Ontology Extraction from Tables on the Web

Masahiro Tanaka and Toru Ishida
Department of Social Informatics, Kyoto University.
Kyoto 606-8501, JAPAN
mtanaka@kuis.kyoto-u.ac.jp, ishida@i.kyoto-u.ac.jp

Abstract

Previous works on information extraction from tables make use of prior knowledge such as a cognition model of tables or lexical knowledge bases for specific domains. However, we often need to interpret table structures in each table differently and to treat lexicons in various domains to more fully utilize the broad range of tables available on the Web. The method proposed in this paper uses relations represented by structures to extract an ontology from a table. Once the interpretations of table structures are given by humans, the table structures are automatically generalized to extract relations from the whole table. We define a formal representation of generalized table structure based on the adjacency of cells and iterative structures. As the result of the comparison with a method proposed in a previous work, it was shown that our method is suited to extraction of various relations which are needed for descriptions in RDF/OWL.

1 Introduction

A large amount of metadata is required to realize the Semantic Web. Tools for manual input and metadata management have been proposed[7, 9]. A content management system adding annotations to its contents is also proposed[10]. However, they are not suited to the creation of a large amount of metadata. Therefore, we need to automatically extract metadata from existing data stored on the Web. Some papers use HTML tag structures to extract information from Web pages[1, 5, 4].

In this paper, we propose a method for ontology extraction from semi-structured tables. Table structures represent relations between data in the table. Therefore, we can extract an ontology from a table on the basis of features of table structures. To extract ontologies from a broad range of tables on the Web, we focus on the following points:

Interpret Table Structures for Each Table The relation that a table structure represents depends on the table. We need to interpret the table structures in each table differently.

Cover Tables in Various Domains Tables collected from the Web often have contents that cover various domains. That is why it is important to process tables without domain-specific knowledge bases. To interpret table structures, previous works make use of prior knowledge such as types of tables determined by examining similarity of data[2, 13] and the cognition model of tables[11]. Domain-specific knowledge bases are also used[6, 12]. The aim of these previous works is to extract relations between data from tables in a specific domain automatically.

Our aim is to extract relations between data from tables in various domains semi-automatically using interpretations of table structures given to each table by humans. Our approach is as follows:

1. Give an interpretation of a table structure
2. Generalize the table structure
3. Extract relations from the whole table

Giving an interpretation to each table by humans enables us to interpret the table structure in each table differently. On the basis of the observation of table structures, we generalize the table structures, which represent particular relations, focusing on the adjacency of cells and iterative structures. We extract relations from the whole table using generalized table structures which represent particular relations. Our approach is easy to apply to tables in various domains because it does not use a domain-specific knowledge base.

In Section 2, we describe the observation of table structures. In Section 3, we explain the formalization of table structure based on our observation. We show the algorithm for extracting relations between data in Section 4. In Section 5, we evaluate our method comparing a method proposed in a previous work, and present a conclusion in Section 6.

2 Observation

In this section, we describe our observations of table structures and relations represented by the table structures.

2.1 Semantics of Table Structure

Table structures represent relations between data in the table. It is reasonable to suppose that a relation represented

Proceedings of the 2005 Symposium on Applications and the Internet (SAINT'06)
0-7695-2262-9/06 $20.00 © 2006 IEEE

Table 1. A price list of PC components

PC Component		
ProductID	ProductName	Price
Memory		
M27_512	PC2700 512MB	$70
M27_256	PC2700 256MB	$40
Processor		
P4_340	Pentium 4 3.40E GHz	$260
P4_280	Pentium 4 2.80A GHz	$140
A64_320	Athlon 64 3200+	$160

by a structure is consistent in one table. We call the relations represented by a table structure the semantics of the table structure.

Take Table 1 as an example. We can interpret "PC Component", "Memory" and "Processor" as classes which individuals belong to. We can also interpret "ProductID", "ProductName" and "Price" as properties of the individuals, and "P4_280", "Pentium 4 2.80A GHz" and "$140" as property values of these properties.

We can extract this relation between the data using semantics of the structure in Table 1, which is described as "A spanning cell represents a class which an individual belongs to. A row which consist of three cells between spanning cells represent properties of an individual. Consecutive rows which consists of three cells represent property values."

2.2 Assumption of Table Structure

When semantics of a structure is given, we can extract relations between data in the structure. To extract relations between data in various structures which have the same semantics, we need to generalize and represent table structures without changing the semantics of the structures.

On the basis of our observations of table structures, we make some assumptions for generalizing and representing table structures as is described below:

Adjacency of Cells We observed that two adjacent cells often have different kinds of data if an edge of the cell contains the edge of the other cell. We make an assumption that a structure of a row and a column is characterized by adjacency of cells and iterative structures in the row or the column.

Relation between Cells in a Row or a Column We observed that cells which are in a particular relation are positioned in a row or a column. We make an assumption that a relation between cells in a row or a column is represented by the structure of the row or the column. We also observed that the same kinds of data are described in serial cells in a row or a column that have the same features. This is the reason we focus on iterative structures in rows and/or columns.

Relation between Cells in Different Rows and Columns We observed that cells that are in different rows or columns and are in a particular relation have a relation with the cell at which the rows and columns intersect. We assume that a relation between cells in different rows or columns is represented by the structure of the region consisting of the rows and columns in which the cells lie.

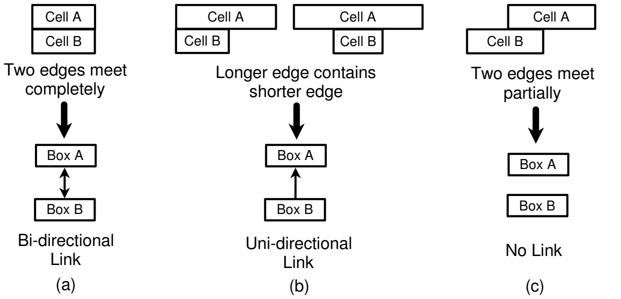


Figure 1. Correspondences between adjacency of cells and a link between boxes

We define a formal representation of generalized table structures based on these assumptions.

3 Formalization of Table Structure

To employ semantics of table structures, we need to formalize the table structures. We define a formal representation of table structures based on the assumptions described in 2.2.

3.1 Adjacency of Cells

As we described in 2.2, we assume the structure of a row and a column is characterized by adjacency of cells in the row or the column.

We introduce a "box" as an element corresponding to a cell, and adjacency of cells are represented as links between boxes. We also introduce a uni-directional link and a bi-directional link between boxes. A link between boxes represents the adjacency of two cells as is shown in Fig. 1.

The three types of adjacency of two cells are shown in the upper part of Fig. 1, and two boxes corresponding to the cells and the link between them are shown in the lower part of Fig. 1. "Box A" corresponds to "Cell A" and "Box B" corresponds to "Cell B". The link between "Box A" and "Box B" represents the adjacency between "Cell A" and "Cell B". When the edges of two adjacent cells meet completely, the two boxes corresponding to the cells are linked bi-directionally. When the longer edge contains the shorter edge, the two boxes corresponding to the cells are linked uni-directionally. When the two edges meet partially, two boxes are not linked.

3.2 A Structure Representing a Relation between Cells in the Same Row or Column

We assume that relations between cells in a row or a column are represented by the structure of the row or the col-

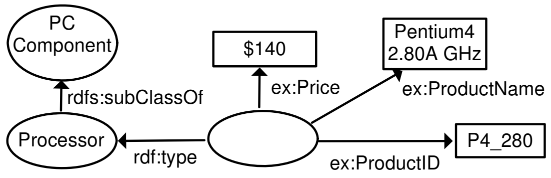


Figure 2. An RDF graph describing the relations between data in Table 1

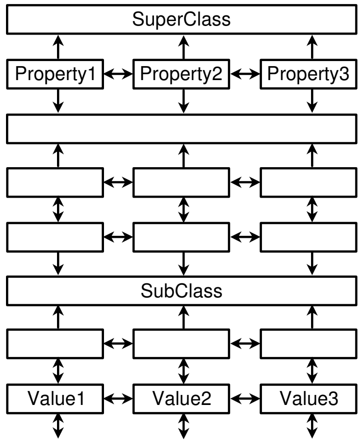


Figure 3. The structure representing relations shown in Fig. 2

umn because cells that are in a particular relation are often in a row or a column.

Take Table 1 as an example. Suppose that Fig. 2 represents the relation between "PC Component", "Processor", "ProductName" and "Pentium 4 2.80A GHz".

On the basis of the assumption, we focus on the structure of the second column in order to represent a relation between a property "ProductName" and a property value "Pentium 4 2.80A GHz". Same goes for two other properties. Cells in which classes "PC Component" and "Processor" are described span over three columns. Therefore we focus on the rectangular region which includes these cells. Fig. 3 shows the structure of the region. The structure is represented as linked boxes. The boxes corresponding to "PC Component" cell, "Processor" cell, cells in the second row and cells in the eighth row are labeled according to the semantics of the structure. The box corresponding to "PC Component" cell is labeled "SuperClass". The box corresponding to "Processor" cell is labeled "SubClass". It is because "PC Component" is a superclass of "Processor" and "Processor" is a class which individuals belong to. In a similar way, the box corresponding to "ProductID" is labeled "Property1" and the box corresponding to "P4_280" cell is labeled "Value1". The numbers added to the labels "Property" and "Value" represent the correspondence between a property and a property value.

We observed that the same kinds of data are often entered in serial cells which have the same features in a row or a column. On the basis of this observation, we assume serial boxes that have the same links should have the same label. Therefore, we integrate the boxes that have the same links to represent serial cells holding the same kinds of data. We introduce symbol "+" to represent structures that contain serial boxes that have the same links.

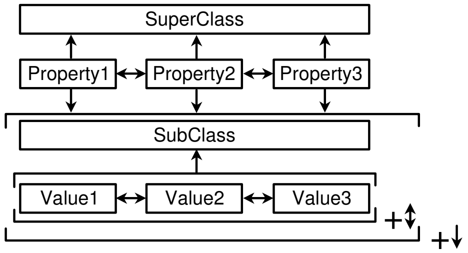


Figure 4. Generalized structures of Table 1

We can revise Fig. 3 to yield Fig. 4 using symbol "+". We do not integrate differently labeled boxes into one box in brackets because differently labeled boxes correspond to cells that have different kinds of data.

We define the representations of table structures that utilize boxes and symbol "+" as "generalized structures".

The generalized structure shown in Fig. 4 matches a structure which represents a relation between a superclass, one or more subclasses, properties and property values. The boxes labeled "Value1", "Value2" and "Value3" in brackets represents the structure of the rows holding the property values. The bi-directional link beside symbol "+" indicates that the three boxes in the brackets can be linked vertically and bi-directionally. The downward link beside symbol "+" indicates that the three boxes in the brackets can be linked downward to a box labeled "Subclass". labeled "Value" can be linked vertically and bi-directionally.

3.3 A Structure Representing a Relation between Cells in Different Rows and Columns

We assume a relation between cells in different rows or columns is represented by the structure of the region consisting of the rows and columns in which the cells are positioned.

Take Table 2 as an example. Let us suppose that we are given the fact an rdf graph shown in Fig. 5 represents. "Dimension8300" is a value of "hasName" property of an individual, "RAM" is a property of the individual and "256MB" is a value of "RAM" property. On the basis of this assumption, the relation between "Dimension8300", "RAM" and "256MB" is represented by the structure of the row and column that include the cells.

We focus on the shaded region in Table 3. Fig. 6 shows the structure of the shaded region in Table 2. The box corresponding to "Dimension8300" is labeled "Name" because "Dimension8300" is a value of "hasName" property. In a

Table 2. A region representing a relation between cells in different rows and columns

	CPU	RAM	HDD
OptiPlex240	P4 1.7GHz	512MB	60GB
Dimension8300	P4 2.4GHz	256MB	80GB
Pavilion505	Cel 2.4Ghz	256MB	40GB

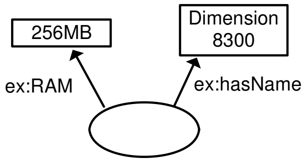


Figure 5. An RDF graph describing the relation between data in Table 2

similar way, the box corresponding to "RAM" is labeled "Property" and the box corresponding to "256MB" is labeled "Value".

We can revise Fig. 6 to yield Fig. 7(a)(b) using symbol "+". We use symbol "+" to represent the iteration of the structure that consists of three boxes. We do not use symbol "+" in the brackets when the structure we represent is a combination of structures of a row and a column. The box labeled "Property" of the generalized structure shown in Fig. 7(a) corresponds to cells in the first row because the box has no link on the top edge. If we used symbol "+" in the brackets, the box would also correspond to cells positioned in the second row. The link beside symbol "+" indicates that all boxes in the brackets can be linked horizontally and bi-directionally. The same goes for Fig. 7(b).

The generalized structure shown in Fig. 7(a) corresponds to reading the table horizontally. The generalized structure shown in Fig. 7(b) corresponds to reading the table vertically.

4 Algorithm for Extracting Table Structure

As described in Section 1, our approach is as follows:

1. Give an interpretation of a table structure
2. Generalize the table structure
3. Extract relations from the whole table

We will explain each step of this approach.

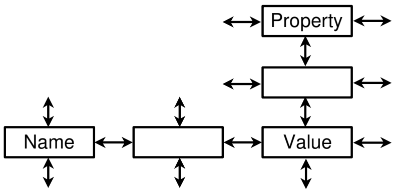


Figure 6. The structure of the shaded region in Table 2

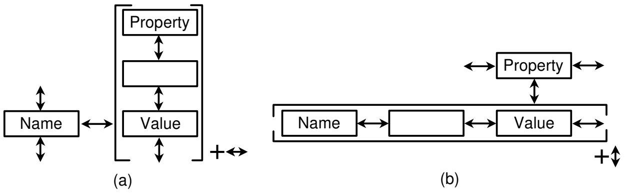


Figure 7. Generalized structures of the shaded region in Table 2

Give an Interpretation of a Table Structure It is not trivial how to exactly design an ontology[3]. On the other hand, ontologies published on the Web don't cover various contents in tables and their designs are not often suited to the interpretations which are needed. To interpret table structures in each table differently, we have to give interpretations of table structures by humans. We give a set of RDF statements describing the relation between data in a structure as an interpretation of a structure.

Take Table 1 as an example. The RDF graph shown in Fig. 2, which represents a relation between words in Table 1, corresponds to a set of RDF statements. This shows that one RDF statement cannot always describe a relation represented by a table structure. This is why we give a set of RDF statements as an interpretation of a table structure. We refer to a set of RDF statements describing a relation represented by a table structure as an "Episode".

Generalize the Table Structure Once an interpretation of a table structure is given as an episode, the structure is automatically generalized by performing the following steps. First, we find cells in which resources and properties of a given episode are described. Next, we represent the structure of the region which includes the found cells as linked boxes. Then we label the boxes according to semantics of the structure. Finally, we group serial boxes which have the same links and represent the structure using symbol "+". However, we do not look for properties such as rdf:type and rdfs:subClassOf because these properties are represented by the table structures themselves.

If the RDF graph shown in Fig. 2 is given, we can obtain the generalized structures shown in Fig. 4 from Table 1.

Extract Relations from the Whole Table We can extract relations between data in the structure which a generalized structure matches. A generalized structure represents relations between data in cells that the labeled boxes correspond to. To find structures that a generalized structure matches, we compare a box corresponding to a cell in a table and a box in the general structure. If the two boxes have the same links, they are considered identical.

If we try to extract relations from Table 1 using the generalized structure shown in Fig. 4, we can extract some episodes describing individuals such as an RDF graph shown in Fig. 8.

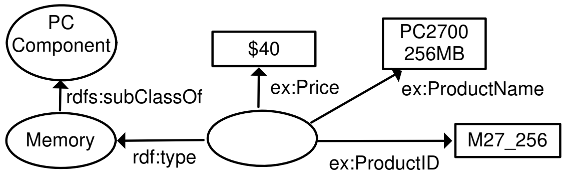


Figure 8. New episode extracted from Table 1

A generalized structure which represents inconsistent relations can be obtained using our generalizing method. Therefore we introduce the following heuristics in order to find such generalized structures.

- We do not use a generalized structure more than one properties or superclasses correspond to one property value or one subclass. It is because such a correspondence rarely appears in a table.
- We do not use a generalized structure in which all boxes has same characteristics. It is because such a generalized structure doesn't correspond to specific relations and matches almost every structure.

Sometimes resources and properties described in extracted episodes are found in the different structures from the structure which is given an interpretation. The extracted episodes can be considered as interpretations of the structures. Therefore, we iterate the steps described above using extracted episodes as interpretations of the structures until we cannot extract any new episodes.

As a result of the iteration, we sometimes find that some episodes describe a word as a class and some describe the word as property. This inconsistency is caused by the ambiguity of words. When such inconsistent episodes are extracted, we eliminate them from the set of extracted episodes.

5 Evaluation

As described in Section 1, the aims of our method are as follows:

- Interpret Table Structures for Each Table
- Cover Tables in Various Domains

We compare our method with the method proposed by Pivk et al.[11] in order to evaluate our method with respect to these points.

5.1 Features of Methods

Pivk's method [11] interprets tables based on the table cognition model[8] and generates a frame from a table. The features of this method are as follows:

Division of a Table into Regions A table is divided into regions. The method distinguishes between two kinds of regions. one represents attributes and another represents attribute values. The kinds of regions are determined based on the location of the regions and the type of data described in the regions such as string, number and date.

Relation between Consecutive Regions A relation between regions is determined based on the locations of the regions. When a region representing attributes and a region representing attribute values are consecutive, the method associates the two regions. When regions representing attributes are consecutive, the name of a method and a parameter of a frame consists of the attributes.

The features of our method are as follows:

Interpretations of Table Structures Our method makes use of interpretations of table structures given by humans. The interpretations are given as a set of RDF statements which describe relations between some data in a table.

Generalization of Table Structure Our method generalizes table structures based on adjacency of cells and iterative structures.

5.2 Evaluation Method

We evaluate our method by comparing our method with the method proposed in [11] focusing on the extraction of relations between data in a table. We will explain each step of the evaluation.

Step 1. We describe the relations between data in a table using RDF statements by humans. The description of anonymous resources and properties which don't appear in a table need to be added appropriately.

Step 2. We apply the two methods to a table. Pivk's method generates a frame from a table. We need to transform the frame into a set of RDF statements in order to compare the result of extraction by this method with the relations described by humans in Step 1. Therefore we consider methods and parameters of the frame as properties. We also consider the relation between attributes which consist of the name of a method or a parameter as a property hierarchy. Our method needs sets of RDF statements describing relations between data in a table as episodes. We describe episodes representing relations related to one individual and apply our algorithm using the episodes.

Step 3. We count the number of lacking descriptions of classes, properties, class hierarchies and property hierarchies extracted by the two methods. We consider relations described by humans in Step 1 as correct relations and compare these relations with the relations extracted by the two method. However, Pivk's method cannot extract properties which don't appear in a table. Therefore we don't care the correctness of names of methods and parameters of the generated frame corresponding to such properties.

5.3 Discussion of Results

The result of extraction depends on table structures. In [11, 13], the following three table classes are proposed with regard to the layout: 1-dimensional table, 2-dimensional table, complex table.

1-dimensinal table Tables belonging to this class have one or more rows of attributes above rows of attribute values. Attribute values correspond to an attribute in the same column. Table 1 belongs to this class.

2-dimensinal table Tables belonging to this class have a rectangular area in which attribute values are described. Attributes are described in one or more rows above the rectangular area and in one or more columns at the left side of the area. Table 2 belongs to this class.

Complex table Tables belonging to this class have various features. On the basis of some features, the following three subclasses of this class are shown in the previous works.

Partition label In a table belonging to this class, Special labels make several partitions of the table. Each partition shares the same attributes. The table shown in Fig. 9(a) is an example of this class.

Over-expanded label In a table belonging to this class, some attributes and some attribute values can expand over multiple cells. The table shown in Fig. 9(b) is an example of this class.

Combination A table belonging to this class consists of several smaller tables. The table shown in Fig. 9(c) is an example of this class. The first two rows and the lower seven rows can be interpreted as two structuraly independent tables.

We gathered tables from three different domains: price list, timetable and statistics. 25 tables were gathered in each domain and 5 tables for each table class were gathered. In order to gather tables, we used a search engine giving keywords "price list", "timetable" and "statistics" and chose tables which actually belong to these domains.

We carried out the steps described in 5.2 using the gathered tables. Table 3 shows the sum of the number of lacking descriptions of classes, properties, class hierarchies and property hierarchies for each table class. The average number of descriptions of classes, properties, class hierarchies and property hierarchies which given episodes represent are shown in the lower part of the second row.

From tables belonging to 1-dimensional table class or 2-dimensional table class, both of the two methods could extract correct correspondences between attributes and values. The reason that there is a difference between the results of extraction by the two methods is due to interpretations of attributes. When we describe relations between data in a table, we should interpret some attributes as properties and some attributes as property values. Which interpretations are correct depends on the semantics of the attributes. However, Pivk's method proposed cannot distinguish between the two kinds of attributes because the method basically interprets a table based on the types and the locations of data. Our method can correctly interpret attributes because our method uses interpretations given by humans. Similarly, attributes which are hierarchically positioned can represent a few kinds of relations such as a property hierarchy and a class and its properties. Our method can also interpret such attributes using interpretations of structures given by humans.

In some tables belonging to partition label class, attributes are described at the top of the table and no attribute is described in each partition. Pivk's method proposed cannot extract correspondences between attributes and attribute values from such tables because the method interprets tables based on the assumption that the regions representing attributes and related attribute values are consecutive. Our method can obtain correct correspondences between attributes and attribute values positioned separately because it uses interpretations given by humans describing the relation between the attributes and the attribute values. However, when a few kinds of data are described in the same structure as labels, our method cannot obtain the correspondence between the structure and the relation between data.

Many tables belonging to over-expanded label class or combination class have attributes and attribute values which are located separately and hierarchical attributes. This is the reason Pivk's method doesn't work well. Tables belonging to these classes often contain cells spanning over multiple rows or columns representing consecutive cells in which the same data are described. Therefore our method doesn't work well for such tables because the structures cannot be represented as iterative structures in which the same kinds of data are described. Table 3 also shows that there is little difference in the cost of giving episodes between simple table classes and complex table classes.

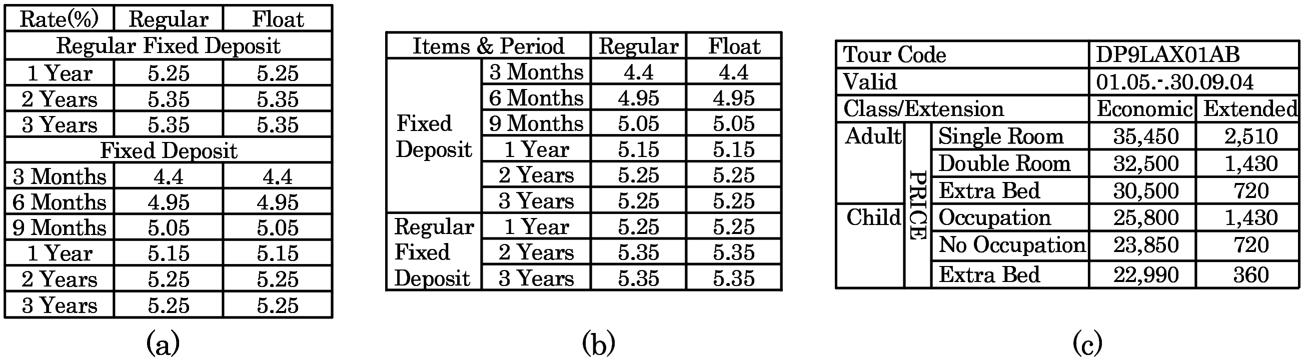
From what has been discussed above, we can conclude that our method can extract detailed relations using interpretations given by humans compared with the automatic approach based on prior knowledge such as a table cognition model and features of types and locations of data. This shows that our method is suited to extraction of various relations, such as property-property value pairs, class hierarchies and property hierarchies, which are needed for descriptions in RDF/OWL.

6 Conclusion

In this paper, we proposed a method for extracting an ontology from a table. The contributions of our work are as follows:

Extracting Relations Based on Given Interpretation Our method extracts relations based on interpretations given by humans, in order to interpret table structures in each table correctly.

Applicability to Tables in Various Domains Our method is easy to apply to tables in various domains because



Rate(%)	Regular	Float
Regular Fixed Deposit		
1 Year	5.25	5.25
2 Years	5.35	5.35
3 Years	5.35	5.35
Fixed Deposit		
3 Months	4.4	4.4
6 Months	4.95	4.95
9 Months	5.05	5.05
1 Year	5.15	5.15
2 Years	5.25	5.25
3 Years	5.25	5.25

(a)

Items & Period		Regular	Float
Fixed Deposit	3 Months	4.4	4.4
	6 Months	4.95	4.95
	9 Months	5.05	5.05
	1 Year	5.15	5.15
	2 Years	5.25	5.25
	3 Years	5.25	5.25
Regular Fixed Deposit	1 Year	5.25	5.25
	2 Years	5.35	5.35
	3 Years	5.35	5.35

(b)

Tour Code			DP9LAX01AB	
Valid			01.05.-.30.09.04	
Class/Extension			Economic	Extended
Adult	PRICE	Single Room	35,450	2,510
		Double Room	32,500	1,430
		Extra Bed	30,500	720
Child		Occupation	25,800	1,430
		No Occupation	23,850	720
		Extra Bed	22,990	360

(c)

Figure 9. Examples of tables

	1-dimensional	2-dimensional	Partition Label	Over-expanded Label	Combination
Method proposed in [11]	10	13	58	19	26
Our method (Given Episode)	1 (5.4)	5 (3.4)	14 (4.9)	16 (4.7)	23 (4.1)

Table 3. Comparison of the number of lacking descriptions

it uses interpretations given by humans and generalized table structures instead of a domain-specific knowledge base.

We applied our approach to many kinds of tables in order to show its usefulness and to clear up possible issues. As a result of our experiments, we confirmed that our method can extract detailed relations using interpretations given by humans compared with the automatic approach based on prior knowledge such as table cognition model and features of types and location of data. This shows that out method is suited to extraction of various relations which are needed for descriptions in RDF/OWL such as property-property value pairs, class hierarchies and property hierarchies.

Acknowledgment

This work was supported by a Grant-in-Aid for Scientific Research (A)(15200012, 2003-2005) from Japan Society for the Promotion of Science (JSPS).

References

[1] N. Ashish and C. Knoblock. Wrapper generation for semistructured internet source. *ACM SIGMOD Records*, 26–4:8–15, 1997.

[2] H. Chen, S. Tsai, and J. Tsai. Mining tables from large scale html texts. In *18th International Conference Computational Linguistics*, pages 166–172, 2000.

[3] H. Cho and T. Ishida. Designing metadata with existing application ontologies. In *IEEE/IPSJ Symposium on Applications and the Internet (SAINT-2006)*, 2006.

[4] P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *13th World Wide Web Conference*, pages 462–471, 2004.

[5] W. Cohen and W. Fan. Learning page-independent heuristics for extracting data from web pages. In *8th World Wide Web Conference*, pages 1641–1652, 1999.

[6] D. Embley, C. Tao, and S. Liddle. Automatically extracting ontologically specified data from html tables with unknown structure. In *21th International Conferenceon Conceptual Modeling*, pages 322–337, 2002.

[7] S. Handschuh and S. Staab. Authoring and annotation of web pages in cream. In *11th International World Wide Web Conference*, pages 462–473, 2002.

[8] M. Hurst. Layout and language: Beyond simple text for information interaction - modelling the table. In *2nd International Conference on Multimodal Interfaces*, pages 243–249, 1999.

[9] J. Kahan, M. Koivunen, E. Prud'Hommeaux, and R. Swick. Annotea: An open rdf infrastructure for shared web annotations. In *10th World Wide Web Conference*, pages 623–632, 2001.

[10] K. Nakatsuka and T. Ishida. Content management for inter-organizational projects using e-mail metaphor. In *IEEE/IPSJ Symposium on Applications and the Internet (SAINT-2006)*, 2006.

[11] A. Pivk, P. Cimiano, and Y. Sure. From tables to frames. In *3rd International Semantic Web Conference*, pages 166–181, 2004.

[12] Y. Tijerino, D. Embley, D. Lonsdale, and G. Nagy. Ontology generation from tables. In *4th International Conference on Web Information Systems Engineering*, pages 242–252, 2003.

[13] H. Wang, S. Wu, I. Wang, C. Sung, W. Hsu, and W. Shih. Semantic search on internet tabular information extraction for answering queries. In *9th International Conference on Information and Knowledge Management*, pages 243–249, 2000.