An Agent Based System Framework for Mining Data Record Extraction from Search Engine Result Pages

P.Kalaivani M.E., (Ph.D)  
Research Scholar  
Department of CSE  
Sathyabama University  
Chennai  
India  
Vaniraja2001@yahoo.com

Dr.K.L Shunmuganathan M.E., Ph.D  
Professor & Head  
Department of CSE  
RMK Engineering College  
Chennai  
India

Abstract

Nowadays, the huge amount of information distributed through the Web motivates studying techniques to be adopted in order to extract relevant data in an efficient and reliable way. Information extraction (IE) from semi-structured Web documents plays an important role for a variety of information agents. In this paper, a framework of WebIE system with the help of the JADE platform is proposed to solve problems by non-visual automatic wrapper to extract data records from search engine results pages which contain important information for Meta search engine and computer users. It gives the idea about different agents used in WebIE and how the communication occurred between them and how to manage different agents. Multi Agent System (MAS) provides an efficient way for communicating agents and it is decentralized. Prototype model is developed for the study purpose and how it is used to solve the complex problems arise into the WebIE. Our wrapper consists of a series of agent filter to detect and remove irrelevant data region from the web page. In this paper, we propose a highly effective and efficient algorithm for automatically mining result records from search engine response pages.

Keywords – Information Extraction, Wrapper, Multi Agent System, JADE, and Search engine Result Page.

1. Introduction

TODAY, the World Wide Web contains several billions of information and is still growing at a very faster rate as most of the people use the internet for retrieving interesting document. Search engines are very important tools for people to reach the vast information on the WWW. Recent studies indicate that Web Searching is the most popular activity on the Internet. Thus developing user friendly tool for extracting the relevant content without accessing the complete data on the outset has become an important concern among the Web mining research communities. The problem of information extraction is to transform the contents of input documents into structured data, and the problem of information extraction from a Web page is to apply information extraction to Web pages. ([1]-[4]). Unlike information retrieval, this concerns how to identify relevant documents from a collection; information extraction produces structured data ready for post-processing, which is crucial to many applications of Web mining and searching tools.

Survey indicates that there are hundreds of thousands of search engines on the web. Not only web users interact with search engines, many web applications also need to interact with search engines. For example, Meta search engine utilize existing search engine to perform search and need to extract the search results from the result page returned by the search engine used [3]. As an another example, deep web crawling is to crawl
data records from (deep web) search engine and it too needs to extract the search result from the result pages returned by search engines.

Data integration and comparison shopping need the data from multiple web data sources [13]. Only when the data are extracted and organized in a structured manner such as tables, can be compared and aggregated, event agents that collect information about events that occurs at different locations and times, and news agents that collect news articles from different sources and presents articles relevant for a specific user. Information stored in free natural language text or with semi-structured format would be too difficult to handle directly without IE for these applications. This paper focuses on the problem of how to extract search result records (SRR) or data records from dynamically generated result pages returned by search engines in response to submitted queries.

A typical result page contains multiple SRR plus some irrelevant to the user query such as navigational panel, advertisements, comments, Menus. The objective is to extract Search Result Record (SRR) and discard irrelevant information from a result page. Web Information Extraction (IE) is the key step of the data integration of multiple sources from the same domain. It tries to extract semi structured data from web sites of specific domain of interest and make it available to queries.

This paper focuses on segmentation and detection of noise issue. This paper is organized as follows: section 2 discusses the related work. Agent Technology and Agent Communication is discussed in Section 3 and Section 4. Section 5 describes system architecture for identifying data records. Section 6 concludes the paper.

2. Related Work

The field of the WebIE has seen active research in the last decade. [1] – [10]. Many researchers have worked on finding efficient WebIE strategies for semi structured information from Web pages. According to the automation degree, all the works can be classified into three kinds: manual, semi-automatic, and automatic. Earlier works were mainly manual and semi-automatic [1, 2]. Most current applications have to extract their desired data from a very large number of web pages. To improve the efficiency and reduce the manual efforts, most recent research focuses on automatic approaches instead of manual or semi-automatic approaches. In this section, we mainly introduce the automatic approaches.

The key component of a wrapper is the algorithm that checks the similarity of data records. Data records are retained and considered valid if they are similar and discarded if they are dissimilar. Some representative automatic approaches are Data Extraction based on Partial Tree Alignment (DEPTA) [2, 3] and Mining Data Region (MDR) [5] use edit distance techniques to check the similarity of the structure of data records. These approaches perform only data record extraction but not data item extraction. The techniques of these works have been discussed and compared in [5].

Examples of systems generating wrappers automatically are RoadRunner [6] and Bing Liu et al. [5] developed an approach called Mining data Records (MDR), which is capable of extracting information from regularly structured Web pages. Their approach depends on two observations regarding the content of the Web page: the location of records describing similar objects and a string matching algorithm. They demonstrated experimentally the validity of these two observations for the tested Web pages.

ViNTs [7] works best on search engine results. The method assumes that the data records are in a contiguous region. If there is noise (advertisement) in the middle, the method classifies either the noise as data records or entirely misses the data records because of noise interruption. In order to generate the wrapper, it requires some result pages with many records and one result page without records which show the limitations of the approach.

Papadakis et al. [8] developed a wrapper called STRAVIES, which employs hierarchical clustering techniques for automated IE from Web pages. This method is restricted by using the cardinality of common ancestors of two nodes as the similarity measure.

The current fully automatic methods, such as Visual segmentation-based data record, VSDR [9] extraction method is based on visual segmentation. The author adapted the VIPS algorithm [10] to perform the initial semantic partition, and store the results in a XML file.
3. Agent Technology and Multi Agent System

Agent Technology is a new concept derived from artificial intelligence. The term agent describes a software abstraction, an idea, or a concept, similar to Object Oriented Programming (OOP) terms such as methods, functions, and objects. The concept of an agent provides a convenient and powerful way to describe a complex software entity that is capable of acting with a certain degree of autonomy in order to accomplish tasks on behalf of the user. But unlike objects, which are defined in terms of methods and attributes, an agent is defined in terms of its behavior.

Agents itself have several characteristics that makes researchers interested to explore the agent technology. The term agent, or software agent, has found its way into a number of technologies and has been widely used, for example, in artificial intelligence, databases, operating systems and computer networks literature. Therefore, an agent is autonomous, because it operates without the direct intervention of humans or others and has control over its actions and internal state. An agent is social, because it cooperates with humans or other agents in order to achieve its tasks. An agent is reactive, because it perceives its environment and responds in a timely fashion to changes that occur in the environment. An agent is proactive, because it does not simply act in response to its environment but is able to exhibit goal-directed behavior by taking initiative. [14]

For real world applications single agent is not enough. So we go for multi-agent. A multi-agent system (MAS) is a system composed of multiple agents acting collectively to reach the goals that are difficult to achieve by an individual agent or monolithic system.

Multi agents system is selected for the proposed system for several reasons. First of all, Integrating data from various sources ie., from various web pages is a large complex task, web pages are highly dynamic and uncertain. Second, agents are capable of independent actions on behalf of a user or owner and can act, capture and manage information automatically when it is necessary. Thirdly, agents can interact with other external systems and can be used to manage both distributed and local knowledge. Fourthly, agents can learn from their own experience. This is particularly important in the field of web mining as the data is constantly modified and updated. This results in the system to perform better over time since the agents have learnt from their previous experiences. Finally, agents have the autonomy and social ability, and multi-agent system is inherently multithreaded for control. Therefore, multi-agent approach is suitable for the development of WebIE systems. Multi agents are:
- Heterogeneous agents having expertise in different areas.
- Self-motivated
- Act to fulfill internal goals
- Share tasks with others
- Communicate and collaborate
- No global or centralized control mechanism.

4. Agent Communications

When we talk of typical multi-agent systems, we mean systems in which usually no agent knows of the existence of a particular other agent. So, the agents need to have a strong ability to communicate in order to forward tasks or ask for information from other agents. These systems are open, which means additional agents can just be placed into the system without the need to announce them to some special instance first. In addition, multi-agent systems may contain several different instances of the same agent whereas in our prototype, each agent has its own specific task which no other agent in the system is able to fulfill.

The cooperation among agents in this system is as follows. At first, Query Result Page is obtained from preprocessing agent and query result is transferred to processing agent. Query result pages are classified by page preprocessing agent and then wrapper agent gets extracted query result page from data tidy. Afterwards processing agent will have list of data regions. We use Iterative Deepening Search (IDS) technique to detect and label the different data regions. In effect iterative deepening combines the benefits of depth-first and breadth-first search.

Agent communication technology mainly includes ACL (Agent Communication Language), KQML (Knowledge Query and Manipulation Language) etc. ACL is selected as the communication language of this system. Based on existing function of ACL, the system can define agent communication description according to its own specific business. [14].
5. Prototype Model-Agent Wrapper

The model is designed to cover the limitations of traditional wrapper and its purpose is to design the intelligent information extraction model based on multi Agent system. In this section we discuss the requirements and the assumptions made for extraction of data records from search result page. Currently, data extraction is facing challenges as below:

(a) Data have different layouts and patterns in web sites.
(b) Data items use different expression forms.
(c) It’s hard to identify and extract when single data record existed in some web sites.

In order to solve problems mentioned above, we decided to use JADE as our implementation Tool for agents. JADE (Java Agent Development Environment) is a middleware that facilitates the development of multi-agent systems. It provides a FIPA (Foundation for Intelligent Physical Agents) compliant environment and implementation of Multi agent system [12]. Also for Agent’s Modeling Language (AUML) is being defined to effective implementation of defining agent roles. Now, we have to go more into details and examine the general architecture for the Agent Wrapper that we want to build. Regarding the agent, we have chosen to use the JADE Tool. As already described in Section 3, agents shall be autonomous entities which act on their own.

Additionally, we have stated that our agents may be compositions of several single agents or entities. This paper suggests an Agent based data record extraction method. In fig 1., the whole process of data extraction has been demonstrated.

Figure 1: System Framework

There are two stages in the Information Extraction from web Pages. The first stage involves parsing the HTML page and organizing it into the DOM tree representation. In the second stage, Agent wrapper extracts data records using tree matching algorithm. Most of the Web pages are described by Hypertext Markup Language (HTML) which is a kind of language. Web pages normally contain a lot of advertisements and other multimedia information such as sounds, images, videos which are not the real value for us to understand pages. Since the form, the script, comments and other elements contained in the Web pages are unsafe codes, we need to remove these elements. A search engine result page is required as input for a parser to parse this web page. Clean the returned HTML documents by using "HTML Parser" and convert HTML documents which include confused labels and irregular format to well-formed XHTML documents, and then build a DOM tree based on the nested hierarchical structure of HTML tags in the Web pages it holds all the information in the pages, or even reconstructs the pages through the tree traversal.

Once the sample web pages are parsed, Agent Wrapper stored and arranged the contents in a DOM tree, which will be used for further processing in the subsequent stages. Once a web page is parsed and represented in a DOM Tree structure, our Agent Wrapper needs to traverse through the DOM Tree and identify the various data regions in the web page. Although we can obtain multiple data areas, there were only one or several data areas we are interested in; hence, we need to further mine interested data areas. To achieve this, we
use Iterative Deepening Search (IDS) technique to detect and label the different data regions. In effect iterative deepening combines the benefits of depth-first and breadth-first search. In general, iterative deepening is preferred search method when there is a large search space and depth of the solution is not known.

Our IDS technique developed is based on the improved and modified version used in MDR [4]. A data region can be defined as a set of data records. Data records in turn can be defined as any records that have similar parent HTML tag contains repetitive sequence of HTML tags and are located in the same level of the DOM tree. In the process of mining interested data areas, this paper utilizes method of heuristic rules. In the case where none of the nodes can satisfy this criterion, then the search will go one level lower and perform the search again on all the lower level nodes. Agent Wrapper treats and labels these similar tag nodes as one group.

5.1 Processing agent

To locate and extract the relevant data region from a pool of available data regions, Agent Wrapper uses heuristic techniques for data extraction; each of them is related to the definition of a data record. The authors of the papers [2, 5, 7, and 10] on Information Extraction in Query Result Record Pages have pointed out several unique features inherent to a data record. We have also made several observations on the constitution of a data record. Based on these observations, we come out with a way to apply heuristic techniques to correctly extract a data region. The following are the observations made by several authors as presented in their papers:

(a) Data Records usually consist of three HTML tags that make up their tree structure.
(b) Data regions are always centered horizontally.
(c) The size of the data region is usually large relative to the area size of the whole page.
(d) All data records are adjoining.
(e) Data Records usually occur more than three times in a given web page.

By investigating a large number of Web pages, we found two Interesting facts. First, data regions are always located in the middle section horizontally on Web pages. Second, the size of a data region is usually large when there are enough data records in the data region.

After the completion of IDS extraction, Agent Wrap will have a list of data regions. Our examination shows that data regions fall into one of the several groups. We group the first set of potential data regions as menus; these typically determine the layout of HTML pages and are usually large in size and highly dissimilar. The second data region group is advertisements, regions of this group are highly similar but with simple structures. The third group of data regions consists of menu bars; these are simple but are nearly similar in structure. The fourth and last group in these groups of data records is relevant to our work, they are the search engine results output. This group of data records is highly similar in structure and large in size. We aim to design our wrapper so that it can extract the last group of data regions, while removing the other irrelevant ones. We proposed Ad filtering Agent to remove advertisements, Menu filtering Agent to remove menus which determine the layout of the HTML page, and finally Irrelevant filtering Agent to remove the remaining irrelevant data records. Data Record filtering Agent is designed to remove data records from Query result page.

In this rule, Agent Wrapper performs the filtering process based on Observation (a). Once the list of the data regions are obtained from IDS Extraction, Ad filtering Agent involves removing data records that have less than three HTML tags in each and every group. The purpose of this filtering stage is to remove advertisement related information. We observe that advertisement usually contains simple structure to present its content, usually a list of hyperlinks as its content. Based on Observation (e), Agent Wrap will filter out irrelevant data records. Data records occurring less than 3 times will be filtered out and excluded for further processing.

5.2 Data store agent

The task of data store agent is to check the extracted data with the help of data check KB. For the acceptable data, they will be stored in database. In contrast, for some questionable data, the system will remove these data from result.

6. Conclusion

We have used the Multi Agents for implementing the system in JADE environment. We have discussed the extraction techniques that our agents use to extract information from Web pages. We are mainly focusing on HTML documents. In this paper we proposed a new WebIE system is decomposed to some sub-problem
applying agent technology and the system framework is designed based on multi-agent. Retrieving relevant content from the web is a very common task. However, the results obtained, by most of the extraction systems do not necessarily produce result that is best possible catering to the user needs. Moreover, the communication and cooperation between agents are introduced. Each agents take full advantage of their autonomous and intelligence to work together so as to carry out the general goal of the WebIE system. With the deepening of this issue, we will further improve this model to make it have more adaptability and expansibility so that the model can be widely used in the field of WebIE. The current system will process only the text information and accepts the HTML/plaintext document as its input. In future the current system may be extended to process the image, audio and video information. The current system may be extended to accept the other types of documents, such as PDF as its input in our further research we plan to extend our works and then implement them.

7. References


