Knowledge-based Information Agents

Xiaoying Gao[1] and Leon Sterling[2]

[1] School of Mathematical and Computing Sciences
Victoria University of Wellington
Wellington, New Zealand
xgao@mcs.vuw.ac.nz

[2] Department of Computer Science and Software Engineering
The University of Melbourne
Victoria, 3010, Australia
leon@cs.mu.oz.au

Abstract. This paper explains our approach to building knowledge-based information agents. Instead of building an agent from scratch, we advocate building an agent by adding knowledge to a framework. Knowledge is classified into three categories - general knowledge, domain specific knowledge and site specific knowledge - which enables knowledge reuse and sharing. The paper details the agent architecture, the components of each category of knowledge and the main functions of the framework.

1 Introduction

Information agents are intelligent pieces of software which can automatically search for information on the WWW [2] [10]. They usually deal with multiple Web sites in a single domain or multiple domains. One key step of building information agents is to extract information from multiple Web sites, that is, to transfer important information to structured data so that more accurate search can be carried out as querying a structured database.

The main challenge of building an information agent is how to make the agent scalable and adaptable. More and more online documents are becoming available and each has a different data format. The number of Web sites and their domains is huge and is growing very fast. Existing Web pages are being updated continuously, and their data formats may be modified at any time without any warning. While it might be easy to handcraft an information agent for one particular Web site in one specific domain for a particular time, how to update the Web site, how to adapt it and make it scalable to new Web sites and new domains, is a big challenge. There is an urgent need to develop methods and tools to ease agent generation and adaptation.

Recent research has used machine learning technology to build scalable agents [1] and to automatically learn information extraction patterns [6] [7] [9]. However, these systems work on relatively structured Web pages. The majority of Web pages with flexible data format, for example, data presented in free text and spread across sentences and paragraphs, are out of reach of current automatic systems.

Our research introduces a knowledge-based approach to support the generation and adaptation of information agents. We view an information agent as a knowledge-based system. The knowledge for guiding information extraction, such as information extraction patterns, is saved in the knowledge base of the agent. The information extraction process is coded as an inference engine. We assume the knowledge can be separated from the information extraction process. Instead of building an agent from scratch, an agent can be generated by adding knowledge bases to a reusable shell. An agent can be adapted to new domains and new Web sites by changing the knowledge bases. In slogan form,

Information Agent = Knowledge Bases + Agent Shell

We focus on building agents for information extraction from semi-structured data, that is data in an intermediate format between data in free text and structured data in databases. Typical examples are Web pages provided by online services such as classified advertisements, product categories, and telephone books. We believe that knowledge plays an important role for information extraction from semi-structured data, and information extraction from semi-structured data can be achieved based on a limited amount of knowledge with only simple natural language processing. Semi-structured data provides the right level of diversity and difficulty for testing our methods.

The rest of this paper is organized as follows. Section 2 discusses the knowledge that is useful for building agents and describes our classification of knowledge into three categories. Section 3 introduces the agent architecture. The two main parts of an agent, the knowledge bases and the information extraction engine are discussed in Sections 4 and 5 respectively. Section 6 gives some experimental results, while the final section concludes this paper.

2 Knowledge Classification

Focusing on information extraction from semi-structured data, we have examined thousands of Web pages. We summarize the knowledge that is useful for guiding the information extraction as follows. We classify knowledge into three categories: general knowledge, domain specific knowledge and site specific knowledge.

- General Knowledge. General knowledge is true for most online documents, if not for all of them, that is, the knowledge is both domain independent and site independent. Typical examples are the common usage of HTML tags, for example, what is a table, what is a paragraph, and what is a line.
- Domain Specific Knowledge. Domain specific knowledge is true in a particular domain. The knowledge is site independent, that is, the knowledge is consistent for most Web sites if not for all of them as long as the Web sites present data in the same domain. For example, in the real estate domain,

each property in an online advertisement has a suburb where it is located; the price for renting a property is usually denoted by a "$", followed by a number, and a unit such as "per week" or "per month". Domain specific knowledge is usually specified using terms in a specific domain and may not generalize to other domains.

- Site Specific Knowledge. Site specific knowledge is true for a particular site. To prevent the intersection with domain knowledge, we define the site specific knowledge being domain independent, that is, if the knowledge is true for this site and also true for this domain, then it is classified into domain knowledge. Site specific knowledge mainly consists of the site specific data formatting conventions, for example, in a particular Web site called NewsClassifieds, suburb names are printed in all capital letters. Site specific knowledge is tailored to a specific site and unlikely to be consistent with other sites.

This knowledge classification enables knowledge reuse and sharing, and also gives guidance for agent adaptation. General knowledge is completely reusable and can be shared for many information extraction tasks. Domain knowledge can be reused and shared for Web sites in the same domain. Site specific knowledge is limited to Web pages on the same site.[1] To adapt an agent to a new domain, new domain specific knowledge is needed. To adapt an agent to a new site, new site specific knowledge needs to be added.

3 Agent Architecture

The architecture of our knowledge-based information agent is shown in Figure 1. An agent contains three knowledge bases and a framework. The three knowledge bases are: General Knowledge Base (G), Domain Knowledge Base (D), and Site Specific Knowledge Base (S). The framework contains three main functions: Web Access Engine, Information Extraction Engine and Matcher.

We detail the three functions as follows:

- The Web Access Engine utilizes site specific knowledge to get access to the Internet and download Web pages. When the information source has a searchable interface, the Web Access Engine needs to use the original user query or the structured query (the IE results from the original query) to interact with the interface.
- The Information Extraction Engine uses general, domain specific and site specific knowledge to extract information from both the user query and Web pages, and to save them as structured data.

[1] We do not expect site specific knowledge can be shared by different domains on the same site, because the information to be extracted differs greatly from one domain to another. Actually, one site with different domains is regarded as different sites, that is, one site has only one specific domain. For example, the NewsClassifieds with real estate advertisements is one Web site and NewsClassifieds with car advertisements is another Web site.

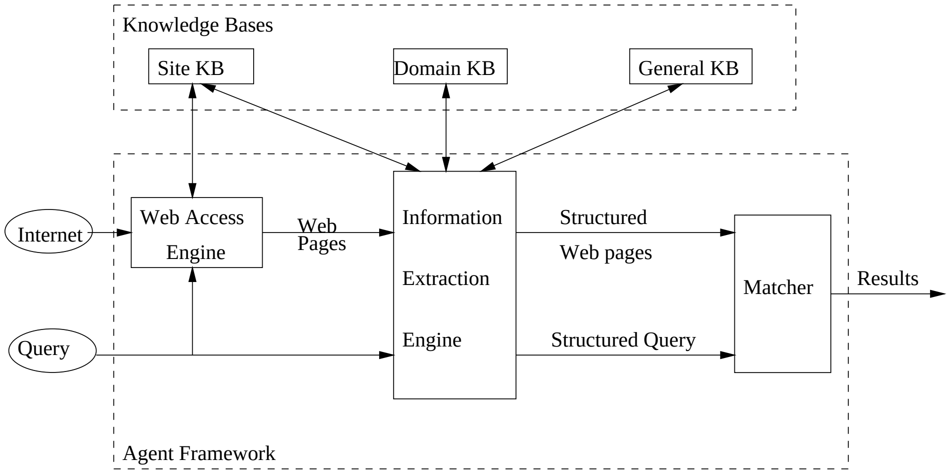

Fig. 1. The Architecture of Knowledge-Based Information Agents

- The Matcher matches the structured query with the structured data of Web pages and outputs the results.

An agent shell includes the framework and the general knowledge base. An information agent can be built by adding the domain knowledge base and the site specific knowledge base to the agent shell. An example of an agent shell is given in [8].

As we mentioned in the introduction, this paper focuses on how to use knowledge to extract information from Web page. The next two sections discuss the main components of three knowledge bases and one main function of the framework, the information extraction engine.

4 Three Knowledge Bases

Focusing on information extraction from semi-structured data, we summarize the main components of the three categories of knowledge as follows:

- General Knowledge (G)
  - The usage of HTML tags, particularly, the page structure levels (word, line, paragraph, page) the tags are linked to.
  - The identification methods of basic data types such as tag, text, character, etc.
  - Common sense knowledge such as related data are often presented together.
- Domain Specific Knowledge (D)
  - The concepts and the relationship of concepts. For example, in real estate domain, the concepts include “real estate ad”, “property”, “suburb”,

"price", "size", "type", etc. The concepts can be put in a hierarchy, for example, "real estate ad" consists of a number of "property", each property consists of "suburb", "price", "size", and "type". The concept in the last level (the atomic concepts) are called knowledge units (KU) in this paper.
  - How to identify the knowledge units (atomic concepts) and how to extract the value of knowledge units. Its major components include the domain specific terminology (for example, in the real estate domain, a suburb database can be used to identify the *Suburb* of each property from online advertisements), and domain specific data formatting conventions (for example, the *Price* for renting a property is usually a "$", followed by a number, and a unit such as "per week" or "per month")
- Site Specific Knowledge (S)
  - Site specific knowledge of the interface of each Web site. Most semi-structured data is presented as the search results of local search engines. In order to interact with the local search engine, the system needs site specific knowledge of the interface of the local search engine. For example, if the interface is an HTML form, the system needs to know the access method "Get" or "Post", and the way to generate query strings.
  - The information extraction patterns for fields (a group of knowledge units).
  - Site specific information extraction patterns for concepts.
  - Site specific usage of HTML tags
  - Site specific concept hierarchy
  - Site specific information extraction pattern for individual knowledge units.

    The site specific knowledge base does not have to be complete. All items except the first one are optional. Site specific knowledge is used for describing some special sites which can not be described by domain specific knowledge.

The three categories of knowledge have different priorities when they are used for information extraction. The priorities are given as follows:

1. Site specific knowledge (S)
2. Domain knowledge (D)
3. General knowledge (G)

During the information extraction process, the site specific knowledge has the highest priority and the general knowledge has the lowest. When there are conflicts between the knowledge, the higher priority knowledge overrides the lower priority knowledge. When we get a particular site, we search for site specific knowledge first. If some site specific knowledge is found, this knowledge is used instead of the associated knowledge in either the general knowledge base or domain knowledge base. For example, if a site specific information extraction pattern is found for a special knowledge unit, then this pattern is used for extracting the knowledge unit, instead of using the more general pattern in the domain knowledge base.

5 Information Extraction Engine

The information extraction engine utilizes the three categories of knowledge, extracts information from Web pages and saves the information as structured data. Figure 2 shows how it works.

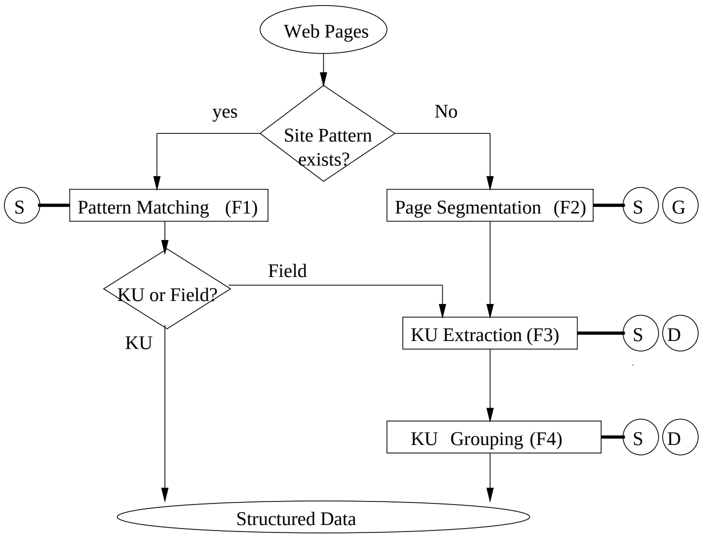

Fig. 2. The Information Extraction Engine

The input to the system is the source file of a Web page with its site name (the Web site where the page is downloaded from). The output is a number of concepts, each concept consists of a number of knowledge units. The system first checks whether there are site specific patterns 1) If yes, then the page is parsed through the pattern matching function. There are two kinds of output: a) if the output consists of concepts or knowledge units, they are directly saved to structured data. b) if the output consists of fields (each field contains one or more knowledge units), then the fields are used as the input to knowledge unit extraction function, and then through a knowledge unit grouping function, to be parsed to structured data and saved. 2) If no, the page is parsed through three functions: page segmentation, knowledge unit extraction and knowledge unit grouping.

The four functions and the categories of knowledge they use are detailed as follows:

- Pattern matching (F1): this function utilizes site specific knowledge, including site specific information extraction patterns for concepts, patterns for individual knowledge units and patterns for a group of knowledge units (fields), to parse a page into concepts, knowledge units and fields. The concepts and

knowledge units are then saved as structured data and the fields are passed to the next function.
- Page segmentation (F2): this function uses general knowledge of HTML tag usage and site specific knowledge of HTML usage, if available, to parse a page into "lines", which can be a line, a table row or an item of a list. It consists of four main steps: 1) representing a page as a character list, 2) rewriting the list using two tokens: tags and text, 3) classifying the tags into groups according to the page structure they represent including "word", "line", "paragraph", 4) segmenting a page into "lines".
- Knowledge unit extraction (F3): this function uses either domain knowledge of knowledge unit identification and extraction or site specific information extraction patterns for knowledge units, if available, to extract knowledge units from each "line".
- Knowledge unit grouping (F4): this function groups knowledge units into concepts and it uses two kinds of knowledge
  - either general knowledge of HTML usage or site specific knowledge, if available, of HTML usage AND
  - either domain knowledge of a concept hierarchy or site specific knowledge, if available, of a concept hierarchy.

6 Experimental Results

Our first information agent CASA (Classified Advertisement Search Agent) was built in 1997 [3] to search online real estate advertisements and help users to find rental property. It successfully searched for information automatically from multiple Web sites. It performed better than local search engines based on keyword matching.

An agent shell was developed based on the generalization of our first agent. The reusable agent framework, including the functions for Web accessing, information extraction and matching, forms the main part of the agent shell. The knowledge bases are completely separated from the framework and only the general knowledge base forms part of the agent shell. The other knowledge bases are kept separate from the shell. New information agents can be built by adding a new domain knowledge base and site specific knowledge base to the agent shell. The agent shell was successfully used to build a car classified advertisement search agent and a soccer score search agent [4].

Our experiments on building agents based on the framework show that:

- Our agent shell can be used to build information agents for multiple domains and multiple sites. Our agent can be easily adapted or extended by modifying or extending its knowledge bases, while most current information agents are tailored to one specific domain and are difficult to scale up.
- Our agents built using the agent shell accept user queries written in restricted natural languages, since the information extraction engine can extract specific requirements from the user query using the same method for extracting structured information from Web pages. The interface of our agents can be

as simple as that of keyword search engines with one single text input field. The interface is easy to generate and easy to use. The interface does not need to change for different domains. This differs from current local search engines, in which different user interfaces need to be designed for different domains.
– The information agents generated using our agent shell show better performance than local search engines based on keyword matching. The reason is that the information extraction engine transfers both the query and Web pages into structured data represented as a set of knowledge units. The matching is carried out between knowledge units which is more accurate than keyword matching.

In order to evaluate the agent's performance on information extraction from Web pages, we tested our agent on Web pages downloaded from over 100 Web sites. This paper will give some results based on our basic corpus. Our basic corpus was built by down-loading Web pages from 24 Web sites, 12 in the real estate advertisement domain and 12 in the car advertisement domain. Most of the Web sites are chosen from the top sites indexed by the search engine LookSmart at http://www.looksmart.com.

We use two parameters widely used in information extraction, precision and recall to evaluate our system. Precision is the percentage of correct responses out of all responses. Recall is the percentage of correct responses out of the total of correct answers. For each page, the information extraction answer keys are generated by manually correcting the output of our system. The performance of our system is evaluated by comparing the output with the answer keys.

In order to evaluate the performance of different steps of information extraction, we calculate precision and recall for the extraction of knowledge units, knowledge unit groups, and concepts.

– knowledge unit. Each knowledge unit is correct if its name and value are the same as that of the manually generated answers. The precision and recall of knowledge units indicate the ability of extracting individual knowledge units from Web pages.
– knowledge unit groups. We define a knowledge unit group as being correct when the correct knowledge units have been put in the right concept (group), ignoring false positive or false negative knowledge units. The precision and recall of knowledge unit groups indicate the ability of grouping knowledge units into concepts.
– Concept. A concept is considered correct when its all knowledge units at the lower levels are correct, that is, the concept is perfect, all of its knowledge units are extracted and all extracted knowledge units are correct. The precision and recall of concepts indicate the ability of extracting a "perfect concept".

The results are given in Table 1. The results show that our agent performs well on multiple Web sites, including Web sites with flexible data formats such as data presented as free text in paragraphs.

Table 1. Information Extraction Results

Domain	KU[a]		KU Groups		Concept	
	P[b]	R[c]	P	R	P	R
Reales[d]	87-100	76-100	75-100	75-100	66-100	60-100
Car[e]	92-100	95-100	94-100	97-100	56-100	56-100

[a] KU: Knowledge Unit
[b] P: Precision
[c] R: Recall
[d] Reales: Real estate advertisement
[e] Car: Car advertisement

7 Conclusion

This paper introduces a framework for building knowledge-based information agents. The knowledge-based approach that separates knowledge bases from other processes supports easy agent generation and adaptation. A new agent can be generated by adding new knowledge bases and an agent can be adapted by changing the knowledge bases. The classification of knowledge into three categories enables knowledge reuse and sharing. The general knowledge base is completely reusable and is built as part of the agent shell. The domain knowledge need to be changed for new domains and the site specific knowledge needs to be extended or learned for new Web sites.

This research focuses on semi-structured data and has been successful at extracting information from multiple Web sites in limited domains. With the rapid growth of the Internet, more and more services are become available online. Many of them present semi-structured data, for example, product catalogs, weather forecasts, phone books and stock market quotations. Our system is very useful for building information extraction systems for these online services. Users can generate their own information extraction system by creating knowledge bases and plugging them into our reusable shell. We believe this is much easier and faster than building a system from scratch.

We are currently working on how to learn part of the knowledge automatically. An algorithm is developed to learn site specific information extraction patterns from tabular Web pages [5]. Future work is needed to improve the learning techniques and to reduce manual work required for building and updating the knowledge bases.

References

1. Doorenbos, R. B., Etzioni, O.i , Weld, D. S.: A scalable comparison-shopping agent. In *Agent 97*, (1997)
2. Etzioni, O., Weld, D. S.: Intelligent agents on the internet: Fact, fiction, and forecast. *IEEE Expert*, 10 no. 4:44–49, (1995)

3. Gao, X., Sterling, L.: Classified advertisement search agent (CASA): A knowledge-based information agent for searching semi-structured text. In *The Practical Application of Intelligent Agents and Multi-Agent Technology*, pages 621–622, London, UK, March 23-25 (1998)
4. Gao, X., Sterling, L.: A methodology for building information agents. In Yun Yang, Minshu Li, and Allan Ellis, editors, *Web Technologies and Applications, Asia Pacific Web Conference (APWeb'98)*, chapter 5, pages 43–52. International Academic Publishers, (1998)
5. Gao, X., Sterling, L.: AutoWrapper: Automatic wrapper generation for multiple online services. In Gilbert H. Young, editor, *World Wide Web: Technologies and Applications for the New Millennium*, chapter 8, pages 61–70. C.S.R.E.A. Press, (2000)
6. Hsu, C.-N.: Initial results on wrapping semistructured web pages with finite-state transducers and contextual rules. In *AAAI'98 Workshop on AI and Information Integration.* (1998)
7. Kushmerick, N.: Wrapper induction: Efficiency and expressiveness. *Journal of Artificial Intelligence*, 118:15–68, (2000)
8. Loke, S. W., Sterling, L., Sonenberg, L., Kim, H.: Aris: A shell for information agents that exploit web site structure. In *PAAM'98*, pages 201–219, London, (1998)
9. Muslea, I., Minton, S., Knoblock, C.: A hierarchical approach to wrapper induction. In *The 3rd conference on Autonomous Agents(Agent'99)*, (1999)
10. Sterling, L.: On finding needles in WWW haystacks. In *the Tenth Australian Joint Conference on Artificial Intelligence(Lecture Notes in Artificial Intelligence 1342)*, pages 25–36, Perth, Australia, 30 November-4 December (1997)