

Mining Implied Semantics of Database

Zhenxin Qu

Information and Safety Engineering School, Zhongnan University of Economics and Law, Wuhan,
430073, China

quzx72@gmail.com

Keywords: mine, semantic, database

Abstract. To mine richer semantics from relational database data, method of mining was discussed. Aiming at specific type of semantics, thirteen rules were proposed. Applying these rules some implied semantics was found naturally. The research shows that these rules have good maneuverability and high efficiency.

Introduction

To solve the problem, Information Island, semantic technology is used to realize interoperation between information systems[1], which helps to reach consistent data interpretation between information systems. During the course, relational database data will be transformed into semantic data. Whether the interoperation is successful depends largely on the quality of transformation. Some technologies have realized automatic transformation, which are on the basis of database schema. But most semantics of relational database data are implied in application programs, database schema cannot embody them. They are implied semantics. So how to mine implied semantics of relational database data cannot depend on existing automatic transformation technologies.

Traditional methods of building ontology are scientific methods, not engineering methods, which have poor maneuverability. With these methods, quality of ontology depends highly on ability and experience of knowledge engineer. To mine implied semantics of relational database data, a set of mining rules are proposed in this paper. The method has high maneuverability and helps to mine semantics with high quality.

General Methods and Principles of Building Ontology

How to build ontology? There is no general method widely accepted. Some methods are often referenced[2]. For example, Stanford University puts forward seven steps of building ontology; METHONLOGY method has initially been proposed to design chemical ontology, later, it's applied into other domain, which has three stages. They are only rough steps or stages, not detail methods.

Gruber[3] proposes five rules of building ontology which are widely accepted. (1) Clear. Definition should be objective, not affected by background. (2) Consistency. Conclusion from existing knowledge should not be contradicted with existing knowledge. (3) Scalability. Basic concept should be provided for foreseeable new task. If there is new concept, it can be defined on the basis of existing concept. (4) Minimum code preference. Code should not depend on some single representation method. (5) Minimum ontology commitment. Constraints on axiom should be the weakest, basic vocabularies should be only necessary.

In fact, these five rules are contradicted. We can only place emphasis on part rules during building ontology.

Ontology Learning

The work building ontology is complex, exhausting and low efficiency, although some tool software can ease the workload. To solve the problem, some algorithms are designed to generate ontology automatically or semi-automatically based on existing data. These technologies are called ontology learning. According to data which have different structure degree, ontology learning methods are classified into three categories.

(1) Generating ontology from unstructured data.

Because unstructured data have no fixed structure, ontology learning is hard relatively. Free text is typical unstructured data, and is also important data source of ontology learning. There are three learning methods based on free text, concept learning, relationship recognition and axiom generating.

Concept learning is mainly based on linguistics and statistics.

More methods can realize relationship recognition, such as method based on dictionary, method based on concept clustering, method based on association rule, method based on model driving, formal concept analysis, machine Bayesian classification, decision tree learning, etc.

Axiom generating has fewer researchers. Hathi system can extract axiom from free text automatically. It analyzes structure of sentences and apply predefined model to extract axiom matched the model.

(2) Generating ontology from semi-structured data.

Semi-structured data have certain structure, but not strict, such as HTML document, XML document and knowledge base. Typical technologies are as follow.

HTML document, XML document, RDF document are semi-structured data, invisible structure hidden in documents can be found and transformed automatically.

Some systems, such as Litkowski, Jannink, Rigau, Hearst, generate ontology from machine readable dictionary.

Suryanto and his colleagues have designed algorithm to extract ontology from existing knowledge base.

(3) Generating ontology from structured data.

Structured data have fixed structure; can be described with unified data model. Relational database data are biggest structured data now. So how to generate ontology from relational database data play a major role in the study[4]. There are two main methods: static method and dynamic method. With static method, database schemas are transformed into concepts and properties of ontology, database records are transformed into ontology instances once and for all. With dynamic method, database schemas are mapped into concepts and properties of ontology, while database records have not been transformed; according to every semantic query, SQL statements are generated and executed, matched records are encapsulated into ontology.

Mining Implied Semantics

If we build ontology observing methods and principles mentioned above, we will find there aren't specific steps, only has spirit. Every designer does work according to his own understanding. If we use existing automatic learning technology, only superficial semantics of data can be found.

To mine more semantics from given relational database, four kinds of mining rules are putted forward.

(1) Concept subdivision

Rule1: Assume that relation R corresponds to ontology concept C, field f of R corresponds to property p.

If value of field f might be empty, then next can be concluded:

NewC rdfs:type owl:class

p rdfs:domain NewC

With Rule1, records of relation R, which have non-empty value on field f, are defined as a new concept NewC. NewC is subdivision of concept C, which only corresponds to part records of relation R.

Rule2: Assume that ontology concept C has property p.

If constraint hasValue is applied on property p, a subdivision of concept C can be gotten.

(2) Mining relation between concepts

Rule3: Assume that C1 and C2 are ontology concepts.

If semantics of C1 and C2 are similar and difference between them is not specifically, then next can be concluded:

C rdfs:type owl:class

C1 subClassOf C and C2 subClassOf C

Rule4: Assume that C1 and C2 are ontology concepts.

If semantics of C1 and C2 are similar and C2 covers C1, then next can be concluded:

C1 subClassOf C2

Rule5: Assume that C1 and C2 are ontology concepts.

If C1 and C2 are different terms, but describe the same thing, then next can be concluded:

C1 equivalentClass C2

Rule6: Assume that C1 and C2 are ontology concepts.

If C1 and C2 have semantic intersect, then next can be concluded:

C = C1 Intersection C2

Rule7: Assume that C1 and C2 are ontology concepts.

If a concept is needed to contain C1 and C2, then next can be concluded:

C = C1 Union C2

(3) Mining relation between properties

Rule8: Assume that P1 and P2 are ontology properties.

If P1 and P2 denote exactly the same property, then next can be concluded:

P1 equivalentProperty P2

Rule9: Assume that P1 and P2 are ontology properties.

If C1 and C2 denote similar property and difference between them is not specifically, then next can be concluded:

P rdfs:type owl:ObjectProperty

or P rdfs:type owl:DatatypeProperty

P1 subPropertyOf p and P2 subPropertyOf p

Rule10: Assume that P1 and P2 are ontology properties.

If C1 and C2 denote similar property and C2 covers C1, then next can be concluded:

P1 subPropertyOf P2

Rule11: Assume that P1 and P2 are ontology properties.

If P1 and P2 denote inverse property, $\text{Domain}(P1)=\text{Range}(P2)$, $\text{Domain}(P2)=\text{Range}(P1)$, then next can be concluded:

P1 inverseOf P2

(4) Mining IF/THEN relation

Rule12: Assume that C1 and C2 are ontology concepts.

If exists “if obj is member of C1, then obj is member of C2”, then next can be concluded:

C1 subClassOf C2

Rule13: Assume that C1, C2 and C3 are ontology concepts.

If exists “if obj is member of C1 and obj is member of C2, then obj is member of C3”, then next can be concluded:

C3 subClassOf C1 and C3 subClassOf C2

Rule is not supported in OWL. So simple IF/THEN relation can be expressed by Rule12, Rule13, domain and range.

There is a relation table, describing population, listed in Table1.

Table 1. Population table

| name | id | gender | birthday | spouse |
|-------|------|--------|----------|--------|
| Jack | 0001 | Male | 1980/1/1 | |
| Peter | 0002 | Male | 1970/8/1 | Aileen |
| Jane | 0003 | Female | 1983/6/3 | Bob |
| Fox | 0004 | Male | 1977/8/9 | |
| White | 0005 | Male | 1982/8/3 | |

The relation table is mapped to concept PERSON, which has properties: name, id, gender, birthday and spouse. We can use rules mentioned above to mine more implied semantics.

(1) To get a subdivision of concept PERSON, which denotes married person, we can define it like this:

```
:MarriedPerson rdfs:subClassOf :PERSON
```

```
:spouse rdfs:domain :MarriedPerson
```

(2) To get a subdivision of concept PERSON, which denotes woman, we can define it like this:

```
<owl:Class rdf:ID="Woman">
```

```
<rdfs:subClassOf>
```

```
<owl:Restriction>
```

```
<owl:onProperty rdf:resource="#gender"/>
```

```
<owl:hasValue rdf:datatype = "http://www.w3.org/2001/XMLSchema#string">
```

```
Female
```

```
</owl:hasValue>
```

```
</owl:Restriction>
```

```
</rdfs:subClassOf>
```

```
</owl:Class>
```

(3) To express the rule that woman is person, we can define it like this:

```
:Woman rdfs:subClassOf :person
```

(4) To express the rule, if somebody is female and married, then she is MarriedWoman, we can define it like this:

```
:MarriedWoman rdfs:type owl:class
```

```
:MarriedWoman rdfs:subClassOf :Woman
```

```
:MarriedWoman rdfs:subClassOf :MarriedPerson
```

Summary

With existing automatic ontology learning system, only superficial semantics of database can be found. When try to mine semantics of database manually, there is no operable methods. Thirteen rules are proposed, which aim at mining specific type of semantics. With these rules, more implied semantics of database can be found.

REFERENCES

- [1] Zhenxin Qu, Shengqun Tang. Deep Semantic Integration for Information System. CINC2009,2009.6. p292-295
- [2] Zhenxin Qu. Semantic web and interoperation of e-government systems. Wuhan: Hubei science and technology press, 2010
- [3] T. R. Gruber, "A translation approach to portable ontologies", Knowledge Acquisition, 5(2):199-220, 1993.
- [4] Zhenxin Qu, Shengqun Tang. Research on Transforming Relational Database into Enriched Ontology. ICACTE2008, 2008.12. p749-753