i \in \{1,..., q\}\}$, where n, m, p and q are positive integers representing the number of entities or characteristics. I_1 is the set of relations between *Entity*1 and A_1; I_2 is the set of relations between *Entity*2 and A_2. Finally, the formal context that we built is $K = (G, M, I)$, where $G = Entity1 \cup Entity2$, $M = A_1 \cup A_2$ and $I = I_1 \cup I_2$.

For Example 1 (Step 1): Let $Entity1 = \{c_{11}, c_{12}, c_{13}\}$, $Entity2 = \{c_{21}, c_{22}, c_{23}\}$, $A_1 = \{a_{11}, a_{12}, a_{13}\}$ and $A_2 = \{a_{21}, a_{22}, a_{23}\}$, then the formal context is $K = (G, M, I)$, where $G = \{c_{11}, c_{12}, c_{13}, c_{21}, c_{22}, c_{23}\}$, $M = \{a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}\}$ and $I = \{(c_{11}, a_{11}), (c_{12}, a_{11}), (c_{12}, a_{12}), (c_{13}, a_{11}), (c_{13}, a_{13}), (c_{21}, a_{21}), (c_{22}, a_{21}), (c_{22}, a_{22}), (c_{23}, a_{21}), (c_{23}, a_{23})\}$. The cross table of this formal context is given in Table 3.

Table 3. Formal context

concepts\characteristics	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}
c_{11}	×					
c_{12}	×	×				
c_{13}	×		×			
c_{21}				×		
c_{22}				×	×	
c_{23}				×		×

Step 2: In order to decide the relations between *Entity*1 (or *Entity*2) and A_2 (or A_1), the formal context is amended based on the similarity matrix. The sub-steps of this process include:

1. For each i and j, with $i \in \{1,..., p\}$, $j \in \{1,...,q\}$, get the similarity between a_{1i} and a_{2j}: $Similarity(a_{1i}, a_{2j})$ from *Matrix*, where $a_{1i} \in A_1$ and $a_{2j} \in A_2$.
2. For each $a_{1i} \in A_1$, find the set of characteristics, $MaxSim\_a_{1i} \subseteq A_2$, whose elements have the highest similarity ($\neq 0$) with a_{1i}:

$$MaxSim\_a_{1i} = \{a_{2j^*} \mid j^* = \arg\max_{j\in\{1,...,q\}} \{Similarity(a_{1i}, a_{2j}) \neq 0\}\} \subseteq A_2 .$$

3. For each $a_{2j} \in A_2$, find the set of characteristics, $MaxSim\_a_{2j} \in A_1$, whose elements have the highest similarity ($\neq 0$) with a_{2j}:

$$MaxSim\_a_{2j} = \{a_{1i^*} \mid i^* = \arg\max_{i\in\{1,...,p\}} \{Similarity(a_{1i}, a_{2j}) \neq 0\}\} \subseteq A_1 .$$

4. In the formal context that has been built, let $P(e,a)=1$ if $(e,a)\in I$. $P(e,a)$ represents the probability that the relation (e,a) belongs to *I*. In the cross table, the mark "×" is replaced by value 1.
5. For each e_{1i} in *Entity*1 and $a_{2j}\in A_2$, we compute the probability that the relation (e_{1i}, a_{2j}) exists in the following manner. If $\exists x\in MaxSim\_a_{2j}$ satisfying $(e_{1i},x)\in I_1$, add the relation (e_{1i}, a_{2j}) into *I* and let $P(e_{1i}, a_{2j}) = Similarity(x, a_{2j})$.
6. For each e_{2i} in *Entity*2 and $a_{1j}\in A_1$, we compute the probability that the relation (e_{2i}, a_{1j}) exists in the following manner. If $\exists y\in MaxSim\_a_{1j}$ satisfying $(e_{2i},y)\in I_2$, add the relation (e_{2i}, a_{1j}) into *I* and let $P(e_{2i}, a_{1j}) = Similarity(y, a_{1j})$.

For Example 1 (Step 2):
1. The similarity matrix that we need has been given in Table 2.
2. The sets $MaxSim\_a_{1i}$: $MaxSim\_a_{11}=\{a_{21}\}$ with $Similarity(a_{11}, a_{21})$ = 0.8, $MaxSim\_a_{12}=\{a_{23}\}$ with $Similarity(a_{12}, a_{23})$ = 0.9, $MaxSim\_a_{13}=\{a_{23}\}$ with $Similarity(a_{13}, a_{23})$=0.7.
3. The sets $MaxSim\_a_{2j}$: $MaxSim\_a_{21}=\{a_{11}\}$ with $Similarity(a_{21}, a_{11})$ = 0.8, $MaxSim\_a_{22}=\{a_{12}\}$ with $Similarity(a_{22}, a_{12})$ = 0.6, $MaxSim\_a_{23}=\{a_{12}\}$ with $Similarity(a_{23}, a_{12})$=0.9.
4. The amended formal context is given in Table 4.

Table 4. The amended formal context

concepts\characteristics	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}
c_{11}	1			0.8		
c_{12}	1	1		0.8	0.6	0.9
c_{13}	1		1	0.8		
c_{21}	0.8			1		
c_{22}	0.8			1	1	
c_{23}	0.8	0.9	0.7	1		1

4.3 Generating Concept Lattice

In the amended formal context, we use the mark "×" to replace every real number, so the existing methods can still be used to generate concept lattice.

For Example 1 (Generating Concept Lattice): Figure 2 illustrates the concept lattice of the amended formal context. Each node represents a formal concept.

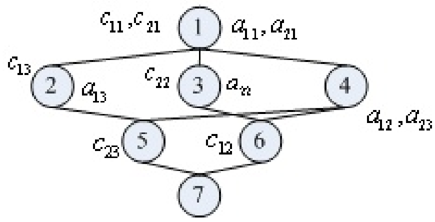

Fig. 2. Concept lattice

4.4 Computing Similarity Measure and Inclusion Measure

In this paper, we focus on the similarity measure and inclusion measure. Based on the concept lattice of the amended formal context, we can easily get the entities which have the same characteristics, namely, the entities grouped in the same nodes in the concept lattice. At the same time, we could also find the inclusion relations between the characteristics sets of the entities in *G*. The characteristics of the lower entities include the characteristics of the upper entities in the concept lattice. Based on the foregoing rationales we can compute the similarity measure and inclusion measure between the entities in *G*. The detailed steps are as follows:

1. For $\forall e \in G$, compute the probability that *e* belongs to the formal concept represented by the node which *e* is associated with in the foregoing concept lattice, denoted by *P*(*e*):

$$P(e) = \Pi_{i=1}^{n} P(e, a_i) \text{, where } a_i \in \{e\}' \text{ and } \{e\}' = \{a \in M \mid (e, a) \in I\} .$$

2. Given e_1 and e_2, where $e_1 \in Entity1$ and $e_2 \in Entity2$, if $\{e_1\}' = \{e_2\}'$, $Similarity(e_1, e_2) = P(e_1) \bullet P(e_2)$. This is based on the assumption that if entities have similar characteristics, they are also similar.
3. Given e_1 and e_2, where $e_1 \in Entity1$ and $e_2 \in Entity2$, if $\{e_1\}' \subset \{e_2\}'$ or $e_1' \supset e_2'$, the inclusion measure between e_1 and e_2 equals to $P(e_1) \bullet P(e_2)$.

The subclass relation between entities can be indicated by different types of characteristics. For example, if the properties of concept *a* include the properties of concept *b*, *a* is the subclass of *b*; if the instances of concept *a* include the instances of concept *b*, *b* is the subclass of *a*. So, when deciding the subclass relation between e_1 and e_2, we should consider the type of characteristics to use.

For Example 1: The probabilities that ontology concepts belong to the corresponding formal concepts are given in Table 5:

Table 5. Probabilities that ontology concepts belong to the corresponding formal concepts

Concepts	c_{11}	c_{12}	c_{13}	c_{21}	c_{22}	c_{23}
Probabilities	0.8	0.432	0.8	0.8	0.8	0.504

From figure2 we can see that the properties of c_{11} and c_{21} are equal, so *Similarity*(c_{11}, c_{21})= 0.8×0.8=0.64.

Based on the assumption that "if the properties of concept *a* include the properties of concept *b*, *a* is the subclass of *b*", we can get:

$Sub(c_{22}, c_{11}) = 0.64$, $Sub(c_{23}, c_{11}) = 0.4032$, $Sub(c_{23}, c_{13}) = 0.4032$,

$Sub(c_{21}, c_{12}) = 0.3456$, $Sub(c_{13}, c_{21}) = 0.64$, $Sub(c_{12}, c_{22}) = 0.3456$.

5 FCA-Mapping

Based on the method presented in section 4, we put forward a new method for ontology mapping: FCA-Mapping, which can be used to establish the equality and subclass mappings between ontologies. FCA-Mapping comprises five steps.

5.1 Preparation

Given two ontologies O_1 and O_2, we will analyze the requirements of the application in which these two ontologies will be used, and acquire domain knowledge and shared vocabulary. At the same time, we need to transform the initial representation of ontologies into a suitable format for the calculations of similarity measure and inclusion measure. For FCA-Mapping, the format can be different, but it should be consistent with the definition 1 in this paper.

5.2 Similarity Measure Based on Syntax Level

The similarity measure on syntax level is the basis of computing similarity measure and inclusion measure on semantic level. Many syntactic characteristics of entities can indicate certain relations between these entities, but these indications are not inevitable. So FCA-Mapping uses a multi-strategy approach: for each pair of entities, we compute multi-similarities based on different syntactic characteristics, and then merge them. In this paper, we refer to the following methods in literature to compute the similarities on syntax level.

1. String equality is a strict measure to compare strings [8]: Given entities e_1 and e_2, strictly compare the names of them:

$$Similarity_{syntax\_1}(e_1, e_2) = \begin{cases} 1, & name(e_1) = name(e_2) \\ 0, & \text{otherwise} \end{cases},$$

name(e) represents the name of e.

2. String similarity measures the similarity of two strings based on edit distance [8]: Given entities $e_1$and e_2,

$$Similarity_{syntax\_2}(e_1, e_2) = \max(0, \frac{\min(|e_1|, |e_2|) - ed(e_1, e_2)}{\min(|e_1|, |e_2|)}) \ .$$

3. Amending the equation in reference [5] to make it consistent with the definition 4 in this paper, we can get:

$$Similarity_{syntax\_2}(e_1, e_2) = \max(0, \frac{\min(|e_1|, |e_2|) - ed(e_1, e_2)}{\min(|e_1|, |e_2|)}) \ .$$

4. Based on domain knowledge and shared vocabulary, compute the similarity measure between entities e_1 and e_2 :

$$Similarity_{syntax4\_}(e_1, e_2) = \begin{cases} 1, & \text{if } e_1 \text{ and } e_2 \text{ is synonymous} \\ 0, & \text{otherwise} \end{cases} \ .$$

In this paper, we just list some existing methods. In practice, we can also choose other methods according to the application requirements. Finally, we get the uniform similarity measure on syntax level for each pair of entities:

$$Similarity_{syntax}(e_1, e_2) = \sum\nolimits_{i=1}^{n} w_i Similarity_{syntax\_i}(e_1, e_2) ,$$

where w_i is the weight for specific $Similarity_{syntax\_i}$ and $\sum\nolimits_{i=1}^{n} w_i = 1$, n is the number of similarity measures.

Here, we adopt the methods in reference [7] to choose w_i being the weight for specific $Similarity_{syntax\_i}$. Finally, we can obtain the similarity matrix, *Matrix*, for all entities of O_1 and O_2.

5.3 Similarity Measure and Inclusion Measure Based on Semantic Level

After obtaining *Matrix*, we can employ the method presented in section 4 to compute the similarity measure and inclusion measure on semantic level. In practice, we should consider different entities and their characteristics.

Table 6 lists some main entities, their corresponding characteristics, and the rules which describe the relations between them. In applications, different rules can be chosen in terms of special requirements.

Finally, given two entities: e_1 and e_2, we can obtain different similarity measures and inclusion measures based on different characteristics and rules:

1. $Similarity_{semantic\_i}(e_1, e_2)$, where $i \in \{1, ..., m\}$, m is the number of similarity measures.
2. $Sub_{semantic\_i}(e_1, e_2)$, where $i \in \{1, ..., p\}$, p is the number of inclusion measures which represent e_1 is a subclass of e_2 .
3. $Sub_{semantic\_i}(e_2, e_1)$, where $i \in \{1, ..., q\}$, q is the number of inclusion measure which represent e_2 is a subclass of e_1.

Table 6. Entities, characteristics and the rules which describe the relations between them.

Entity	No.	Characteristic	Rules
Concepts	R1	Properties	Given two concepts *a* and *b*, if the properties of *a* and *b* are equal, *a* and *b* are also equal. If the properties of *b* include the properties of *a*, *b* is the sub-concept of *a*.
	R2	Sub-concepts	Given two concepts *a* and *b*, if the sub-concepts of *a* and *b* are equal, *a* and *b* are also equal. If the sub-concepts of *b* include the sub-concepts of *a*, *a* is the sub-concept of *b*.
	R3	Sibling-concepts	Given two concepts *a* and *b*, if the sibling-concepts of *a* and *b* are equal, *a* and *b* are also equal.
	R4	Instances	Given two concepts *a* and *b*, if the instances of *a* and *b* are equal, *a* and *b* are also equal. If the instances of *b* include the instances of *a*, *a* is the sub-concept of *b*.
Properties	R5	Sub-properties	Given two properties *a* and *b*, if the sub-properties of *a* and *b* are equal, *a* and *b* are also equal. If the sub-properties of *b* include the sub-properties of *a*, *a* is the sub-property of *b*.
	R6	Parent-concepts	Given two properties *a* and *b*, if the parent-concepts of *a* and *b* are equal, *a* and *b* are also equal.
Instances	R7	Parent-concepts	Given two instances *a* and *b*, if the parent-concepts of *a* and *b* are equal, *a* and *b* are also equal.
	R8	Properties	Given two instances *a* and *b*, if the properties of *a* and *b* are equal, *a* and *b* are also equal.

5.4 Similarity Measure and Inclusion Measure Aggregation

After foregoing steps, given two entities e_1 and e_2, we may get many similarity measures and inclusion measures, and need to merge them. Finally, for each pair of entities, we get the following three values:

1. Similarity measure: $Similarity(e_1,e_2)=\sum_{i=1}^{n+m} w_i Similarity_i(e_1,e_2)$, where $\sum_{i=1}^{n+m} w_i=1$, n is the number of similarity measures on syntax level, and m is the number of similarity measures on semantic level.
2. Inclusion measure: $Sub(e_1,e_2)=\sum_{i=1}^{p} w_i Sub_{semantic\_i}(e_1,e_2)$, where $\sum_{i=1}^{p} w_i=1$, p is the number of inclusion measures, which considers e_1 as a subclass of e_2.
3. Inclusion measure: $Sub(e_2,e_1)=\sum_{i=1}^{q} w_i Sub_{semantic\_i}(e_2,e_1)$, where $\sum_{i=1}^{q} w_i=1$, q is the number of inclusion measures, which considers e_2 as a subclass of e_1.

The final relation measure for each pair of entities e_1 and e_2 is:

$$Relating(e_1,e_2)=\max\{Similarity(e_1,e_2),Sub(e_1,e_2),Sub(e_2,e_1)\}\ .$$

The final type of relation between e_1 and e_2 is determined by the measure with the maximal value.

5.5 Establishing Mappings

After aforementioned steps, we have obtained the relation measure for each pair of entities, and then we should decide how to establish mappings between these entities.

For e_1 and e_2, the corresponding correlative relation between them should be established when they satisfy:

$$Relating(e_1, e_2) = \max\{Relating(e_1, e_{2_j}) \mid \forall e_{2_j} \in E_2\}$$
$$Relating(e_1, e_2) = \max\{Relating(e_{1_i}, e_2) \mid \forall e_{1_i} \in E_1\}$$

6 Empirical Evaluation and Results

We have employed FCA-Mapping in the project "Shared Apparatus and Advanced Instruments System in Universities" supported by Ministry of Education, China. In this project, three pairs of ontologies were chosen as our experimental objects, which describe the database model of three apparatus websites: http://www.3gst.cn, http://www.cers.edu.cn and http://www.csts.net.cn.

Firstly, we analyze the requirements of this project and acquire shared vocabulary. Secondly, we compute the similarity measures on syntax level and get the similarity matrixes for each pair of ontologies based on the method presented in section 5.2. Thirdly, in the process of computing the similarity measure and inclusion measure on semantic level, we use the rules R1-R3, R5 and R6 listed in Table 6. Subsequently, we adopt the method introduced in section 5.4 to merge the similarity measures and inclusion measures that have been computed. Finally, we establish mappings according to the method introduced in section 5.5. In the whole process, we let w_i equal to mean value.

We use standard information retrieval metrics [8] to evaluate FCA-Mapping:

1. Recall: $r = \dfrac{\#correct\_found\_mappings}{\#possibel\_existing\_mappings}$;
2. Precision: $p = \dfrac{\#corrent\_found\_mappings}{\#all\_found\_mappings}$;
3. F-Measure: $f = 2pr/(p+r)$.

For the first pair of ontologies, the number of entities is 80, $r = 0.833$, $p = 0.909$, and $f = 0.869$; for the second pair of ontologies, the number of entities is 83, $r = 0.9$, $p = 0.9$, and $f = 0.9$; for the third pair of ontologies, the number of entities is 99, $r = 0.917$, $p = 0.846$, and $f = 0.880$.

Figure 3 depicts the results of experiments, from which we can see that FCA-Mapping achieved satisfactory results for our test ontologies. Compared with other methods, FCA-Mapping shows its strength in that it can automatically establish not only equal relations but also subclass relations.

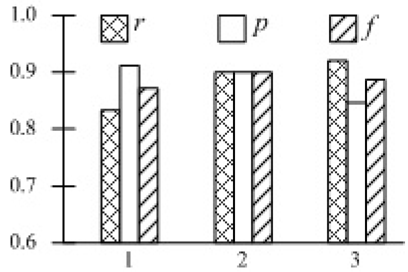

Fig. 3. Results of experiments

7 Conclusion

The ontology mapping problem arises in many scenarios. In this paper we have presented a novel method for ontology mapping: FCA-Mapping. Firstly, we analyzed the definition of ontology and ontology mapping, and put forward the notion of relation measure which is used to establish mappings between entities of different ontologies. In this paper, we focused on the similarity measure and inclusion measure which are two specific types of relation measures. Subsequently, based on FCA theory, we presented a method for computing the similarity measure and inclusion measure. Finally, we introduced FCA-Mapping in details, which can automatically establish not only equal relations but also subclass relations between entities of different ontologies. Experiments have shown that FCA-Mapping could achieve satisfying results on the test ontologies.

References

1. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping - the state of the art. The Knowledge Engineering Review, 18(1) (2003) 1-31
2. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic web. In: Proceedings of the 11th international conference on World Wide Web. ACM Press, New York (2002) 662-673
3. Pan, L.Y., Song, H., Ma, F.Y.: A Macrocommittees Method of Combining Multistrategy Classifiers for Heterogeneous Ontology Matching. In: Li, Q., Wang, G., Feng, L. (eds.): Advances inWeb-Age Information Management. WAIM 2004. Lecture Notes in Computer Science, Vol. 3129. Springer-Verlag, Berlin Heidelberg New York (2004) 672-677
4. Lacher, M.S., Groh, G.: Facilitating the exchange of explicit knowledge through ontology mappings. In: Proceedings of the 14th International FLAIRS Conference. AAAI Press, USA (2001) 305-309
5. Stoilos, G., Stamou, G., Kollias, S.: A String Metric for Ontology Alignment. In: Gil, Y., Motta, E., Benjamins, V.R. et al. (eds.): The Semantic Web – ISWC 2005. LNCS 3729. Springer-Verlag, Berlin Heidelberg New York (2005) 624-637
6. Kalfoglou, Y., Schorlemmer, M.: IF-Map: An Ontology-Mapping Method Based on Information- Flow Theory. Journal on Data Semantics I. LNCS 2800. (2003) 98-127

7. Ehrig, M., Sure, Y.: Ontology Mapping - An Integrated Approach. In: Bussle, C., Davies, J., Fensel, D. et al. (eds.): The Semantic Web: Research and Applications. ESWS 2004. LNCS 3053. Springer-Verlag, Berlin Heidelberg New York (2004) 76-91
8. Ehrig, M., Staab, S.: QOM - quick ontology mapping. In: Harmelen, F.V., McIlraith, S., Plexousakis, D. (eds.): The Semantic Web – ISWC 2004. LNCS 3298 . Springer-Verlag, Berlin Heidelberg New York (2004) 683-696
9. Euzenat, J., Valtchev, P.: Similarity-based ontology alignment in owl-lite. In: Proceedings of the European Conference on Artificial Intelligence. (2004) 333-337
10. Castano, S., Ferrara, A., Montanelli, S.: Matching Ontologies in Open Networked Systems: Techniques and Applications. In Spaccapietra, S., et al. eds. Journal on Data Semantics V, LNCS 3870. Springer-Verlag, Berlin Heidelberg New York (2006) 25-63
11. Stumme, G., Ehrig, M., Handschuh, S. et al.: The Karlsruhe view on ontologies. Technical report, German: University of Karlsruhe, Institute AIFB, (2003)
12. Su, X.: A text categorization perspective for ontology mapping. Technical report. University of Science and Technology, Norway (2002)
13. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: an algorithm and an implementation of semantic matching. In: proceedings of the European Semantic Web Symposium, LNCS 3053. Springer-Verlag, Berlin Heidelberg New York (2004) 61-75
14. Ehrig, M., Haase, P., Hefke, M., Stojanovic, N.: Similarity for ontologies - a comprehensive framework. In: Bartmann, D. et al. (eds.). Proceedings of the 13th European Conference on Information Systems, (2005)
15. Wille, R.: Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchie. In: Ganter, B., Stumme, G., Wille, R. (eds.): Formal Concept Analysis. LNAI 3626. Springer-Verlag, Berlin Heidelberg New York (2005) 1–33
16. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag, New York (1999)
17. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Fast computation of concept lattices using data mining techniques. In: Proceedings of 7th International Workshop on Knowledge Representation Meets Databases. Berlin, Germany (2000) 129-139.