Semi-Structured Data Extraction from Heterogeneous Sources

Xiaoying Gao and Leon Sterling
Intelligent Agent Laboratory
Department of Computer Science and Software Engineering
The University of Melbourne,
Parkville 3052, Australia
{xga, leon}@cs.mu.oz.au

Abstract

This paper concerns the extraction of semi-structured data from Web pages generated from multiple on-line services. This task is addressed by representing the schemas for semi-structured data and crafting generic wrappers based on the schemas. We introduce a hybrid representation method for schemas of semi-structured data, consisting of a concept hierarchy and a set of knowledge unit frames. A content-based and structure-bounded information extraction algorithm is developed to build the generic wrapper, which utilizes the schemas and takes advantage of the semi-structured page layouts. The main advantages of the system are that a single wrapper can be applied to multiple Web sites, and the wrapper can handle resources with missing data and data presented in free texts, which can not be wrapped by existing techniques. The hybrid representation has been used for writing schemas for seven domains. Experiments in two domains, on-line real estate advertisements and car advertisements, show that the generic wrapper is robust for many flexible data presentations and page structures.

Keywords: Information extraction from the Web; Information agents.

INTRODUCTION

Information extraction from Web pages has become a key research topic nowadays. It has great potential in a variety of research areas such as intelligent information agents, content-based information retrieval, knowledge discovery, and decision making. The task addressed in this paper is semi-structured data extraction from heterogeneous, semi-structured documents such as Web pages generated from online services. One example of the task is extracting important data such as price and size from online classified advertisements.

Previous work on scaleable agents (Doorenbos et al, 1997) and wrapper generation (Kushmerick et al, 1997; Ashish and Knoblock, 1997) has successfully applied machine learning technology to semi-structured data extraction. However, these approaches depend on a strict format for data presentation and rely heavily on HTML. In Doorenbos et al (1997), data are assumed to be presented in a uniform format and related data are presented in one single line. The algorithm introduced in Kushmerick et al (1997) works on tabular pages where data are presented in a fixed order without any missing items. The system by Ashish (Ashish and Knoblock, 1997) works on pages organized hierarchically. Semi-structured text (an example is shown in Figure 1 and the extracted results are given in Figure 2) is beyond the scope of such systems.

```
HOUSES & UNITS <BR>
RESERVOIR<BR>
<B>1 BR</B> SC flat, fully furnished, $110 pw <BR>
<B>2 BR</B> unit, BIR's, lnge, gas appl, $135 pw
<BR>
Ph 9478 2222<BR>
```

Figure 1. Fragment of a Web Page

```
Real estate ad #12
{       Property #10 {Suburb #3 "reservoir"
                 Size #4   "1"
                 Type #5 "flat"
                 Price #6 "110"}
        Property #11 {Suburb #3 "reservoir"
                 Size #7 "2"
                 Type #8 "unit"
                 Price #9 "135"}
}
```

Figure 2. The Extracted Results

In previous work on wrapper generation, including manual (Hammer et al, 1997), semi-automatic (Ashish and Knoblock, 1997; Adelberg, 1998), and automatic (Kushmerick et al, 1997; Doorenbos et al, 1997) methods, the wrapper generated is site specific, that is, an individual wrapper has to be built for each Web site.

The goal of our research is to build a generic wrapper which is flexible and robust with respect to data presentations and page structures, and thus can work on multiple Web sites. Our research proceeds in two steps:

- Designing a representation method for describing schemas of semi-structured data.
- Developing a generic information extraction algorithm based on the schemas.

The rest of this paper presents the task specification, followed by the schema representation method and the information extraction algorithm. Experimental results are presented next, followed by comparison with related work. The final section concludes this paper and discusses future work.

TASK SPECIFICATION: KNOWLEDGE UNITS EXTRACTION AND GROUPING

The task addressed in this paper is semi-structured data extraction from the output pages of online services. Figure 1 is a fragment of a semi-structured document taken from online real estate advertisements and Figure 2 shows the extracted results in the Object Exchange Model (OEM) (Abiteboul, 1997). Based on an examination of the characteristics of semi-structured documents, and the consideration of data models for semi-structured data such as OEM, we split information extraction into two sub tasks. The first task is to extract the knowledge units, called simple objects in OEM. The second task is to group the knowledge units into concepts, called complex objects in OEM.

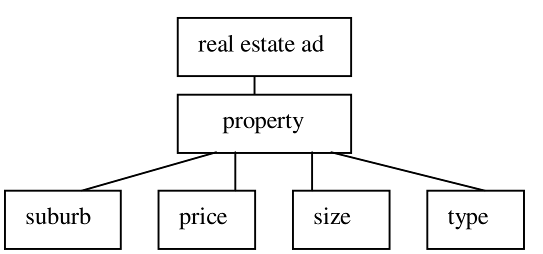
We define *knowledge units* as the smallest concepts, the lowest level concepts in the concept hierarchy. The names and values of knowledge units are the basic and important data to extract. For example, in the real estate advertisement domain, “suburb”, “price”, “size”, “type” are knowledge units. The task is to extract their values, group them to comprise the concept “property”, and then group a set of properties to the concept “real estate ad”.

SCHEMA REPRESENTATION: CONCEPT HIERARCHY AND KNOWLEDGE UNIT FRAMES

Despite the diversity of data structures, regularities can be found in concept organisations and concept descriptions. The discovered regularities can be represented as schemas. It is true that page structures and data presentations differ from page to page and site to site, and they can even be inconsistent within one page. However, ignoring the HTML tags, on examining the semi-structured text, that is, the nongrammatical and telegraphic text, there are regularities. For example, an advertisement for a real estate property usually has suburb, price, size, and type information. The price for a rental real estate property often starts with a “$”, then a number, and ends with a unit such as “per week” or “per month”. These regularities are important for guiding information extraction, and they are consistent across Web sites so long as the sites are in the same domain, so that they can be regarded as the schemas for semi-structured data. We introduce a hybrid representation method for the schemas of semi-structured data, in which schemas are represented as a concept hierarchy and a set of knowledge unit frames.

Concept Hierarchy

The concept hierarchy is regarded as a simple domain ontology, which is a concept net with a single link relationship “consists of” to connect the concepts. The lowest level concepts are knowledge units. The concept hierarchy for the real estate advertisement domain and its internal representation is shown in Figure 3.



sets_of(“real estate ad”, “property”)

a_set _of(“property”, [“suburb”, “price”, “size”, “type”])

Figure 3. Concept Hierarchy for Real Estate and its Representation

We use three predicates to specify the "consists of" relationship and to describe the concept hierarchy. They are

- sets_of (Concept1, Concept2): Concept1 consists of one or more Concept2.
- a_set_of (Concept1, ConceptList): Concept1 contains elements from ConceptList.
- a_sequence_of (Concept1, ConceptList) : Concept1 contains elements from ConceptList, in the same order as they appear in ConceptList.

<u>Knowledge Unit Frames</u>

We have adapted a frame-based representation for knowledge units which are the lowest level concepts in the concept hierarchy. Each knowledge unit is represented as a set of frames. Each frame represents a regular pattern. Each regular pattern consists of a set of lexicon items such as word, punctuation, tag, etc. The syntax of knowledge unit frames is shown in Figure 4.

Knowledge Unit Frame NO.
  No of lexicon items: *integer*
  Value extraction function: *a function*
  Certainty factor: *0~1*

Sub Frame No.
  Style: *one of {tag, database, char, word, phrase, number, string}*
  Instances: *string list*
  Exceptions: *string list*
 Pattern No.
   pattern : *a function*
   Certainty factor: *0~1*
  Max length: *integer*
  Min length: *integer*
  Max: *integer*
  Min: *integer*

Figure 4. The Syntax of Knowledge Unit Frames

Each frame has 3 slots and a set of sub frames.

- The number of lexicon items defines the number of sub frames included.
- The value extraction function defines how to extract the value of the knowledge unit. For knowledge units with values in different units such as "$800 p/month" and "$200 pwe", their values are converted to the default unit.
- A certainty factor is introduced to provide a criterion for choosing between frames.

Each sub frame has between 2 and 8 slots.

- The Style slot specifies the data type of the lexicon item. It should be pointed out that one of the lexicon styles is "tag", which means HTML tags can be used to define lexicons in our system, but they are not the unique or major components. Lexical items in the "database" type are linked to a built in database such as a "suburb" database.
- The Instances slot contains positive keywords[1] in a list.
- The Exceptions slot contain negative keywords[2] represented in a list.
- The lexicon extraction pattern slot actually has two sub slots: one is a pattern such as any_number(X) for extracting "123", "5,000", etc.; the other is a certainty factor to specify the priority of the pattern.
- The Max length and Min length slots specify string length constraints.
- The Min and Max slots specify range constraints for number lexicons.

---

[1] Positive keywords are the trigger words and keys for a concept.
[2] Negative keywords are the words which suggest that the concept is not present.

Towards Semi-Automatic Schema Generation

Handcrafting schemas is time consuming. We have explored methods for automatically generating some knowledge unit frames. Knowledge units are classified into two groups. One is "database" type such as "suburb", "car make" and "car model". The second is "pattern" type such as "price", "size", "year of manufacture". We adapted different learning strategies for the two types of knowledge units.

The representation for the "database" type of knowledge units, particularly the database that the lexicon is linked to, can be learned from structured Web pages. Since a lot of similar services are available online, it is not difficult to find one which is on the same domain but presents data as highly structured tabular data. By handcrafting wrappers for this structured site, values of the knowledge unit can be extracted to generate the database.

The "pattern" type of knowledge units can also be learned from structured pages. But since the patterns for knowledge units usually are consistent on one site, it is not efficient for browsing through one site and only learning one pattern. Therefore we prefer to ask users to give examples. The user gives as many examples as he/she can, the more diverse the better. Some rules can be learned from these examples. The learned rules are translated to frames.

Currently, human interaction is needed to confirm, reject or modify the learned frames. The value extraction function needs to be manually added. Human interaction is also needed to adjust the order of frames. The learning program puts the knowledge unit frames in alphabetical order. The order is changed to reflect the priority, that is, commonly used and easily identified frames should be put at the top.

THE GENERIC WRAPPER: CONTENT-BASED AND STRUCTURE-BOUNDED WRAPPING

Even though semi-structured data are not presented in a uniform format, a generic wrapper can be built by taking advantage of the semi-structured page layout together with the schemas. This is based on the following analysis:

- Knowledge units can be identified by their schemas in frames.
- One concept can be briefly distinguished from another by grouping the knowledge units according to the concept hierarchy although the concept can be incomplete (some knowledge units are missing) or can have extra information.
- The knowledge units in a concept may appear in a certain order. If not, the closer the knowledge units are, the more likely they belong to the same concept. For example, the "price", "size" in the same line (or paragraph) usually belong to the same "property".
- Knowledge units appearing in a title may belong to all of the concepts in the following section.

We developed a content-based and structure-bounded information extraction algorithm to build the generic wrapper. This section starts with an introduction of the information extractor and the page representation, and then details the information extraction algorithm.

Schemas Translated to Information Extractors in DCGs

Definite Clause Grammar rules (DCGs) are special Prolog rules (Sterling and Shapiro, 1994). Research work has shown that DCGs are very good for pattern matching (Bonnet and Bressan, 1997; Taylor, 1995). A simple example DCG rule for extracting price from "$200 pw" is:

```
rule("price", X)→ ["$"], any_number(X), [" "], ["p", "w"].
```

In our work, a generic knowledge unit extractor is crafted which dynamically translates knowledge unit frames to DCG rules. Each frame is translated to one rule. For example, if a frame has 4 sub frames, then the DCG rule body has 4 parts. Each part extracts one lexical item. When all parts in the body succeed, a knowledge unit is successfully identified, and then the value extraction function is called to get the value for the knowledge unit.

The concept extractor consists of 23 DCG rules to deal with the 3 predicates used for representing the concept hierarchy. The 23 rules are put in three groups with different priorities. Traditional DCGs are generally good at parsing ordered, continuous and complete patterns. Our rules, as an extension of DCGs, can handle free order, partial and incomplete structures. Weights are introduced to handle incomplete concepts. Thresholds are given for the sum of the weights of matching knowledge units to distinguish concepts, partial concepts and non concepts. Rules are also designed to deal with the possibilities of knowledge unit inheritance across a set of concepts.

Pages Represented as Structure Boundaries and Content Groups

We represent a Web page as a list with two types of elements: structure boundaries and content groups.

Each structure boundary is represented as

`b(Structure)`

We regard four structure boundaries as important for the task: word, line, paragraph and title. An HTML tag recognizer is designed to classify the tags into these 4 groups. All other tags not classified are grouped to "other" and represented as b(other). For text not in HTML or predefined text within <pre> and </pre>, specific labels such as space, <CR> and blank lines etc. are used as structure boundaries.

Each content group is represented as:

`g(ConceptName,Type,SubConceptList, ConceptIDList)`

There are five types of content groups:

- ku :knowledge units
- concept
- parts: part of a concept
- part_sets: a group of similar parts of a concept
- concept_set: a set of concepts

Some examples are:

g("suburb", ku, ["suburb"], [1]) ; a single knowledge unit "suburb"

g("property", parts, ["prices", "sizes"], [2,3]). ; part of "property" consisting of two knowledge units "price" and "size"

The Information Extraction Algorithm

The algorithm is content-based because the knowledge unit extraction is based on its frames and the grouping is based on the concept hierarchy. It is also structure-bounded because structure boundaries are used to guide the grouping. When information extraction is in progress, the page structure boundaries are deleted step by step from low level "word" to higher level "title", and the small content groups are grouped together to form bigger content groups according to the concept hierarchy. While the page representation becomes more and more abstracted, new concepts are extracted one by one.

The information extraction algorithm is detailed in three steps:

1. Page segmentation
   - A Web Page is read in as a character list.
   - The list is rewritten using two tokens: tag and text.
   - HTML tags are classified to b(word), b(line), b(paragraph) and b(title)
   - Text and b(word) are grouped to "text line"

2. Knowledge unit extraction
   - Knowledge unit frames are translated to information extractor in DCGs.
   - Each "text line" is parsed character by character using DCGs to extract knowledge units.
   - Knowledge units with low certainty factor are refined.
   - Repeated Knowledge units are deleted.

3. Knowledge unit grouping
   - The page is represented as a list consisting of structure boundaries and content groups.
   - The page representation is modified by deleting structure boundaries step by step, so content groups in different page structure levels (line, paragraph, and paragraph with title) become adjacent elements in the list.
   - For each page structure level, for each concept, the concept extractor, which consists of 23 rules, is recursively called.
   - Extracted concepts are saved in OEM format.

The information extraction process for an example page (page shown in Figure 1, results shown in Figure 2) is given in Figure 5.

1) The page fragment

[HOUSES & UNITS
<P>RESERVOIR
<B>1 BR </B>SC flat, fully furnished, $110 pw
<B>2 BR </B>unit, BIR's, lnge, gas appl, 135 pw
Ph 9478 2222
</P>]

2) Page segmentation and knowledge unit extraction (Step 1 & Step 2 of the algorithm)

[g("type",ku,["type"],[1]), g("type",ku,["type"],[2]), b(line),
b(para), g("suburb", ku, ["suburb"], [3]), b(line),
b(word), g("size", ku, ["size"], [4]), b(word), g("type", ku, ["type"], [5]), g("price", ku, ["price"], [6]), b(line),
b(word), g("size", ku, ["size"], [7]), b(word), g("type", ku, ["type"], [8]), g("price", ku, ["price"], [9]), b(line),
b(line), b(para)]

3) Word boundaries are deleted, then the concept extractor is called (Rule 3 is triggered)

[g("type",ku,["type"],[1]), g("type",ku,["type"],[2]), b(line),
b(para), g("suburb", ku, ["suburb"], [3]), b(line),
g("property", part, ["size", "type","price"], [4,5,6]), b(line),
g("property", part, ["size", "type", "price"], [7, 8, 9]), b(line),
b(line), b(para)]

4) Line boundaries are deleted, then the concept extractor is called (Rule 4 and Rule 11 are triggered)

[g("type",ku,["type"],[1]), g("type",ku,["type"],[2]),
b(para), g("suburb", ku, ["suburb"], [3]),
g("property", part_sets, ["size", "type","price"], [[4,5,6], [7,8, 9]]),
b(para)]

[g("type",ku,["type"],[1]), g("type",ku,["type"],[2]),
b(para),
g("property", part_sets, ["suburb""size", "type","price"], [[3,4,5,6], [3, 7, 8, 9]),
b(para)]

5) Paragraph boundaries are deleted, then the concept extractor is called (Rule 17 and Rule 23 are triggered)

[g("type",ku,["type"],[1]), g("type",ku,["type"],[2]),
g("property", concept_set, ["suburb""size", "type","price"], [10, 11]).

[g("type",ku,["type"],[1]), g("type",ku,["type"],[2]),
g("real estate ad", concept, ["property", "property"], [12])]

Figure 5. Information Extraction from an Example Page

EXPERIMENTAL RESULTS

Our schema representation has been used to write schemas for seven domains (Sterling, 1998), including sports scores, university information, corporate mergers, infectious diseases, bushfires, port information for shipping movement, and classified advertisements. For example, the schema for real estate advertisements contains 6 concepts represented as 11 predicates, 4 major knowledge units in 31 frames and a "suburb" database. The schema for car advertisements has 3 concepts represented in 5 predicates, 9 knowledge units in 24 frames, a "car make" database and a "car model" database. The schema representation has proven to be:

- Expressive. The representation of lexical items combines keywords, patterns, and also many other constraints such as string length.
- Simple. Only three predicates and a set of frames are sufficient for schema representation. Concepts are easily put in a hierarchy. Even very complex concepts can be represented by recursively using the three predicates.

- Open. All predicates and frames are independent, and can be easily added and deleted. The schema does not have to be complete. It can be tailored to suit a user's specific interests. For example, if the user is only interested in a "Ford" car, representation of other car makes is not essential.
- Reusable. Some concepts and knowledge units can be shared across domains.

The generic wrapper is tested on Web pages downloaded from 7 real estate advertisement sites (No.1~7 in table 1) and 12 car advertisement sites (No. 8~19 in table 1). These web sites come from two countries: Australia and the United States. Half of the Web sites (No 10~19) are chosen from an independent index LookSmart, where No.10~14 are the top 5 hits in LookSmart index on worldwide automobile market and No.11~19 are the top five hits in LookSmart index on Australian automobile market. These Web sites represent a reasonable diversity of page structures, including pages presenting concepts with missing knowledge units, knowledge units in free order, knowledge units in different metric units, and knowledge units written in free text. Our system is able to extract data from 18 sites. The only one site failed (site No.14) presents data in images.

No.	Resources	URL
1	NewsClassifieds Australia : real estate	http://www.newsclassifieds.com/
2	Fairfax@Market: real estate	http://market.fairfax.com.au/
3	Australian Real Estate Online	http://www.property.com.au/
4	Melbourne Trading Post: real estate	http://www.tradingpost.com.au
5	VUT University: available accommodation	http://www.vut.edu.au/~sth/house.html
6	Seattle Times Classified Ads: rentals	http://www.seatimes.com/classified/rent/
7	Los Angeles Times Classifieds: rentals	http://www.latimes.com/HOME/CLASS/REALEST/ RENTAL/
8	Yahoo Classifieds: Autos and Motorcycles	http://classifieds.yahoo.com/automobiles.html
9	Infoseek Classifieds	http://www.classifieds2000.com/cgi-cls/display.exe? partner=Infoseek&path=Auto~Car~Search
10	Auto Trader Online	http://vision.traderonline.com/auto/index.shtml
11	AutoConnect	http://209.143.231.6/
12	AutoWeb	http://www.autoweb.com/
13	Calling All Cars	http://www.cacars.com/
14	Car talk	http://cartalk.cars.com/
15	AutoWeb	http://www.autoweb.com.au/
16	CarSales	http://www.carsales.com.au/
17	Melbourne Trading Post: cars	http://www.tradingpost.com.au/
18	Fairfax@Market: motors	http://market.fairfax.com.au/
19	Fowles Auction Group	http://www.fowles.com.au/

Table 1: Web Sites and their URL

We started the experiment in the real estate advertisement domain and then adapted it to the car advertisement domain. The schema for real estate was built based on site 1,2,3,4,5 and the schema for cars was built based on site 8, 9, 18. From the experiment, it was found that the completeness and accuracy of the schema influenced the information extraction performance. After we refined the schemas based on all the web sites, the performance was dramatically improved. Future work is needed for exploring methods on automatic schema extension.

Precision and *Recall* are used to evaluate the information extraction performance. Each concept is considered correct when all knowledge units at the lower level are correct. The performance on concept extraction (overall performance of step 1,2 &3) reflects both knowledge unit extraction (Step 2) and knowledge unit grouping (Step 3). *Precision* and *Recall* are used to evaluate system performance on both knowledge unit extraction and grouping. Each knowledge unit is correct if its name, value and location are the same as that of the manually generated answers. We define a knowledge unit group as being correct when the correct knowledge units have been put in the right concept, ignoring false positive or false negative knowledge units.

The Web sites are classified into two groups: one is tabular pages (site No. 8,11,12,15,16), where knowledge units are items in tables; the other is "semi-structured pages" (the remaining 13 sites), in which knowledge units are not separated as items in tables or lists. For tabular pages, the performance on knowledge unit grouping is always perfect. Both precision and recall for knowledge units and concepts can reach 100% after the schemas are refined. For semi-structured pages, the performance varies according to the flexibility and complexity of data presentation. The results are shown in Table 2 (row 1). In particular, for two typical semi-structured Web sites (site No. 1, 2), more detailed experiments are carried out on a large data corpus. The results are also given in table 2 (row 2, 3). These two sites collect advertisements from many different newspapers and the advertisements are written in free text. To our knowledge, no previous wrapping technology is able to extract data from these sites.

Web sites	Knowledge units extraction (Step 2)			Knowledge unit grouping (Step 3)			Concept extraction (Overall performance)		
	Total No.	Precision	Recall	Total No.	Precision	Recall	Total No.	Precision	Recall
Semi-structured pages	1454	90~99	76~100	348	84*~100	80~100	348	56~97	56~97
NewClassifieds (Site No. 1)	1082	95	95	312	75*	76	312	69	70
Fairfax@Market (Site No.2)	353	93	93	94	79	78	94	70	70

* Row 1 and row 2 are results on different data corpus, that is why "75" is not in the range of "84~100". It is the same case with recall for knowledge unit grouping.

Table 2: Information Extraction Results

RELATED WORK

In previous research on semi-structured data, schemas are represented as labeled graphs (Quass et al, 1996; Buneman et al, 1997; Calvanese et al, 1998). We introduced a hybrid representation scheme, in which concepts are distinguished from knowledge units and they are represented separately as a concept hierarchy and a set of frames. Previous research is from a database point of view and focuses on data models and query languages (Abiteboul, 1997; Buneman, 1997). Our approach addresses the fundamental problem of identifying and extracting database records. Our schema representation has been successfully used to address the task of extracting records.

Two systems are similar to ours regarding the methods of knowledge units representation and extraction. InfoExtractor (Smith and Lopez, 1997) uses frames to represent all concepts and the frames are used to guide concept extraction. The concepts are linked hierarchically by using high level concepts as the context of low level concepts. Documents are abstracted to one or more evaluation contexts such as header, body, and summary. Each concept frame consists of slots to specify the concept's evaluation context, identification pattern, and identification keywords. Concept extraction is mainly through approximate keywords and terms matching. The concept frames are handcrafted. There are no details given on the experimental results. The other is research on entities extraction (Nado and Huffman, 1997; Huffman and Baudin, 1997), in which heuristic knowledge is represented and used for identifying entities such as person names, company names, etc.

A survey paper by Ion Muslea (Muslea, 1999) summaries research on using machine learning for information extraction. STALKER (Muslea, 1999) is a wrapper induction system which uses *Embedded Catalog Tree* formalism to describe the hierarchy of documents, and generates wrappers for hierarchical organized HTML documents. SoftMealy (Hsu, 1998) is another wrapper induction system that represents wrappers as finite-state transducers, which is able to work on documents in different formatting conventions and to extract items in various orderings. There is also research on adapting machine learning technology for information extraction from natural languages to semi-structured data extraction. Some typical examples are Soderland (1998), Soderland (1997) and Freitag (1998). In these three systems, the extraction rules automatically learned can work on flexible data presentations and multiple Web sites. Differing from these machine learning approaches, our system is a knowledge-based approach and it focuses on knowledge representation rather than induction, training or natural language processing.

Some interactive wrapper generation toolkits such as W4F (Sahuguet, 1999) and NoDoSE (Adelberg, 1998) have become available recently. They provide GUI interfaces where humans can interact with the system and guide the wrapper generation. Research on adding machine-readable semantics to HTML, such as DOM (W3C, 1998) which is a key component of the XML family of standards, and SHOE (Luke, 1997), shows the potential of formalising the data presentation format on the WWW. These two approaches rely on the WWW authors to add semantic tags to their pages. There are also some large scale projects on information integration such as TSIMMIS (Molina, 1995), information manifold (Kirk, 1995), SIMS (Arens, 1996) and JEDI (Huck, 1998). These projects focus on data model, query and extraction languages. Our research is from a different point of view, which aims to recognise and extract concepts from normal Web pages based on knowledge and heuristics, and our system is smaller in scale. It would be interesting to see how our system can be integrated with some of the systems as a case study.

CONCLUSIONS AND FUTURE WORK

This paper addresses semi-structured data extraction by representing schemas and designing generic wrappers based on schemas. We introduce a hybrid representation method for the schemas of semi-structured data, in which a schema can be represented as a concept hierarchy and a set of knowledge unit frames. This uniform representation enables automatic generation of some of the frames. We develop a content-based and structure-bounded information extraction algorithm to build a generic wrapper, which utilizes the schemas and takes advantages of the page structures. The hybrid schema representation has been used to write schemas for seven domains. Experiments in two domains show that the generic wrapper is robust on flexible data presentations and page structures, and thus can work on multiple Web sites.

Comparing with previous research on semi-structured data extraction (Doorenbos et al, 1997; Kushmerick et al, 1997; Ashish and Knoblock 1997), our approach has the following advantages:

- The wrapper is Web site independent. The content-based and structure-bounded information extraction algorithm is able to work on multiple Web sites by taking advantage of both the schemas and the semi-structured page layouts.
- It gives no strong bias as to data presentation. It handles concepts with knowledge units which are missing, in free order, in different metric units or even written in free text. The only assumption on data presentation is that one knowledge unit can be extracted in a single line. This is usually true since we define knowledge units as small concepts. Also this can be easily changed to "paragraph" for domains with long knowledge units.
- It utilizes HTML for segmentation but it does not heavily rely on HTML. Knowledge units are extracted based on an analysis of the text rather than the text format. The tag classification does not have to be perfect because the knowledge unit grouping is through the concept extractor which utilizes the concept hierarchy. The system can extract data from plain text.

One limitation of our method is that the system needs a domain specific schema, and so instead of being Web site specific, the wrapper is domain specific. A new schema needs to be generated for adapting our system to a new domain. We have addressed this problem by exploring methods for automatic schema generation.

Future work is needed to address the following topics:

- Testing the system on more domains and more Web sites.
- Exploring methods towards schema automatic generation and extension. For example, designing algorithms for detecting regular expressions; Another issue is to adapt a schema to a new cultural environment. For example, to automatically adjust the value extraction function for price in dollars and price in francs.
- Automatically detecting extraction mistakes. Assuming results from one site are similar, mistakes can be found by comparing the results to look for irregularity and abnormalities.
- Considering other concept relationships such as "not consist of" and testing how they can influence information extraction performance.

ACKNOWLEDGEMENTS

We would like to thank Liz Sonenberg for helpful comments, and thank Seng Wai Loke for writing schemas for a number of application domains. The first author is supported by an AusAid scholarship and the ARC has provided partial support through its small grant scheme.

REFERENCES

Abiteboul, S. 1997. "Querying Semi-Structured Data". In Proceedings of the International Conference on Database Theory, 1-18. Delphi, Greece.

Adelberg, B. 1998. "NoDoSE- A Tool for Semi-Automatically Extracting Structured and Semistructured Data from Text Documents". *SIGMOD RECORD*, 27(2): 283-294.

Arens Y., Hsu C. N., and. Knoblock C. A, (1996) "Query Processing in the SIMS Information Mediator", in *Advanced Planning Technology*, A. Tate, Ed. Menlo Park, CA: AAAI Press.

Ashish, N., and Knoblock, C. A. 1997. "Semi-automatic Wrapper Generation for Internet Information Sources". In Proceedings of Cooperative Information Systems.

Bonnet, P., and Bressan, S. 1997. "Extraction and Integration of Data from Semi-structured Documents into Business Applications". In Proceedings of Industrial Applications of Prolog (INAP), Kobe, Japan.

Buneman, P., 1997. "Semistructured Data". In Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Tucson, Arizona.

Buneman, P.; Davidson, S.; Fernandez, M.; and Suciu D. "Adding Structure to Unstructured Data". In proceedings of ICDT, 336-350. Delphi, Greece.

Calvanese, D.; Giacomo, G. D.; and Lenzerini, M. 1998. "What can Knowledge Representation Do for Semi-Structured Data?". AAAI-98, 205-210.

Doorenbos, R. B.; Etzioni, O.; and Weld, D. S. 1997. "A Scalable Comparison-Shopping Agent". Agent' 97.

Freitag, D. 1998. "Information Extraction from HTML: Application of a General Machine Learning Approach". AAAI-98.

Hammer, J.; Garcia-Molina, H.; Cho, J.; Aranha, R.; and Crespo, A. 1997. "Extracting Semistructured information from the Web". Workshop on Management of Semi-structured Data, Tuscon, Arizona.

Huck G., Fankhauser P., Aberer K., and Neuhold E., (1998) "Jedi: Extracting and Synthesizing Information from the Web", presented at the 3rd IFCIS international Conference on Cooperative Information Systems, COOPIS'98, New York, USA.

Huffman, S. B., and Baudin, C. 1997. "Toward Structured Retrieval in Semi-structured Information Spaces". IJCAI-97, 751-756. Nagaya, Japan.

Kirk T., Levy A. Y., Sagiv Y., and Srivastava D., (1995) "The Information Manifold", in Working Notes of the AAAI Spring Symposium Series on Information Gathering from Distributed, Heterogeneous Environment.

Kushmerick, N.; Weld, D. S.; and Doorenbos, R. 1997. "Wrapper Induction for Information Extraction". IJCAI-97, 729-735. Nagoya, Japan.

Luke S., Spector L., Rager D., and Hendler J., (1997), "Ontology-based Web Agents", the First International Conference on Autonomous Agents, AA-97.

Molina H. G., Hammer J., Ireland K., Papakonstantinou Y., Ullman J., and Widom J., (1995), "Integrating and Accessing Heterogeneous Information Source in TSIMMIS", in Working Notes of the AAAI Spring Symposium: Information gathering from Heterogeneous, distributed environments, Stanford University.

Muslea I., (1999) "Extraction Patterns for Information Extraction Tasks: A Survey", presented at The AAAI-99 Workshop on Machine Learning for Information Extraction.

Muslea I., Minton S., and Knoblock C., (1999), “A Hierarchical Approach to Wrapper Induction”, presented at 3rd conference on Autonomous Agents.

Nado, R. A., and Huffman, S. B. 1997. “Extracting Entity Profiles from Semi-structured Information Spaces”. Workshop on Management of Semi-Structured Data, in conjunction with PODS/ SIGMOD, Tucson, Arizona.

Quass D. And et. al. “LORE: A Lightweight Object Repository for Semistructured Data”. In Proceedings of the ACM SIGMOD international Conference on Management of Data, Montreal, Canada.

Sahuguet A. and Azavant F., (1999), “WysiWyg Web Wrapper Factory (W4F)”, submitted to WWW8.

Hsu C.-N., (1998), “Initial Results on Wrapping Semistructured Web pages with Finite-State Transducers and Contextual Rules”, presented at AAAI'98 Workshop on AI and Information Integration.

Smith, D., and Lopez, M. 1997. “Information Extraction for Semi-Structured Documents”. Workshop on Management of Semi-Structured Data, in conjunction with PODS/ SIGMOD, Tucson, Arizona.

Soderland, S. 1998. “Learning Information Extraction Rules for Semi-structured and Free Text”, draft, http://www.cs.washington.edu/homes/soderlan/.

Soderland, S. 1997. “Learning to Extract Text-based Information from the World Wide Web.” In Proceedings of Third International Conference on Knowledge Discovery and Data Mining (KDD-97).

Sterling, L., “Surveying and Reviewing Methods of Representing Scenarios”, a Report to the Information Technology Division DSTO, Salisbury, SA.

Sterling, L., and Shapiro, E. (1994). *The Art of Prolog*, The MIT Press.

Taylor, A. 1995. “Using Prolog for Biological Descriptions”. In Proceedings of The third international Conference on the Practical Application of Prolog, 587-597. Paris, France.

W3C, (1998) "The Document Object Model", *http://www.w3.org/DOM*.