

## **GENERATING ONTOLOGY BASED ON FORENSIC INQUEST DOCUMENTS STRUCTURED ON THE WEB**

**SONGIL CHA<sup>1</sup>, KYONGJIN SOK<sup>2</sup>, CHANGHO KIM<sup>3</sup>,  
KINAM SIN<sup>4</sup> and UNSOK RYU<sup>5</sup>**

<sup>1</sup>College of Information Science

University of Sciences

Pyongyang

Democratic People's Republic of Korea

<sup>2</sup>Institute of Information Technology

University of Sciences

Pyongyang

Democratic People's Republic of Korea

<sup>3</sup>Department of Information Technology

Kim Hyong Jik University

Pyongyang

Democratic People's Republic of Korea

<sup>4</sup>Faculty of Mathematics

Kim Il Sung University

Pyongyang

Democratic People's Republic of Korea

e-mail: [15bf12025@hit.edu.cn](mailto:15bf12025@hit.edu.cn)

---

2010 Mathematics Subject Classification: 68T30, 68Wxx.

Keywords and phrases: forensic medicine ontology, inquest report, Web tables, hierarchical relationship, class-instance.

Received May 20, 2019

<sup>5</sup>School of Information Science  
Kim Il Sung University  
Pyongyang  
Democratic People's Republic of Korea

### **Abstract**

Ontology has become a fundamental and critical component for developing different applications on the Web. With the phenomenal growth of the Web resources, to generate automatically ontologies by using existing documents on the Web has gotten more and more attention. Forensic medicine ontology is used to narrate and reason the knowledge of a forensic domain. Previous studies for constructing forensic medicine ontologies have not automatically constructed by using all the semantic features of the Web documents. Hereby, it is difficult to rapidly construct an instance and property elements of ontology from the Web documents that are increasing daily. This paper focuses on the ontology generation method from forensic inquest documents structured on the Web. In order to generate most ontology elements for reasoning, we discuss how to generate hierarchical relationships, non-hierarchical relationships, and property elements based on structural and semantic characteristics of forensic inquest report. An empirical investigation of our method has proved the effectiveness of forensic medicine ontology generation. Experimental results show that our ontology generation method can generate with high quality rapidly and correctly important instances and properties of a forensic domain.

### **1. Introduction**

Recently, the Semantic Web comprises techniques that promise to dramatically improve the current WWW (World Wide Web) and its use (Antoniou and van Harmelen [1]; Obrst et al. [18]). With the emergence of the Semantic Web and the growing number of heterogeneous data sources, the benefits of ontologies are becoming widely accepted. Ontology has become an advanced technology in artificial intelligence and knowledge engineering, playing an important role in knowledge representation, knowledge acquisition and ontology application (Guarino et al. [12]; Serrano [23]; Dietze et al. [8]; Baclawski et al. [2]; and Goy et al. [11]). As the base of the ontology applications, the ontology creation

has become a research hotspot. The Web ontologies define terms used as data (metadata) for explaining things of a special domain. Nowadays, researchers are paying attention to the automatic transformation of the Web resources in the areas into ontologies (Spanos et al. [25]; Zhang et al. [31]; Cha and Han [6]). In order to provide the necessary means to widely apply ontologies to various fields, there are today many proposals for using ontology learning and machine learning, and until now the study on domain ontology learning has been flourishing.

Rupasingha et al. [22] proposed a Web service clustering method by calculating the semantic similarity of Web services by using the ontology learning method. El Asikri et al. [10] described the commonalities of the areas, such as the semantic web and data mining, in order to resolve the problem of extracting useful and shared knowledge, as well as solve the problem of the interoperability between Web systems by using the ontology learning from Web content. Rupasingha et al. [21] presented a method for calculating Web service similarity using both ontology learning and machine learning that uses a support vector machine for similarity calculation in generated ontology instead of an edge count base method. Kumara et al. [17] proposed clustering approach that considers the complex data type as well as the simple type in measuring the service similarity. This approach used a hybrid term similarity method which proposed in their previous work to measure the similarity. Song et al. [24] reviewed the related concepts and methods of ontology construction and extension, proposed an automatic ontology extension method based on supervised learning and text clustering. This method used the K-means clustering algorithm to separate the domain knowledge and to guide the creation of a training set for Naive Bayes classifier.

Jupp et al. [16] presented Webulous that is an application suite for supporting ontology creation by design patterns and provided simple mechanisms for the addition of new content in order to reduce the overall cost and effort required to develop ontologies. Peng et al. [19] proposed a

method which can learn a heavy-weighted medical ontology based on medical glossaries and Web resources, in order to deal with heterogeneous knowledge in the medical field. Wei et al. [29] presented a semi-automatic construction method for agricultural professional ontology from Web resources. For semi-structured web pages, this method automatically extracted and stored structured data through a program, built pattern mapping between relational database and ontology through human-computer interaction, and automatically generated a preliminary ontology, finally completed checking and refining by domain experts.

The Web is an enormous resource of information contained in billions of individual pages. Most information resource on the Web is presented in the form of semi-structured or unstructured documents, encoded as a mixture of loosely structured natural language text and template units. Yu et al. [30] proposed a modified hierarchical concepts tree building method by applying a pruning algorithm to the graph. They used the clue words to product queries containing hierarchical relation to get corpus rich in concepts hierarchical relation through the search engine from Web. Vasilateanu et al. [28] proposed a semantic search engine for relevant documents in an enterprise, based on automatically generated domain ontologies, with observing on the component for ontology learning and population. Dixit et al. [9] focused on generating domain specific ontology for retrieving hidden web contents. In this paper, a knowledge base used in automatically filling up search interfaces for retrieving hidden web data.

Web tables are used mainly for structuring information, and they are important means to represent structured information. Table structures represent relations between data in the table. Therefore, we can easily extract ontology from a table based on the features of table structures without the use of syntax analysis. However, understanding of table contents requires table structure comprehension and semantic interpretation, which exceed the complexity of corresponding linguistic

tasks. Previous studies for extracting information from Web tables are centralized to interpret table structure: Pivk et al. [20] focused on understanding table-like structures only due to their structural dimension, and the table model consists of physical, structural, functional, and semantic components. The authors transformed the most relevant table types into F-logic frames, and also demonstrated and evaluated the successful generation of frames from HTML (Hyper Text Markup Language) tables. Jung et al. [14] suggested a method for extracting table-schemata based on table structure and heuristics. Using this method, a table is converted into a table-schema and a triple. Moreover, since it is important to determine whether or not a table has meaningfulness that is related to the structural information provided at the level of the table head, the authors further investigated the types of tables, established the features that distinguished meaningful tables from others, built a training data set using these features and constructed a classification model by using a decision tree. However, this method cannot handle a table without head and also cannot extract an appropriate head by using background colour and font. Further, Chavan and Shirgave [7] introduced a method for determining the meaningfulness of a table and extracting the head contents from the meaningful table. The authors applied table mining to general HTML documents, separated meaningful tables from decorative tables, and extracted the information using the head. Guo et al. [13] presented a synthetic Web table extraction method based on the ontology and structure. This approach firstly locates the tables based on heuristic rules, and then analysis the table structure according to the label and the title ontology, at last extract and save the table data on the basis of the obtained characteristics. Tao and Embley [27] offered a solution for the common special case in which so-called sibling pages are available, show that the system can successfully identify sibling tables, generate structure patterns, interpret tables using the generated patterns, and automatically adjust the structure patterns as it processes a sequence of hidden-web pages.

As specified above, it can be found that the work mentioned above mainly focused on how to extract information from the tables, instead of investigating how to extract ontologies from the Web tables, and the detailed ontology extraction rules are not provided. In addition, most of the research endeavours to interpret the table by using the structural characteristics of the table. But most Web tables are designed by humans, thus, it has a certain limit to automatically interpret tables using only structural information of the table. Jung et al. [15] detected that, generally, a table provides a semantic core element in a head and proposed a method for obtaining class-instance relationship and triple using heuristics for extracting table schemata based on semantic core element. Although the authors in Jung et al. [15] proposed heuristics for detecting semantic characteristics based on the location of table cells, they did not mention which becomes a semantic core element.

Through the experimental observation on semantic features of documents structured on the Web, we found that if there are proper nouns on the table, then they can become a semantic core element. The previous research (Cha et al. [5]) proposed algorithms to automatically generate total instances or properties belonging to ontology class from a Web table, taking several instances or properties belonging to the class composed of a proper noun as the seed. Also, the previous research (Cha et al. [4]) proposed a method for classifying the Web tables and extracting several relationships. The work focused on interpreting the property range as string only, and thus it can't extract many property elements. Based on the observation above, the existing work for extracting table information did not discuss how to extract the sufficient property elements used for ontology reasoning, such as property with the hierarchical relationship and class-instance relationship, properties for restricting properties, and other property types.

To this end, this paper focuses on the extracting property elements based on interpreting table contents by using structural and semantic characteristics of the table. That is, in this paper, we propose a method to generate ontologies with sufficient property that is used for ontology reasoning. The paper is structured as follows. In Section 2, we present a generation method of the hierarchical relationships and the property elements corresponding to triples extracted from Web tables. In Section 3, we describe a concrete method for generating the non-hierarchical relationship. In Section 4, we evaluate our method according to the experimental result. In Section 5, we provide a conclusion to our work.

## **2. Generating Hierarchical Relationships and Property Elements from the Web Tables**

In this section, we describe automatically generating techniques of OWL (Web Ontology Language) ontology through generating of hierarchical relationship and property elements from the Web tables. The generated hierarchical relationships and property elements include is-a relationship, class-instance relationship, super-sub relationship (`rdfs:SubPropertyof`), triple (`Owl:ObjectProperty/Owl:DatatypeProperty`) and properties for restricting properties (`rdfs:domain`, `rdfs:range`) (Bechhofer et al. [3]).

### **2.1. Generating hierarchical relationships**

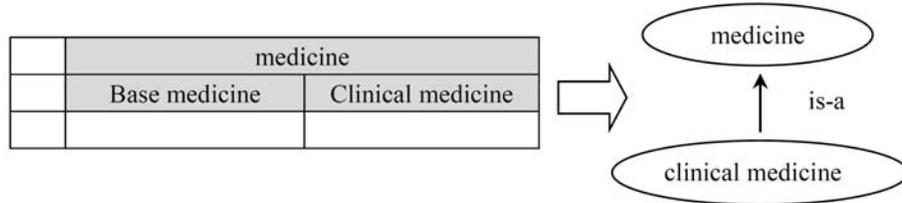
If the head of the table consists of a simple row (or column), we can extract a class from a table head. However, to classify data cells, the head of the table can also have hierarchical categories (Cha et al. [4]). In the case of this, several classes with a hierarchical relationship can be extracted. At this point, appears a lot of case which shows that relationship between SuperCategory and low category isn't is-a relationship that is considered from the perspective of property inheritance. For example, an event-human relationship such as operation-outstanding becomes a hierarchical relationship and also there

are class-instance relationships. For the purpose of extracting a hierarchy of categories with is-a relationship, this subsection presents string matching approach on the hierarchy of categories and approach using structural characteristics of tables, that is a method to identifying the semantics in head strings.

### **2.1.1. String matching on the hierarchy of categories**

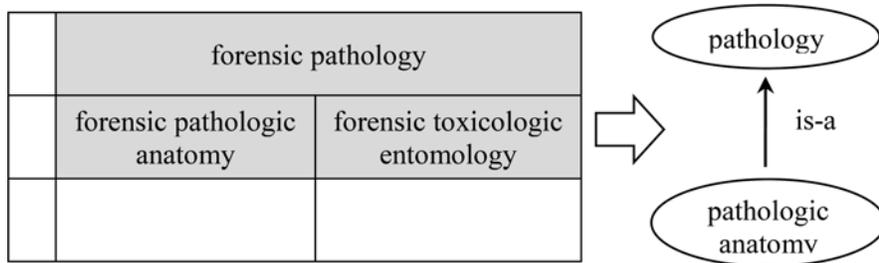
SubCategories, which are lower than certain category, are to subdivide concept. In many cases, they consist of a compound word that contains SuperCategory name. For example, we can obtain hierarchies such as ‘medicine - forensic medicine’ and ‘investigation - investigation report’ in Web documents of forensic medicine domain. The former is a is-a relationship that is considered from the perspective of property inheritance, the rear isn’t appropriate for is-a relationship. In order to extract the is-a relationship, in this subsection, we discuss the later string-matching method and the elimination method of the first-string accordance part on the compound word of the category hierarchy.

The later string matching is the approach extracting SubClass which SubCategory name is an ‘any string + SuperCategory name’ through the comparison between SuperCategory name and SubCategory name that have framed the hierarchy of categories. Take Figure 1 as an example. Herein, we can extract relationship such as ‘clinical medicine’ is-a ‘medicine’ from SuperCategory-SubCategory such as ‘medicine’-‘clinical medicine’ by the later string matching. This approach can be applied to 1-generation detachment SuperCategory-SubCategory connection and can also be applied to N-generation detachment category connection. For example, we can extract the connection such as ‘forensic medicine’ is-a ‘medicine’ from ancestor-descendant connection such as ‘medicine’-‘social medicine’-‘forensic medicine’.



**Figure 1.** Example of later string matching.

The elimination of the first string accordance part is the approach to remove a part which corresponds with SuperCategory name from the front of SubCategory name. Take Figure 2 as an example. Herein, we can extract relationship such as ‘pathologic anatomy’ is-a ‘pathology’ by removing the first-string accordance part such as ‘forensic’. This method has a benefit to extract is-a relationship which isn’t depended on string duplicate. This approach can also be applied to N-generation detachment category connection:

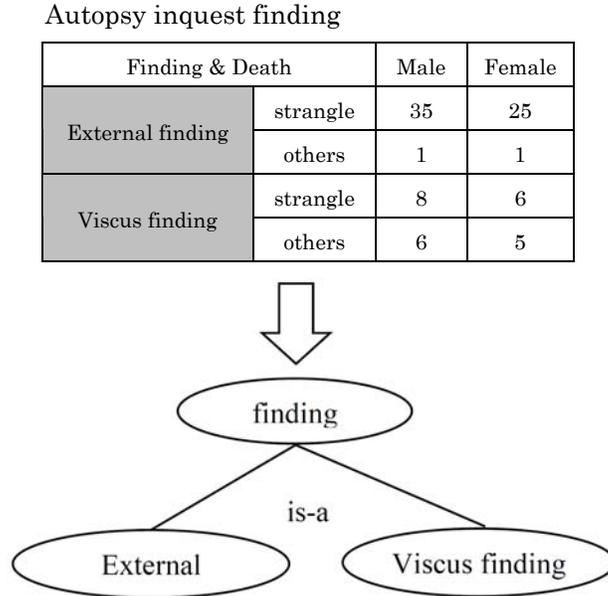


**Figure 2.** Elimination of first-string accordance part.

**2.1.2. Method using structural characteristics of table**

In the table structure with partitioned data regions and the complex table structure, the spanned cell is used for representing the semantic hierarchical structure of head elements (Cha et al. [4]). In the case of the spanned cell being used for abstracting data cells, the spanned cell is called Semantic-Relation Label (hereafter SRL). If pay attention to SRL, we can extract the is-a relationship. Take Figure 3 as an example. Herein

shows an approach for extracting is-a relationship based on the spanned cell (SRL). In Figure 3, the data region is partitioned as two regions by SRL such as ‘External finding’ and ‘Viscus finding’, we can extract is-a hierarchical relationship which is framed by two SubClass.



**Figure 3.** Using structural characteristics of the table.

## 2.2. Generating class-instance relationships

As we mentioned above, generally, we can construct a class from a table, but an instance belonging to the class is determined according to the head location in the table.

Through an analysis of the table structure, we may obtain the following fact: the table head is only located in the row or the column of the table in otherwise cases except the 2-dimensional table. If the table head is located in the row, each row of the table’s data region is considered as an instance. Likewise, if the table head is located in the column, each column of the table’s data region is also considered as an instance. However, in the case of a 2-dimensional table, the table head is located in the uppermost row and left-most column. Therefore, if we

simply consider the row or the column as an instance, it is difficult to extract means from the instance of the class. Accordingly, we construct class-instance relationship (`rdf:type`) after conversion of the table structure. Figure 4 shows the process of constructing class-instance relationship after conversion of the 2-dimensional table.

In order to identify each instance belonging to the class, we append a row identifier column to the table and interpret the identifier as the instance. In Figure 4, we interpret 'Allocation of Nurse' as a class, identifier '#1' as an instance of this class.

### **2.3. Generating triples**

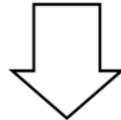
In this subsection, we derive a triple from Web tables and determine a property name. Deriving a triple from the Web tables consists of the following three parts:

- In accordance with structural and semantic characteristics of the table need to determine each element constructing a triple from the table.
- After confirming elements of the triple, need to determine clearly a property name.
- Need to determine whether the property is object property or data type property (`Owl:ObjectProperty/Owl:DatatypeProperty`).

Allocation of Nurse

	Room	Mon	Tue	Wed	Thu	Fri
Morning	1	Lee	Lin	Yuo	Tasi	Lai
	2	Jiang	Lai	Tasi		Lin
Afternoon	1	Lee	Wang	Huang	Yuo	Yuo
	2		Lai	Yuo	Jiang	Tasi

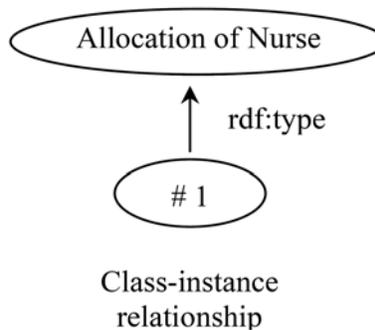
(a) Original



Allocation of Nurse

Date	Time	Room	Nurse
Mon	Morning	1	Lee
Mon	Morning	2	Jiang
Mon	Afternoon	1	Lee
Mon	Afternoon	2	
Tue	Morning	1	Lin

(b) Result

**Figure 4.** Examples of table conversion.

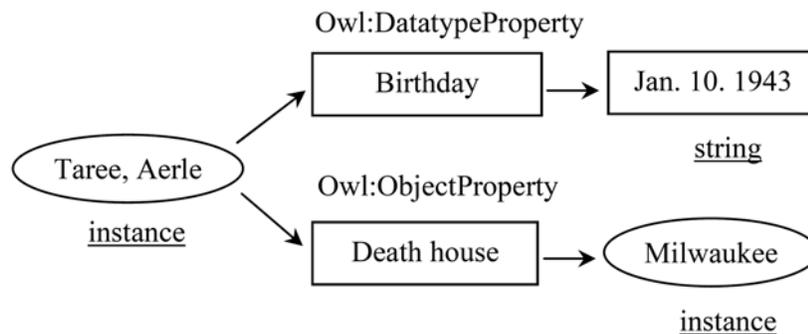
Deriving a triple from the Web tables is classified into two groups based on the head location of Web tables and the structure of its inner elements: firstly, in the case of the first group, we take a set of three elements ‘instance-property name-property value’, in which are mapped the table title as an instance, the head item of the table as a property, and the body element corresponding to the head item as a property value, as a triple. Figure 5 shows an example of triple extraction from the Web table belonging to the case history sheet. Then, in the case of second group (in this case, in the head of the table there is a semantic core

element), we take a set of three elements ‘instance-property name-property value’, in which are mapped an element of column (row) belonging to the semantic core element as an instance, other head item of the table as a property, and the body element corresponding to the head item and the instance as a property value, as a triple. Figure 6 shows an example of triple extraction from the table belonging to the second group. In the case of the triple extraction, we can define the property as ‘Owl:ObjectProperty’ or ‘Owl:DatatypeProperty’. If the property is ‘Owl:ObjectProperty’, then the property value is an instance, and if the property is ‘Owl:DatatypeProperty’, then the property value is a string. In general, if the body element corresponding to the property contains number, then the property value is a string. Take property values such as ‘Jan. 10, 1943’ of Figure 5, ‘19’ and ‘FUT3’ of Figure 6 etc. as an example.

Spot field investigation report

Name	Aerle Taree
Injury at	Penetrating injury, Stab wounds
Birthday	Jan. 10, 1943
Death house	Milwaukee
Death date	Aug. 13, 2016
Cause of death	Overdose of drugs
More	Wikipedia-Aerle Taree

(a)



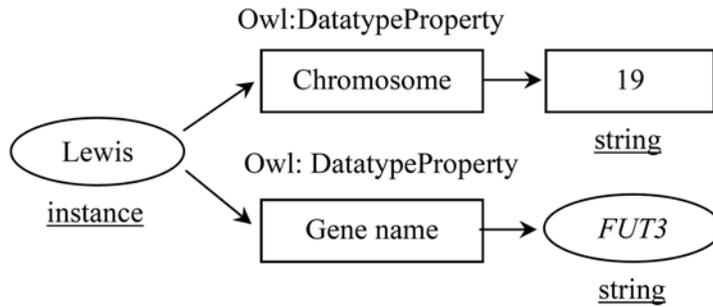
(b)

**Figure 5.** Triples generated from the table of the first group.

Erythrocyte blood group system

System	Antigen number	Chromosome	Gene name
ABO	4	9	<i>ABO</i>
P	1	22	<i>P1</i>
Lutheran	18	19	<i>LU</i>
Kell	22	7	<i>KEL</i>
Lewis	3	19	<i>FUT3</i>
Duffy	36	1	<i>FY</i>
Kidd	7	18	<i>JK</i>
Diego	2	17	<i>AE1</i>

(a)



(b)

**Figure 6.** Triples generated from the table of the second group.

#### 2.4. Properties for restricting properties

Firstly, we extract the property domain (rdfs: domain) for each property that is extracted by Subsection 2.3. The property domain is defined as a class which instance (i.e., the subject of the triple) belongs to.

We extract the property domain from the class-instance relationship that is obtained already by the method described in Subsection 2.2. That is, we found the instance that is matched with the subject of a triple from the class-instance relationship extracted already and take the class which the instance belongs to as a property domain. In Figure 6(a),

'Chinese cities' is a semantic core element. Accordingly, we interpret that 'Chinese cities' is a class and each element of the second column such as 'Shanghai', 'Beijing' etc. is an instance belonging to the class. In this case, the domain of the property 'Total population' is 'Chinese cities'.

Then, we extract property range (rdfs: range) for each property that is extracted by Subsection 2.3. Contrary to the extraction of property domain, it is difficult to extract property range for entire properties. We extract a range of property from a class-instance relationship and is-a relationship that is obtained already. That is, we found the instance that is matched with the object of a triple from the class-instance relationship extracted already and take the class which the instance belongs to as a property range. In addition, in order to extract property range that isn't extracted by using above method, we found a class that is matched with the object of a triple from the is-a relationship extracted already and take a super class which the class belongs to as a property range. In Figure 5(a), the range of property 'Place of birth' is a class 'City of Wisconsin' that is class of the class-instance relationship extracted already.

In case that a class obtained as domain or range of each property matches with the already extracted class of is-a relationship, we also extract super notion of this is-a relationship as domain or range.

## **2.5. Generating super-sub relationship**

In this subsection, we derive a super-sub relationship (rdfs: SubPropertyOf) for properties of the triples that are derived by Subsection 2.3. In Subsection 2.1, we described a method for extracting is-a relationship that is a hierarchical relationship between classes. The same can be done for properties. In general, P2 is a sub property of P1 if P1 exists whenever there is P2.

In order to derive super-sub relationship between properties, firstly, we take property values corresponding to an instance that is subject of a triple. Then, in case there is an inconsistent property name in the triples corresponding to this property values, we extract these property names as the candidate of super-sub relationship. We take the property in which there is a small number of property values as sub property candidate for two different property names. Finally, we decide these properties as a super-sub relationship if two properties contain the same domain and the same range. Figure 7 shows an example of a super-sub relationship extracted from a set of triples. Finally, Algorithm 1 shows a procedure for generating properties with the super-sub relationship.

**Algorithm 1.** Super-subRelation(TS): derives all properties with the super-sub relationship from the set of triples

**Input:** TS: table of relational DB to extract properties with the super-sub relationship

**Output:** SSR: set of properties extracted

1: Connect to database

2: cursor=db.cursor()

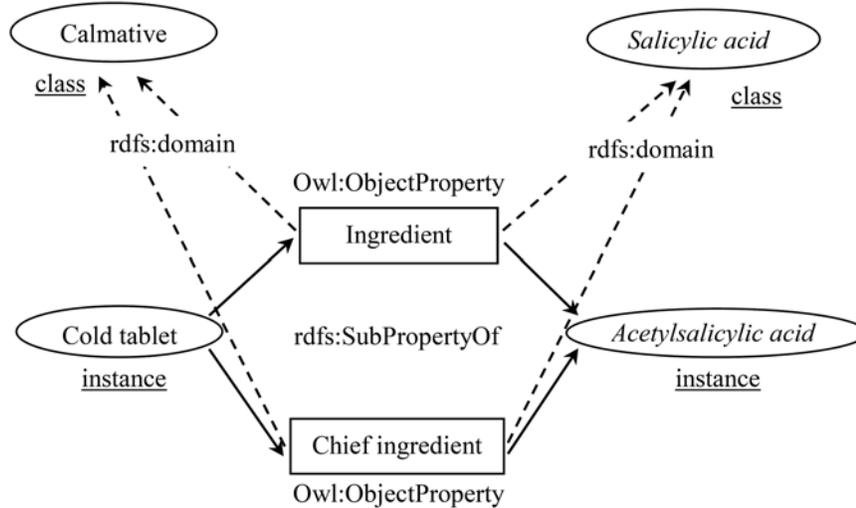
3: cursor.execute ('SELECT first\_term, second\_term, third\_term FROM TS WHERE first\_term NOT IN (SELECT first\_term FROM TS GROUP BY first\_term HAVING count (second\_term) = count (DISTINCT second\_term)) AND first\_term NOT IN (SELECT first\_term FROM TS GROUP BY first\_term HAVING count (DISTINCT second\_term) = 1) ORDER BY second\_term ASC')

4: triple\_set=cursor.fetchall()

5: Select super-sub property candidate, and compare property domain and property range of the candidate

6: SSR←properties with same domain and range

7: return SSR



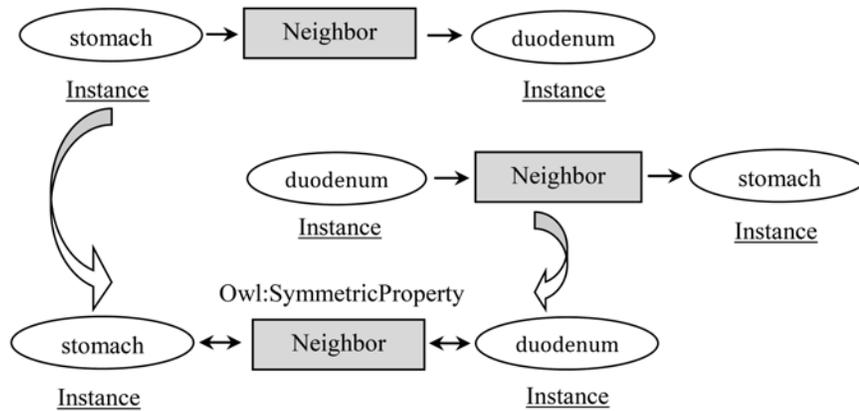
**Figure 7.** Generated super-sub relationship.

### 3. Generating Non-Hierarchical Relationships

In this section, we propose the detailed method and algorithm for generating non-hierarchical relationships (Owl:SymmetricProperty, Owl:TransitiveProperty, Owl:FunctionalProperty, Owl:InverseFunctionalProperty).

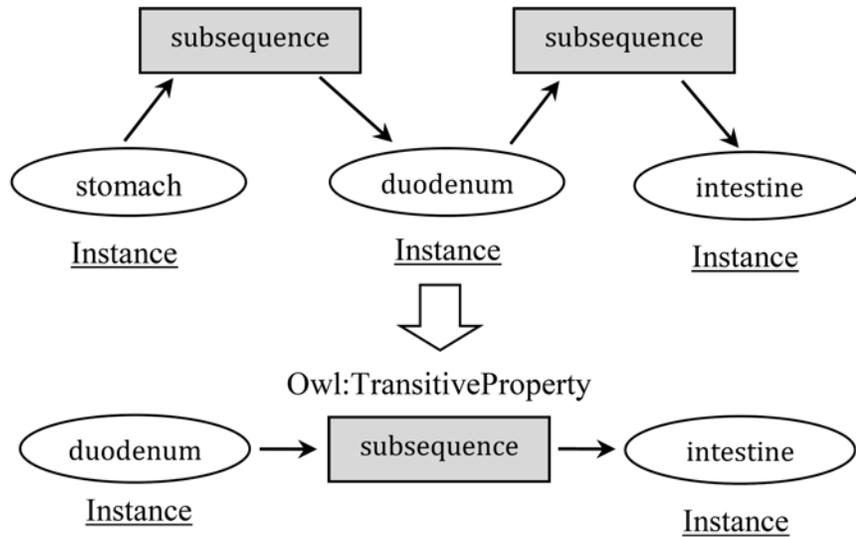
#### 3.1. Generating non-hierarchical relationships

In this subsection, we generate non-hierarchical relationships from the triples that are extracted by Subsection 2.3, i.e., we obtain a symmetric property, a transitive property, a functional property, and an inverse functional property from the tables. Firstly, we generate a symmetric property (Owl:SymmetricProperty). In case that is also established in a triple ‘subject Y - property P - object X’ if there is a property P that is established in a triple ‘subject X - property P - object Y’, the property P is known as a symmetric property. In order to derive the symmetric property, we match subject X and object Y corresponding to each property for entire triples that are extracted by Subsection 2.3. Figure 8 shows an example of a symmetric property extracted from the table.



**Figure 8.** Generated symmetric property.

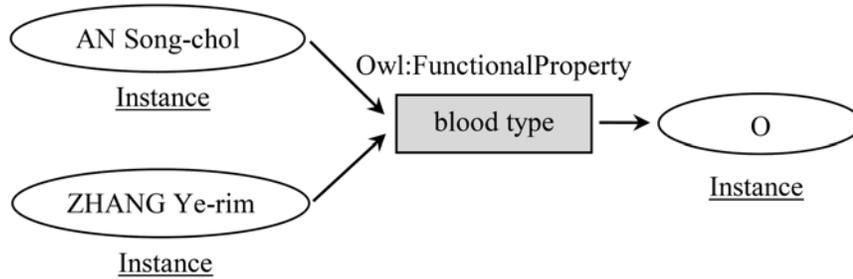
Then we generate a transitive property (Owl:TransitiveProperty). In case that is also established in a triple 'subject X - property P - object Z' if there is a property P that is established in a triple 'subject X - property P - object Y' and a triple 'subject Y- property P - object Z', the property P is known as a transitive property. We match subject X and object Y corresponding to each property for entire triples that are extracted by Subsection 2.3 and extract the transitive property. We remove symmetric properties from transitive properties obtained. Figure 9 shows an example of a transitive property extracted from the table.



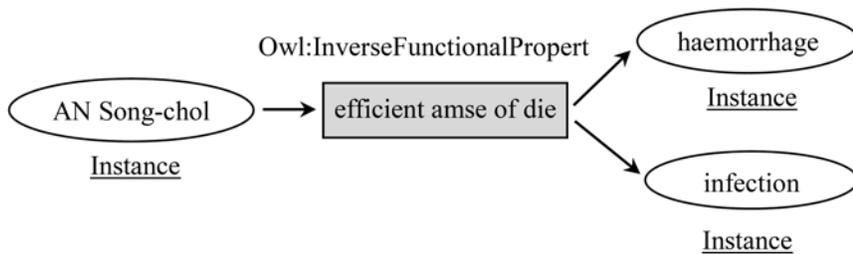
**Figure 9.** Generated transitive property.

Then we generate a functional property (Owl:FunctionalProperty). In case that object Y has at most one value for each subject X of a triple ‘subject X - property P - object Y’, the property P known as a functional property. Figure 10 shows an example of a functional property extracted from the table.

And lastly, we generate an inverse functional property (Owl:InverseFunctionalProperty). In case that subject X has at most one value for each object Y of a triple ‘subject X - property P - object Y’, the property P is known as an inverse functional property. Figure 11 shows an example of an inverse functional property extracted from the table.



**Figure 10.** Generated functional property.



**Figure 11.** Generated inverse functional property.

### 3.2. Algorithms for generating non-hierarchical relationships

We employ the tables of the relational database for saving the triples extracted. Firstly, Algorithm 2 shows a procedure for generating properties with a symmetric relationship. The pseudo-code given in Algorithm 2 is a few codes for programming language python. Algorithm 3 shows a procedure for extracting properties with the transitive relationship. Then Algorithm 4 shows a procedure for deriving properties with the functional relationship. Algorithm 5 shows a procedure for deriving properties with the inverse functional relationship.

**Algorithm 2.** SymmetricRelation(TS): derives all properties with the symmetric relationship from the set of triples

**Input:** TS: table of relational DB to extract properties with the symmetric relationship

**Output:** SR: set of triples extracted

1: Connect to database

2: cursor=db.cursor()

3: cursor.execute ('SELECT TS1. first\_term, TS1.second\_term, TS1.third\_term FROM TS AS TS1, TS AS TS2 WHERE TS1. first\_term= TS2. third\_term AND TS1. second\_term = TS2. second\_term AND TS1. third\_term= TS2. first\_term ORDER BY TS1. second\_term ASC')

4: SR =cursor.fetchall()

5: print SR

6: return SR

**Algorithm 3.** TransitiveRelation(TS): derives all properties with the transitive relationship from the set of triples

**Input:** TS: table of relational DB to extract properties with the transitive relationship

**Output:** TR: set of triples extracted

1: Connect to database

2: cursor=db.cursor()

3: cursor.execute ('SELECT TS1. first\_term, TS1.second\_term, TS1.third\_term FROM TS AS TS1, TS AS TS2, TS AS TS3 WHERE TS1.third\_term= TS2. first\_term AND TS1. second\_term= TS2. second\_term AND TS1. first\_term= TS3. first\_term AND TS2. second\_term= TS3. second\_term AND TS2. third\_term= TS3. third\_term ORDER BY TS1. second\_term ASC')

4: TR =cursor.fetchall()

5: print TR

6: return TR

**Algorithm 4.** FunctionalRelation(TS): derives all properties with the functional relationship from the set of triples

**Input:** TS: table of relational DB to extract properties with functional relationship

**Output:** FR: set of properties extracted

1: Connect to database

2: cursor=db.cursor()

3: cursor.execute ('SELECT DISTINCT second\_term FROM TS WHERE second\_term IN (SELECT second\_term FROM TS WHERE second\_term NOT IN (SELECT second\_term FROM TS GROUP BY first\_term, second\_term HAVING count(\*)>1)) AND second\_term NOT IN (SELECT second\_term FROM TS GROUP BY second\_term HAVING count(\*)=1) ORDER BY TS. second\_term ASC')

4: FR =cursor.fetchall()

5: print FR

6: return FR

**Algorithm 5.** InverseFunctionalRelation(TS): derives all properties with the inverse functional relationship from the set of triples

**Input:** TS: table of relational DB to extract properties with the inverse functional relationship

**Output:** IFR: set of properties extracted

1: Connect to database

2: cursor=db.cursor()

3: cursor.execute ('SELECT DISTINCT second\_term FROM TS WHERE second\_term IN (SELECT second\_term FROM TS WHERE second\_term NOT IN (SELECT second\_term FROM TS GROUP BY third\_term, second\_term HAVING count(\*) > 1)) AND second\_term NOT IN (SELECT second\_term FROM TS GROUP BY second\_term HAVING count (\*) = 1) ORDER BY TS. second\_term ASC')

4: IFR =cursor.fetchall()

5: print IFR

6: return IFR

#### 4. Evaluation

We implemented our algorithms and applied them to a set of the Web tables gathered from the Web pages of forensic medicine domain. In this section, we evaluated our method based on experimental observation. For generating of the property elements, we employed the table types which are discussed in the paper (Cha et al. [4]). In order to generate the property elements, firstly, we collected randomly 115 Web tables of forensic domain among around 2300 Web pages of the 'disease' domain using search key 'disease' and 'drug' of Google search engine. We also used a part of Peng's data [19]. Peng's collected 565 disease entries according to the tag of "disease" in Baidupedia. Then, according to the

method for extracting the property elements proposed in Section 2, we generate the property elements based on the experimental setting. In order to evaluate the experimental results, we used accuracy and as follows:

$$Accuracy = \frac{\text{Number of properties generated correctly}}{\text{Total number of generated properties}} \times 100; \quad (1)$$

$$Acquisition = \frac{\text{Number of properties generated correctly}}{\text{Total number of actual properties}} \times 100. \quad (2)$$

Tables 1 and 2 show the evaluation results of total property elements extracted by the experiment. Table 3 shows the evaluation results of the accuracy between our method and Peng's method.

**Table 1.** Evaluation of accuracy on the built hierarchical relationships and property restriction

Property elements	Number of properties	Accuracy (%)
Is-a relationship	102	98.43
Class-instance relationship	65	98.63
Super-sub relationship	77	99.50
Property domain	51	98.44
Property range	128	98.27
Total	423	98.65

**Table 2.** Evaluation of acquisition on the built non-hierarchical relationships

Property elements	Number of properties	Acquisition (%)
Symmetric property	28	67.34
Transitive property	16	51.78
Functional property	157	67.83
Inverse functional property	243	59.46
Total	444	61.6

**Table 3.** Compare our method with Peng’s alias relation extraction method

Item	Total	Correct	Wrong	Accuracy (%)
Peng’ result	938	925	13	98.61
Our result	423	417	6	98.65

In this experiment, the total number of the extracted triples is 24326, and the total number of the properties is 688, in which the number of data type property is 18 and the number of the object property is 9.

Until now, several ontology properties can be extracted from the tables in forensic medicine documents based on the proposed approach in the paper, and the experimental result shows that the approach is efficient. However, through the analysis of the approach and the experimental result, it can be found that there are also still several limitations of the current approach in this paper: (i) we found that the acquisition rate of the extracted non-hierarchical relationships is low when the triples extracted in this experiment are rather limited. Also, it should be pointed out that the extracted accuracy of data type property and object property greatly affects the accuracy of functional property and inverse functional property. Moreover, the purpose of using the acquisition rate in our experimental evaluation is to analyze the actually generated degree of non-hierarchical relationships. The non-hierarchical relationship extraction method proposed in our work only relies on the extracted triples, so the acquired non-hierarchical relationships are very accurate. However, if the property satisfying the definition condition of relationship does not exist in the extracted triples, then it can’t extract the non-hierarchical relationships; (ii) as mentioned in Section 1, the existing approaches for extracting table information did not discuss how to extract the sufficient property elements used for ontology reasoning. Therefore, we compared our method with Peng’s disease alias relation extraction method only about the performance effectiveness of the proposed hierarchical relationship generation algorithm; and (iii) although the approach in this paper can extract several ontology

properties (including is-a relationship, class-instance relationship, super-sub relationship, triple, properties for restricting properties, and non-hierarchical relationships as mentioned in the Sections 2 and 3 of the paper) from the Web tables in forensic medicine documents, in practice it is hard to cover every existing type of a table, and also the approach can't extract all kinds of properties occurring in the ontology definitions, such as equivalent relationship and properties with cardinality constraints (Bechhofer et al. [3]). In our near future work, we will further consider these aspects in depth.

## 5. Conclusion

In this paper, we proposed a method to build ontologies with sufficient property that is used for ontology reasoning in forensic medicine domain. Through the experimental observation for generating different properties from the Web tables in forensic medicine documents, we presented that, the Web tables are useful resources in which we can extract a non-hierarchical relation as well as hierarchical relations such as class-instance relationship, is-a relationship, and super-sub relationship. When interpreting and straightening out properties according to our proposed algorithm, to determine accurately property name, gives a decisive impact on the analysis of properties in the future. In our future work, a more precise method for generating other relationships will be developed based on extracted triples.

## Acknowledgement

This work was supported in part by the Science & Technology Development Fund of the University of Sciences (100/425-63053-243/1). The authors are grateful to our colleagues and criminalist at the Centre for Natural Science Research and Police College for their help in collecting the web pages of the Forensic Medicine Department and for discussing.

## References

- [1] G. Antoniou and F. van Harmelen, *A Semantic Web Primer*, Second Edition, Cambridge, MA: MIT Press, 2008.
- [2] K. Baclawski, E. S. Chan, D. Gawlick, A. Ghoneimy, K. Gross, Z. H. Liu and X. Zhang, Framework for ontology-driven decision making, *Applied Ontology* 12(3-4) (2017), 245-273.  
DOI: <https://doi.org/10.3233/AO-170189>
- [3] S. Bechhofer, F. V. Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. E. Patel-Schneider and L. A. Stein, *OWL: Ontology Web Language Reference*, Retrieved October 30 (2004), 2012.  
From <http://www.w3.org/TR/owl-ref>
- [4] S. Cha, Z. M. Ma, J. W. Cheng and F. Zhang, Learning of ontology from the web-table, In *Proceedings of the 8th International Conference on Fuzzy Systems and Knowledge Discovery* (2011), 1454-1458.  
DOI: <https://doi.org/10.1109/FSKD.2011.6019792>
- [5] S. Cha, Z. M. Ma and X. L. Jiao, Automatically extracting domain ontology based on semantic characteristics of web tables, *Journal of Computer Information Systems* 8(5) (2012), 1829-1836.
- [6] S. Cha and M. Han, A Bootstrapping method for automatic constructing of the web ontology instances and properties, *International Journal of Web & Semantic Technology* 9(3) (2018), 13-28.  
DOI: <https://doi.org/10.5121/ijwest.2018.9302>
- [7] M. M. Chavan and S. K. Shirgave, A methodology for extracting head contents from meaningful tables in web pages, In *Proceedings of the 2011 International Conference on Communication Systems and Network Technologies* (2011), 272-277.  
DOI: <https://doi.org/10.1109/CSNT.2011.66>
- [8] H. Dietze, T. Z. Berardini, R. E. Foulger, D. P. Hill, J. Lomax, D. Osumi-Sutherland, P. Roncaglia and C. J. Mungall, TermGenie – a web-application for pattern-based ontology class generation, *Journal of Biomedical Semantics* 5 (2014), 1-13; Article 48.  
DOI: <https://doi.org/10.1186/2041-1480-5-48>
- [9] Manvi, A. Dixit, K. K. Bhatia and B. Wadhwa, Generating domain specific ontology for retrieving hidden web contents, In *Proceedings of the International Conference on Information Systems and Computer Networks* (2014), 66-71.  
DOI: <https://doi.org/10.1109/ICISCON.2014.6965220>

- [10] M. El Asikri, J. Laassiri, S. D. Krit and H. Chaib, Contribution to ontologies building using the semantic Web and Web mining, In Proceedings of the International Conference on Engineering & MIS (2016).  
DOI: <https://doi.org/10.1109/ICEMIS.2016.7745329>
- [11] A. Goy, D. Magro and M. Rovera, On the role of thematic roles in a historical event ontology, *Applied Ontology* 13(1) (2018), 19-39.  
DOI: <https://doi.org/10.3233/AO-170192>
- [12] N. Guarino, D. Oberle and S. Staab, *What is an Ontology?*, 2nd Edition, Berlin: Springer (2009), 1-17.
- [13] C. Guo, S. Ma and D. R. Yuan, A web table extraction method based on structure and ontology, In *Advanced Data Mining and Applications*, LNAI 8933 (2014), 695-704.  
DOI: [https://doi.org/10.1007/978-3-319-14717-8\\_55](https://doi.org/10.1007/978-3-319-14717-8_55)
- [14] S. W. Jung, M. Y. Kang and H. C. Kwon, Hybrid approach to extracting information from web-tables, In *Proceedings of the 21st International Conference on the Computer Processing of Oriental Languages* (2006), 109-119.  
DOI: [https://doi.org/10.1007/11940098\\_11](https://doi.org/10.1007/11940098_11)
- [15] S. W. Jung, M. Y. Kang and H. C. Kwon, Constructing domain ontology using structural and semantic characteristics of web-table head, In *Proceedings of the 20th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (2007), 665-674.  
DOI: [https://doi.org/10.1007/978-3-540-73325-6\\_66](https://doi.org/10.1007/978-3-540-73325-6_66)
- [16] S. Jupp, T. Burdett, D. Welter, S. Sarntivijai, H. Parkinson and J. Malone, Webulous and the webulous Google add-on-a web service and application for ontology building from templates, *Journal of Biomedical Semantics* 7 (2016); Article 17.  
DOI: <https://doi.org/10.1186/s13326-016-0055-3>
- [17] B. T. G. S. Kumara, I. Paik, K. R. C. Koswatte and W. H. Chen, Ontology learning with complex data type for web service clustering, In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining* (2014), 129-136.  
DOI: <https://doi.org/10.1109/CIDM.2014.7008658>
- [18] L. Obrst, M. Gruninger, K. Baclawski, M. Bennett, D. Brickley, G. Berg-Cross, P. Hitzler, K. Janowicz, C. Kapp, O. Kutz, C. Lange, A. Levenchuk, F. Quattri, A. Rector, T. Schneider, S. Spero, A. Thessen, M. Vegetti, A. Vizedom, A. Westerinen, M. West and P. Yim, Semantic web and big data meets applied ontology: The Ontology Summit 2014, *Applied Ontology* 9(2) (2014), 155-170.  
DOI: <https://doi.org/10.3233/AO-140135>

- [19] J. Peng, Y. R. Du, Y. Chen, M. Zhao and B. Pei, Medical ontology learning based on Web resources, In Proceedings of the 12th Web Information System and Application Conference (2015), 116-119.  
DOI: <https://doi.org/10.1109/WISA.2015.10>
- [20] A. Pivk, P. Cimiano and Y. Sure, From tables to frames, In Proceedings of the Third International Semantic Web Conference (2004), 166-181.
- [21] R. A. H. M. Rupasingha, I. Paik and B. T. G. S. Kumara, Calculating web service similarity using ontology learning with machine learning, In Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (2015), 201-208.  
DOI: <https://doi.org/10.1109/ICCIC.2015.7435686>
- [22] R. A. H. M. Rupasingha, I. Paik, B. T. G. S. Kumara and T. H. A. S. Siriweera, Domain-aware web service clustering based on ontology generation by text mining, In Proceedings of the 7th IEEE Annual Information Technology, Electronics & Mobile Communication Conference (2016).  
DOI: <https://doi.org/10.1109/IEMCON.2016.7746301>
- [23] J. M. Serrano, Applied Ontology Engineering in Cloud Services, Networks and Management Systems, Heidelberg: Springer, 2012.
- [24] Q. X. Song, J. Liu, X. F. Wang and J. Wang, A novel automatic ontology construction method based on web data. In Proceedings of the 10<sup>th</sup> International Conference on Intelligent Information Hiding and Multimedia Signal Processing (2014), 762-765.  
DOI: <https://doi.org/10.1109/IIH-MSP.2014.194>
- [25] D. E. Spanos, P. Stavrou and N. Mitrou, Bringing relational databases into the semantic web: A survey, Semantic Web 3(2) (2012), 169-209.
- [26] M. Tanaka and T. Ishida, Ontology extraction from tables on the web, In Proceedings of the International Symposium on Applications and the Internet (2006), 284-290.  
DOI: <https://doi.org/10.1109/SAINT.2006.45>
- [27] C. Tao and D. W. Embley, Automatic hidden-web table interpretation, conceptualization, and semantic annotation, Data & Knowledge Engineering 68(7) (2009), 683-703.  
DOI: <https://doi.org/10.1016/j.datak.2009.02.010>
- [28] A. Vasilateanu, N. Goga, E. A. Tanase and I. Marin, Enterprise domain ontology learning from web-based corpus, In Proceedings of the 6<sup>th</sup> International Conference on Computing, Communication and Networking Technologies (2015), 112-117.  
DOI: <https://doi.org/10.1109/ICCCNT.2015.7395227>

- [29] Y. Y. Wei, R. J. Wang, Y. M. Hu and X. Wang, From web resources to agricultural ontology: A method for semi-automatic construction, *Journal of Integrative Agriculture* 11(5) (2012), 775-783.

DOI: [https://doi.org/10.1016/S2095-3119\(12\)60067-7](https://doi.org/10.1016/S2095-3119(12)60067-7)

- [30] H. Yu, X. Q. Lv and L. P. Xu, Use web resources to construct ontology concept hierarchy, In *Proceedings of the International Conference on Applied Science and Engineering Innovation* (2015), 1006-1011.

DOI: <https://doi.org/10.2991/asei-15.2015.198>

- [31] F. Zhang, Z. M. Ma and W. J. Li, Storing OWL ontologies in object-oriented databases, *Knowledge-Based Systems* 76 (2015), 240-255.

DOI: <https://doi.org/10.1016/j.knosys.2014.12.020>

