

Semantic Search on Internet Tabular Information Extraction for Answering Queries

H. L. Wang

Department of Computer Science,
National Tsing Hua University.

wyvern@cs.nthu.edu.tw

C. L. Sung

Department of Computer Science,
National Tsing Hua University.

clsung@rtlab.cs.nthu.edu.tw

S. H. Wu

Institute of Information Science,
Academia Sinica.

shwu@iis.sinica.edu.tw

W. L. Hsu

Institute of Information Science,
Academia Sinica.

hsu@iis.sinica.edu.tw

I. C. Wang

Institute of Information Science,
Academia Sinica.

kiki@iis.sinica.edu.tw

W. K. Shih

Department of Computer Science,
National Tsing Hua University.

wshih@cs.nthu.edu.tw

ABSTRACT

Although extracting information from tables is essential for Internet information agents, most tables are designed for human eyes and their layout and semantic meanings are not well defined. In practice, encoding the layout of each information source is impossible. This work presents a novel semantic search approach capable of extracting information from general tables. Semantic ontology allows our agents to read tables in the same knowledge domain with different layouts. In addition, a system of layout syntax and a set of transformation rules are defined to transform tables into databases without losing their semantic meanings.

Keywords

Information Extraction, Data Mining, Semantic Query, and Table Understanding.

1. INTRODUCTION

Much useful information is presented in tabular form on the Internet. Extracting information from these tabular information sources is essential for developing Internet information agents. For example, many web sites use tabular forms to list prices. Developing a price-comparing agent requires knowledge on how to extract the price information from these web sites. The agent may need to answer questions such as "Can you tell me the web site that sells the Palm Pilot at the lowest retail price?" Conventional information retrieval technologies cannot answer this question.

A software agent can answer this question through the following steps:

Step 1. Search the web pages related to the user's queries,

Step 2. Identify all related tables in all retrieved web pages,

Step 3. Extract all necessary information from the related tables,

Step 4. Integrate information read in Step 3, and

Step 5. Answer questions in appropriate forms.

The following three sub-steps can implement step 3:

Step 3.1 Identify the semantic relation of table cells,

Step 3.2 Convert the table into data with database form, and

Step 3.3 Extract target information from data in database form by query languages.

Time	Morning	Afternoon	Time	Mon	Morning	John Wang (2002)
Mon	John Wang (2002)	Indy Lai (2005)	Time	Mon	Afternoon	Indy Lai (2005)
Tue	Jimmy Lin (2007)	Wendy Lee (2001)	Time	Tue	Morning	Jimmy Lin (2007)
Wen	Indy Lai (2005)	John Wang (2002)	Time	Tue	Afternoon	Wendy Lee (2001)
Thr	Jimmy Lin (2007)	John Wang (2002)	Time	Wen	Morning	Indy Lai (2005)
Fri	Wendy Lee (2001)	Indy Lai (2005)	Time	Wen	Afternoon	John Wang (2002)
Sat	John Wang (2002)		Time	Thr	Morning	Jimmy Lin (2007)

(a)

(b)

Figure 1. (a) A hospital timetable. (b) The first seven reading paths of the timetable.

This work focuses on the conversion issue of tables. Our previous work concentrated on understanding the arrangement of table cells [8]. The query language in step 3.3 could be a natural language. Herein, a FAQ structure [6] is used to extract the constraints in queries and select targets from the database generated in Step 3.2.

The data mining approach in [7] must apply a learning process. In this study, the learning process is replaced by constructing a knowledge base in which all necessary ontology of natural language for reading table are included. Our approach has the potential to read tables in a website. Herein, a system of layout syntax is defined to denote the table layouts. This approach markedly differs from the approach of enumerating all possible layout features as applied by pertinent literature [1][2][3][4]. The layout syntax allows us to define a set of semantics preserving transformation rules to normalize tables with different layouts into standard database tables. For example, the table in Fig. 1 (a) is a timetable taken from a hospital's web page. Although humans can read the timetable, computers cannot. Figure 1(b) shows the corresponding reading path of each information unit, which is in a database format and comprehensible for a software agent. Although the timetable contains twelve reading paths, Fig. 1 (b) lists only the first seven.

Herein, the ability to understand a table is defined as the ability to answer queries asked by a normal user. For a table containing certain information that humans can read, the program should be able to read it as well. If humans can answer certain questions by reading a table, the program should be able to answer the questions by reading the table. However, tables can be simple or complex. The necessary semantic knowledge that an agent needs to understand a complex table may be unavailable in the ontology base. However, with some common knowledge on the common table, the agents can still extract certain information from the tables. Hence, in this paper, we propose a novel leveled knowledge schema to specify the relation between the knowledge and the information that can be extracted.

2. PROBLEM DEFINITION

This paper focuses on the following goal: "Given a table A and the related domain knowledge¹, convert table A into a database table B, while keeping the semantic equivalence of table A and table B." The complexity of tables must be investigated to achieve this goal. While Section 2.1 categorizes tables into different classes, Section 2.2 discusses how to process different classes of tables at different knowledge levels.

2.1 Table Classification

Previous works on table classification focused on extracting the table's logical structure [1][3][4]. Since this is insufficient for our purposes, this investigation classifies tables based on the semantic and layout structure knowledge.

From a semantic aspect, a **knowledge base** is used herein to identify the semantics of natural language queries and the content of tabular cells. Our knowledge base contains the ontology of **concepts** and **instances**. Each cell may have one information item. In our knowledge base, the item can be identified as either a concept or an instance of some concepts.

In the layout aspect, we propose three table classes: 1-dimensional tables, 2-dimensional tables, and complex tables. The first two classes are used more often and are simpler than complex tables. Herein, table cells are divided into labels and entries according to the identification in the semantic aspect.

1-dimensional tables: 1-dimensional tables may have one or several rows of labels above the rows of entries. The tables in a relational database are 1-dimensional table. The entries in a different column represent instances of different concepts. The labels of each column identify the categories of concepts by which we access entries in the column. For example, Fig. 2(a) illustrates the style of a normal 1-dimensional table and Fig. 2 provides an example of a 1-dimensional table.

2-dimensional tables: 2-dimensional tables have a rectangular area of similar entries. Each entry in this rectangular area represents an instance of the same concept. One or several rows of labels are above the rectangular area. These labels are instances of

¹ Another problem is from where and how the knowledge is obtained. Both the input query and the input table contain the semantics that can refer to their domain knowledge. Our approach allows us to obtain related domain knowledge from the input query since it must be known when extracting the query constraint.

other concepts. Instances in different rows belong to different concepts. At the left side of the rectangular area could be one to several columns of labels. These labels are also instances of different concepts. Instances in the same column belong to the same concept. Instances in different columns belong to different concepts. Above the columns of entries can be their concepts or empty cells. For example, Figs. 3(a)(b) show styles of 2-dimensional tables and Figs. 3(c)(d) provide illustrative examples.

C_1	...	C_n
I_1^{*c}	...	I_n^{*c}

Flight	Day	Departs	Arrives	Stops/Via
LM202	Sun	11:10	13:20	0
LM208	Mon	20:20	21:45	0
LM208	Thu	20:45	22:10	0
LM208	Tue	20:00	22:10	0
LM451	Mon	10:40	12:05	0
LM963	Tue	10:40	12:05	0

(a) (b)

Figure 2. (a) An abstract 1-dimensional table, (b) A 1-dimensional table example of (a). Where C_X denotes a string representing concept X in a cell and I_X denotes a string representing an instance of concept X in a cell. In addition, $*c$ denotes a repetition in a column and $*r$ denotes a repetition in a row. Moreover, $$ denotes a repetition in a rectangular area. Section 3.2 provides more details.**

?	I_Y^{*r}
I_X^{*c}	I_Z^{**}

C_1	...	C_n	I_Y^{*r}
I_1^{*c}	...	I_n^{*c}	I_Z^{**}

Time	Morning	Afternoon
Mon	John Wang (2002)	Indy Lai (2005)
Tue	Jimmy Lin (2007)	Wendy Lee (2001)
Wen	Indy Lai (2005)	John Wang (2002)
Thr	Jimmy Lin (2007)	John Wang (2002)
Fri	Wendy Lee (2001)	Indy Lai (2005)
Sat	John Wang (2002)	

	Room	Mon	Tue	Wed	Thu	Fri	Sat
Morning	1	Joe Jiang	Jean Tasi	Joe Huang	Hellon Yuo	Hellon Yuo	Collin Lee
	2		Joe Jiang	Collin Lee	Jean Tasi	Joe Jiang	
	3	Jean Tasi	Collin Lee	Jean Tasi	Joe Huang	Collin Lee	Hellon Yuo
Afternoon	1	Collin Lee	Jean Tasi	Hellon Yuo	Joe Jiang	Joe Huang	
	2		Joe Jiang			Jean Tasi	
	3		Joe Huang	Collin Lee	Hellon Yuo	Collin Lee	

(a) (b) (c) (d)

Figure 3. (a) (b) are abstract 2-dimensional tables, (c) is a 2-dimensional table example of (a), and (d) is a 2-dimensional table example of (b).

Complex table: In addition to 1-dimensional tables and 2-dimensional tables, there are complex tables. Complex tables can have many features. Some of those features are described as follows:

- **Partition label:** Special labels between the data entries can make several partitions on the data entries. Each partition shares the same labels at the top of the table. For example, Figs. 4(a) and 7 illustrate this situation. In addition to that semantic identification in tables with this feature is complicated, the concept-instance relation cannot be obtained in closing cells.
- **Over-expanded label:** For example, in Fig. 4(b), each entry I_X spans two or three I_Y under the same label C_X . In Fig. 6, there are two labels span 11 and 4 sub-labels, respectively.
- **Combination:** Some large tables are a combination of several similar small tables. For example, in Fig. 4(c), four small tables

merge into a larger one. In Fig. 5, two tables are merged into a larger one.

■ **Multiple items in a cell:** Some tables may have two or more items, can be recognized as instances of the same or different concepts, placed in the same cell. For example, in Fig. 4(d), each entry has two instances in the second column. In Fig. 5, many cells have two different items: the doctor's name and the ID number of the doctor². The relation between items in the same cell is referred to as inner cell relation [4].

■ **Vertical writing:** The sequence of text is presented vertically in a cell, which commonly occurs in Chinese web pages. For example, in Fig. 4(e), "ABC" and "DEF" are written vertically in a cell. However, the real sequence in the HTML source is "ADBEFC".

■ **Forward reduction:** For example, comparing Fig. 4(f) and Fig. 4(g) reveals two empty cells in Fig. 4(f), which are not meaningless.

C_{X1}	...	C_{Xn}
I _p		
I_{X1}^c	...	I_{Xn}^c
I _p		
I_{X1}^c	...	I_{Xn}^c

(a)

C_X	C_Z
I _Y	I _Z
I _X	I _Y
I _Y	I _Z
I _X	I _Y
I _Y	I _Z
I _X	I _Y
I _Y	I _Z

(b)

C_X	C_Y	C_X	C_Y
I_X^c	I_Y^c	I_X^c	I_Y^c
C_X	C_Y	C_X	C_Y
I_X^c	I_Y^c	I_X^c	I_Y^c

(c)

C_X	C_Y	C_Z
I _X	I _Y	I _Z
I _X	I _Y	I _Z
I _X	I _Y	I _Z
I _X	I _Y	I _Z
I _X	I _Y	I _Z

(d)

A D
B E
C F

(e)

Date	Room	Doctor
Mon	1	Jenny
	2	Penny
Tue	1	David
	1	Josh

(f)

Date	Room	Doctor
Mon	1	Jenny
	2	Penny
Tue	1	David
	1	Josh

(g)

Figure 4. (a) Partition labels. (b) Over-spanned labels. (c) Combination. (d) Multiple items in a cell. (e) Vertical writing. (f) Forward reduction. (g) Original form of (f).

Morning					
Room	Mon	Tue	Wed	Thu	Fri
1	Y. C. Chen 10201	H. T. Hur 10201	H. S. Yang 10201	Y. C. Chen 10201	H. S. Yang 10201
2	J. L. Chen 10202	C. H. Lee 10202	H. T. Hur10202	H. S. Yang 10202	H. T. Hur10202
3		T. L. Tsai 10213	T. L. Tsai 10203	H. T. Hur10203	J. L. Chen 10203
Afternoon					
Room	Mon	Tue	Wed	Thu	Fri
1	T. L. Tsai 20201	L. H. Huang 20201	J. L. Chen 20201	J. L. Chen 20201	T. L. Tsai 20201
2	C. H. Lee 20202	A. C. Lai 20202			C. H. Lee 20202

Figure 5. An illustrative example of combination and multiple items in a cell

² Multiple items can occasionally be represented as one item if the distribution in every cell is regular. For example, the doctor's name and ID is an example of the concept: 'doctor name & id'.

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

Figure 6. A table with over-spanned labels

Rate (%)	Regular	Float
Regular Fixed Deposit		
1 Year	5.05	5.05
2 Years	5.1	5.1
3 Years	5.1	5.1
Fixed Deposit		
3 Monthes	4.35	4.35
6 Monthes	4.6	4.6
9 Monthes	4.7	4.7
1 Year	5	5
2 Years	5.05	5.05
3 Years	5.05	5.05

Figure 7. A table with over-partition labels

2.2 Knowledge Levels

Developing a generic algorithm to understand all tables is extremely difficult. Although certain tables can be understood with less knowledge, others cannot. In particular, tables in different knowledge domains can have different layout features. For example, consider the train timetable in Fig. 8 in which the concept *train number* can be used to access instances of its descent concepts *origin station* (KEE-LUNG) and *destination station* (PING-TUNG). There is still no explicit clue for selecting the descent concept that the string 'KEE-LUNG' should be treated as an instance of.

Herein, the understanding of table is divided into different classes based on different levels of knowledge required to understand tables. In doing so, four different levels of knowledge will be investigated.

Level 0: No table knowledge is supported. Without any table knowledge, a trivial search can be done by some keyword searching rules. However, only trivial questions can be answered.

Level 1: Preliminary table layout knowledge is supported, i.e., treat the tables as standard 1-dimensional or 2-dimensional tables. With knowledge of this level, the position of each text string can be recognized. The ability to answer questions does not markedly improve. At this level, tabular position accessing questions can be answered.

Level 2: Knowledge to distinguish between concept and instance is supported. With knowledge of this level, agents can identify the semantics of the content in each cell and the query constraints in a natural language.

Level 3: Domain knowledge to distinguish different tables is supported. With knowledge of this level, the agent can read more complex tables.

Our idea is to denote the possible layouts by **layout syntax grammar** in different table domains and use these denotations to do template matching. The matched template is used to determine the semantic of cell content. **Semantics preserving transformation** is then applied to do the layout transformation.

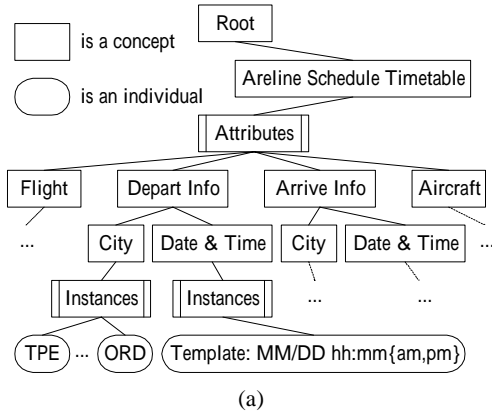
Train Number				Start Time	Arrival Time	Remark
CHU KUANG	11	TAL-TUNG NEW	To	KAO-HSIUNG	05:15 07:52	Every Day
FU HSING	101	KEE-LUNG	To	PING-TUNG	06:03 09:08	Every Day
TZE CHIANG	1003	M	To	KAO-HSIUNG	07:00 09:16	Every Day
TZE CHIANG	1005	M	To	KAO-HSIUNG	07:40 09:46	For FRI SAT SUN only.
TZE CHIANG	1007	M	To	KAO-HSIUNG	07:55 10:11	Every Day
CHU KUANG	41	M	To	CHIA-I	08:23 11:06	Every Day
TZE CHIANG	1009	M	To	KAO-HSIUNG	08:40 10:40	Every Day
TZE CHIANG	1011	M	To	KAO-HSIUNG	09:00 11:15	Every Day

Figure 8. Partial view of a Taiwan train timetable

3. METHODOLOGY

3.1 Knowledge Base

Our knowledge base, constructed with a concept-sensitive model [5], contains the ontology knowledge to identify the semantics in text strings. For example, the ontology in Fig. 9(a), which is used to read the airline schedule in Fig. 9(b), contains the attributes of *flight*, *departure information*, *arrival information*, and *aircraft types*. The departure information has sub-attributes of *city* and *date-&time*. The arrival information also has the same sub-attributes. Each attribute is a **concept**, and the sub-attributes are called **descent concepts**. Each attribute has its **synonym** and **instances**. For example, the concept of departing information may appear as "depart" or "departing". Instances of the departing city can be the name or shortcut of any international cities.



Flight	Departing		Arriving		Aircraft
	City	Date & Time	City	Date & Time	
EVA Airways 10	TPE	07/14 11:50pm	YVR	07/14 07:40pm	744
American Airlines 6647 *	YVR	07/14 09:00pm	LAX	07/14 11:45pm	737
Air Canada 9800 *	TPE	07/14 11:50pm	YVR	07/14 07:40pm	744
American Airlines 6501 *\$	YVR	07/15 06:35am	LAX	07/15 09:27am	737
China Airlines 61	TPE	07/14 08:10pm	FRA	07/15 06:50am	M11
American Airlines 83\$	FRA	07/15 10:40am	ORD	07/15 01:05pm	763
American Airlines 473\$	ORD	07/15 02:30pm	LAX	07/15 04:35pm	738

(b)

Figure 9. (a) Ontology for airline schedule timetables, (b) an airline schedule timetable.

Ontology knowledge is first applied by template matching skills [8] to recognize possible semantic items in the text of each cell. Ambiguities might arise when the same text fragment can be

recognized as different concepts or instances of different concepts. For example, "TPE" can be recognized as instances of the departing city or the arriving city.

Fortunately, individuals often arrange items in some meaningful order and place related items close together, implying that the ambiguities can be reduced by identifying the relationships between cells. For example, in the same column, the term "departing" is recognized as departure information and the term "city" is recognized as the city of departure information and the city of arrival information. The term "city" below the term "departing", the city of departure information is more closely related to the departure information. Hence, the city of departure information for this cell is selected herein instead of the city of arrival information.

In our approach, four relationships are used to distinguish the role of each cell as a table label or table data. These four relationships are ordered by their importance and listed as follows:

- (1) A concept to one of its instances, denoted as C_X-I_X ;
- (2) A concept to one of its descent concepts, denoted as C_X-C_X' ;
- (3) A concept to an instance of its descent concept, denoted as C_X-I_X' ; and
- (4) An instance to another instance of the same concept, denoted as I_X-I_X .

3.2 Table Layout Syntax

Our table layout syntax consists of semantic symbols and layout symbols. They are used as area notations and concatenating operators respectively.

Semantic cell notations:

C_X denotes a string represents concept X in a cell.

I_X denotes a string represents an instance of concept X in a cell.

? is a don't-care symbol, and can be any kind of cell.

Area notations:

c denotes a repetition in a column. For example, I_X^c denotes Fig. 10(a).

r denotes a repetition in a row. For example, I_X^r denotes Fig. 10(b).

** denotes a repetition in a rectangle area. For example, I_X^{**} represents Fig. 10(c). The first * is size of rows and the second * is size of columns.

The above * can also be replaced by numbers, variables or expressions to denote a deterministic size of repetition. For example, I_X^{3r} represent three concatenate cells of I_X in the same row.

Concatenating operator:

| is an operator capable of concatenating two sides in top-bottom direction. For example, $C_X|I_X^c$ represents Fig. 10(d).

- is an operator capable of concatenating two sides in left-right direction. For example, $C_X-I_X^r$ represents Fig. 10(f).

() is a compound operator that guarantees the inner expression is calculated first.

Π represents a repetition of left-right concatenation. For example, in Fig. 10(e), $\Pi_{i=1}^n (C_{X_i} I_{X_i}^{*c}) = (C_{X_1} I_{X_1}^{*c}) \dots (C_{X_n} I_{X_n}^{*c})$ and $I_X^{*r} = \Pi_{i=1}^n I_{X_i}$.

Σ represents a repetition of top-bottom concatenation. For example, in Fig. 10(g), $\Sigma_{i=1}^n (C_{X_i} I_{X_i}^{*r}) = (C_{X_1} I_{X_1}^{*r}) \dots (C_{X_n} I_{X_n}^{*r})$ and $I_X^{*c} = \Sigma_{i=1}^n I_{X_i}$.

Definition: A monotonic area is a rectangular area of cells that have the same concept, implying that they are strings representing the same concept or they are strings representing instances of the same concept. Monotonic areas are C_X , I_X , C_X^{*r} , I_X^{*r} , C_X^{*c} , I_X^{*c} , C_X^{**} , and I_X^{**} .

Lemma: If a table can be recursively binary partitioned until each partition is a monotonic area, then this table can be represented by table layout syntax.

Proof: For each partition P , a binary partition produces two partitions P_X and P_Y . Assume that P_X and P_Y can be represented by the layout syntax $G(P_X)$ and $G(P_Y)$, respectively. Since P_X and P_Y are produced by a binary partition, they are either concatenated from top to bottom or from left to right. Hence, $G(P) = P_X P_Y$ if they are concatenated from top to bottom. $G(P) = P_X P_Y$ if they are concatenated from left to right.

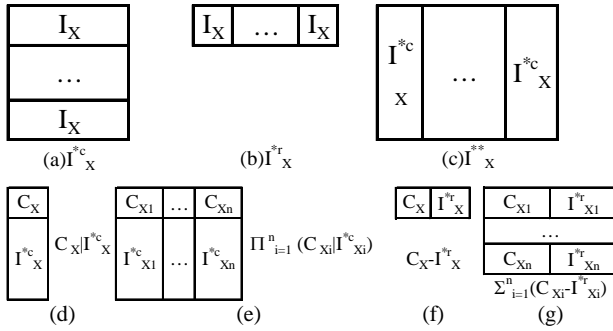


Figure 10. Areas represented by layout syntax grammar.

3.3 Semantics Preserving Transformation

The semantics preserving transformation is a mapping from a layout to another layout, while keeping the semantic relations between cells. This transformation aims to transform the arrangement from 1-dimensional or 2-dimensional tables into a standard database. A standard database can assist the storage of data and compare data from different information resources.

Herein, Γ is defined to be a set of layouts. For example, $\Gamma_{i=1,3} (I_{X_i} I_{Y_i}^{*r})$ is a set of tables: $I_{X_1} I_{Y_1}^{*r}$, $I_{X_2} I_{Y_2}^{*r}$, $I_{X_3} I_{Y_3}^{*r}$. For 1-dimensional and 2-dimensional tables, the following two transformation rules can be used to derive the corresponding database table.

Table 1. Table transformation rules

1-Dimensional Table	From	$C_X I_X^{*r}$
	To	$C_X I_X^{*r}$
2-Dimensional Table	From	$(? I_X^{(m-h)c} - (?^{(h-1)*} I_{X_i}^{*r} I_Y^{(m-h)*} Z))$
	To	$\Gamma_{i=1, (m-h)} (?^{(h-1)*} I_{X_i} I_Y^{*r} I^{(m-h)*} Z)$

3.4 Natural Language Query

The ability to concisely recognize the query constraints, output fields, and target tables in the natural language (*NL*) query input would allow us to implement this *NL* query by a SQL SELECT command “SELECT <output fields> from <target table> where <constraints>”. Three major questions must be answered: (a) How can we recognize the necessary information in a *NL* query? (b) How can we obtain a systematic approach to define tables and its fields in order to extract information and answer queries? and (c) How can we manage all database tables created to store the information extracted from web tables?

First, the field names of a database table must be defined. The fields of query constraints and the fields of the target database table must have the same name if they represent the same meaning. For example, for queries on the airline schedule timetables, the “when ... depart?” in a query and the “departure time” in a table access the same concept. In the proposed approach, the concepts are unique identities with respect to their various representations. In addition, the ontology in our knowledge base can be applied to recognize concepts and their instances in text strings, and make no difference on queries and table cells. For implementation, directly applying the concept identity string³ in our knowledge base as the database field names is an effective approach. Hence, field names are concept identity strings, and field values are their instance values. The query constraints are instance values and output fields are concepts. In this manner, problem (b) can be solved.

Problem (a) can be solved by FAQ structures in [6]. In the FAQ structure, keywords, concepts, instances, and their order are denoted for pattern matching. The full matched FAQ structure is selected to recognize the necessary information in the input query. The FAQ structure also contains a schema to control the set of query constraints, output fields, and the target table. The target table and query constraints are set from matched instances. In an intelligent agent, users might occasionally ask questions interactively. Hence, some information from previous interactions must be known.

Problem (c) is an information integration problem. All of the extracted information cannot be simply merged in the same database table for the same ontology because some extra information may be required, such as constraints; however, they do not appear in the input tables. For example, their timetables do not contain the ownership information of the hospital name and address. Simply merging the information extracted from timetables of all hospitals and ignoring their affiliation would not allow us to distinguish the ownerships from the query results. Hence, information extracted by the same ontology might be stored in individual tables. In addition, extra information may be required to select the target tables before accessing the data in these tables. Importantly, the information extracted outside the tables must be integrated using information integration approaches [9].

³ All the nodes in our knowledge bases have their unique identities.

4. EXAMPLE⁴ AND EXPERIMENT

Consider the table in Fig. 11(a). Its class can be recognized and, then, it can be transformed into a standard database. Applying the ontology in Fig. 12 allows us to recognize the semantics of cells in this table as shown in the table of Fig. 11(b). In this table, cells are grouped into four monotonic areas as shown in the table of Fig. 11(c). Applying the transformation mechanism in section 3.3 for 2-dimensional tables, we rearrange the cells in the form of Fig. 11(d). Finally the table in Fig. 11(a) is transformed into the database format table in Fig. 11(e). In the database format table, every row is a reading path of the table in Fig. 3.

The effectiveness of the proposed approach is demonstrated by selecting tables from twenty three web pages. In our experiment, the above two semantics preserving transitions in section 3.3 are used to transform the database format. Herein, only one ontology is used to identify the semantic information for each cell. Among our twenty three input tables, thirteen are 2-dimensional tables, and ten are complex ones. Figure 13(a) summarizes our preliminary results, which are highly promising for 2-dimensional tables. In addition, a success ratio of 84.62% is achieved for 2-dimensional tables. For 2-dimensional tables, eleven of them can be transformed into database tables without any error. Another one failed due to too many empty cells and the other one failed due to the lack of ontology knowledge to recognize necessary information for some cells. However, all of the ten complex tables failed. They contain features of combination, partition, vertical writing, forward reduction, and errors. The features of combination, partition, and vertical writing can be removed by a preprocessing. To demonstrate that our approach can be extended to complex tables, we apply a hand preprocessing to remove these features on eight related complex tables. Among these tables, seven can be transformed into database format correctly. The total success ratio increased to 70%. The total success ratio increased from 47.83 to 78.26. The other features require advanced knowledge level for further recognition.

Without preprocessing				With preprocessing			
Cases	Success	Fail	Rate (%)	Cases	Success	Fail	Rate (%)
2D	11	2	84.62	2D	11	2	84.62
Complex	0	10	0.00	Complex	7	3	70.00
Total	11	12	47.83	Total	18	5	78.26

Figure 13. (a) Experimental results without preprocessing, (b) experimental results with preprocessing to remove features for complex tables.

5. CONCLUSION

This work demonstrates how intelligent agents can extract the tabular information for answering queries. With the assistance of ontology knowledge, the intelligent agents can distinguish concepts and instances in each table cell. With the layout syntax grammar, intelligent agents can recognize a particular layout. In addition, the semantics preserving transformation is developed to transfer a table into the standard database form. This paper also presents a picture of investigation on the tables with respect to different table classes crossing different knowledge levels.

6. REFERENCES

- [1] Xinxin Wang (1996), "Tabular abstraction, editing, and formatting," PhD Thesis, Department of Computer Science, University of Waterloo.
- [2] Hurst, Matthew & Douglas, S. (1997). "Layout and Language: Preliminary investigations in recognizing the structure of tables," Proceedings of the Fourth International Conference on Document Analysis and Recognition, 28-31 August, Ulm, Germany.
- [3] Douglas, Shona, Hurst, M., & Quinn, D. (1995). "Using Natural Language Processing for Identifying and Interpreting Tables in Plain Text," Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval, pp.535-546, Las Vegas, Nevada. University of Nevada.
- [4] Hurst, Matthew (1999), "Layout and Language: Beyond Simple Text for Information Interaction - Modelling The Table," Proceedings of The 2nd International Conference on Multimodal Interfaces, Hong Kong
- [5] Wen-Lian Hsu, Yi-Shiou Chen and Yuan-Kai Wang (1998), "A Context sensitive model for concept understanding," Proceedings of ITALLC 98, pp.161-169.
- [6] Wen-Lian Hsu, Yi-Shiou Chen and Yuan-Kai Wang (1999), "Natural language agents – An agent society on the Internet," Proceedings of PRIMA 99.
- [7] Chun-Nan Hsu and Chien-Chi Chang. "Finite-State Transducers for Semi-Structured Text Mining (1999)," Proceedings of IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications, Stockholm, Sweden.
- [8] Huei-Long Wang, W. L. Hsu, Y. S. Chen, T. L. Lau, C. H. Tang, H. M. Yeh, W. K. Shih (1999), "A Streamlined Approach for Tabular Information Extraction," Proceedings of NCS99.
- [9] Chun-Nan Hsu and Craig A. Knoblock. "Semantic Query Optimization for Query Plans of Heterogeneous Multi-Database Systems," IEEE Transactions on Knowledge and Data Engineering, 1999.

⁴ All the examples in the paper are in Chinese. We translate them in to English for international readers' convenience.

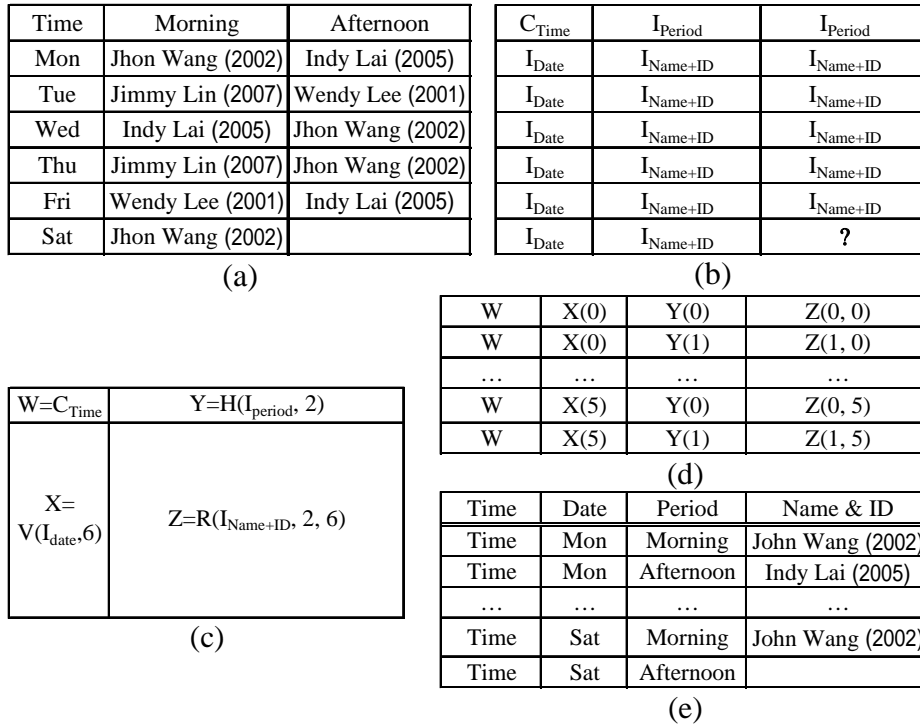


Figure 11. (a) The original table, (b) The outcome after the step 3.1, (c) 4 monotonic areas, (d)(e) Transformed from Fig. 11(a) by applying the transformation rules in section 3.3

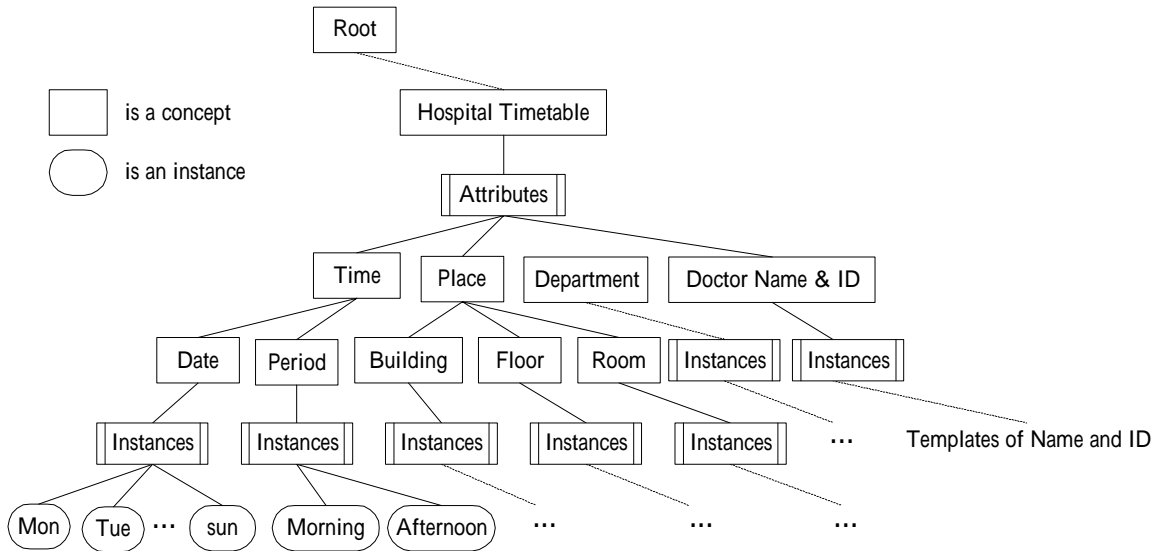


Figure 12. Ontology for the hospital timetables