Asian Journal of Information Technology 4 (3): 41-48, 2005
Special Issue on Machine Learning in China - Selected Papers of CCML'04

A Hybrid Machine Learning Approach for Extracting Information from WWW

Kun Yu, Zhi Cai, Xufa Wang, Qingsheng Cai
Department of Computer Science and Technology,
University of Science and Technology of China, Hefei, P. R. China

Abstract: This paper presents a hybrid machine learning approach to extract information from WWW. It applies structure analysis to improve the extraction accuracy, with 96.5% average precision and 96.7% average recall for static web page, and 100% precision and recall for dynamic web page. Furthermore, the working time is short (< 800 ms) and the number of learning examples is small (< 4) due to little user participation. Our results prove that this approach offers the attractive advantageous of fast, convenient and high-accuracy requirements of practical applications.

Key words: machine learning, structure analysis, information extraction, WWW

Introduction

Recently, many researchers have paid great attention to information extraction (A.H.F. Laender *et al.*, 2002; Alon Y. Levy *et al.*, 2000; J. Hammer *et al.*, 1997; P. Atzeni *et al.*, 1997; Robert B. Doorenbos *et al.*, 1997; Mike Perkowitz *et al.*, 1997; Ling Liu *et al.*, 2001; Ling Liu *et al.*, 2000; David Butter *et al.*, 2001; D.W. Embley *et al.*, 1999; A. Sahuguet *et al.*, 1999; Mathias Bauer *et al.*, 2000; N. Kushmerick, 2000; N. Kushmerick *et al.*, 2003; C. Hgu *et al.*, 1998; Ion Muslea *et al.*, 1999) to accomplish web information managing work (A.H.F. Laender *et al.*, 2002; Florescu *et al.*, 1998; R. Gaizauskas *et al.*, 1998). With the rapid increase of information on World Wide Web, it is imperative for us to apply this technique to actual application. This trend brings forward more requirements to information extraction, including short working time, convenience and with high-accuracy.

Because the current technologies, such as handmade "wrapper" approach (J. Hammer *et al.*, 1997; P. Atzeni *et al.*, 1997; Robert B. Doorenbos *et al.*, 1997; Mike Perkowitz *et al.*, 1997) and structure analysis approach (Ling Liu *et al.*, 2001; Ling Liu *et al.*, 2000; David Butter *et al.*, 2001; D.W. Embley *et al.*, 1999; A. Sahuguet *et al.*, 1999; Mathias Bauer *et al.*, 2000) usually deal with information manually by experts, and the rapid change in both format and the content of some sources (e.g. Web pages) makes it very expensive or even unfeasible to be managed manually, applying information extraction to practical application is believed very difficult. Thus, there has been increasing interest in applying machine learning in information extraction (N. Kushmerick, 2000; N. Kushmerick *et al.*, 2003; C. Hgu *et al.*, 1998; Ion Muslea *et al.*, 1999). These methods hardly need domain knowledge and can carry out the various changes automatically, which makes machine learning based approach being used more frequently in practical area than other approaches. However, some problems still exist, such as low-accuracy in some special circumstances and time-consuming in its learning phase. The target of this paper is then, to find a method to remedy the drawbacks of current machine learning based approach and satisfy the practical requirements.

Among the existent information extraction approaches, the structure analysis based approach usually treats web pages as a tag tree according to the nested characteristic of HTML, and then gets the extraction rule by manually analyzing the position of target information in the tag tree with experts' participation. It has higher precision and recall compared with other approaches (Ling Liu *et al.*, 2001; Ling Liu *et al.*, 2000; David Butter *et al.*, 2001; D.W. Embley *et al.*, 1999; A. Sahuguet *et al.*, 1999; Mathias Bauer *et al.*, 2000). To learn extraction rules automatically and achieve high extraction accuracy, we integrate structure analysis with machine learning to extract information. Moreover, we use user participation to shorten the learning time. In this paper, two kinds of web pages were tested for the evaluation of this approach. One is static web page, which has forty randomly selected newspapers sites. The other is dynamic web page that is returned by search engines from ten Internet shopping markets. The experimental results show that for static web pages, the average amount of learning examples is only 2.1; the total working time is as short as 264.9 millisecond per page; and the average precision and average recall, as high as 96.5% and 96.7%, can be achieved respectively. For dynamic web pages, the average number of learning examples is 2.2; the working time is a little longer (626.44 millisecond per page); but it gets 100% precision and recall on all the testing web sites. These results prove that our approach not only combines the merits of both machine learning based approach and structure analysis based approach, but also avoids their drawbacks at the same time. And it can meet the requirements of practical applications simultaneously.

Approach Design: Our hybrid machine learning approach is composed of four phases: syntactical structure

normalization, extraction rule learning, information extraction, and estimating and rule modifying (Fig. 1). It first analyzes the structure of a web page and parses it as a tag tree. Then it induces extraction rules based on the examples offered by user. At last, it extracts target information in the tag tree with the extraction rules. In order to improve precision, this approach also needs user's feedback about extracted messages to modify extraction rules.

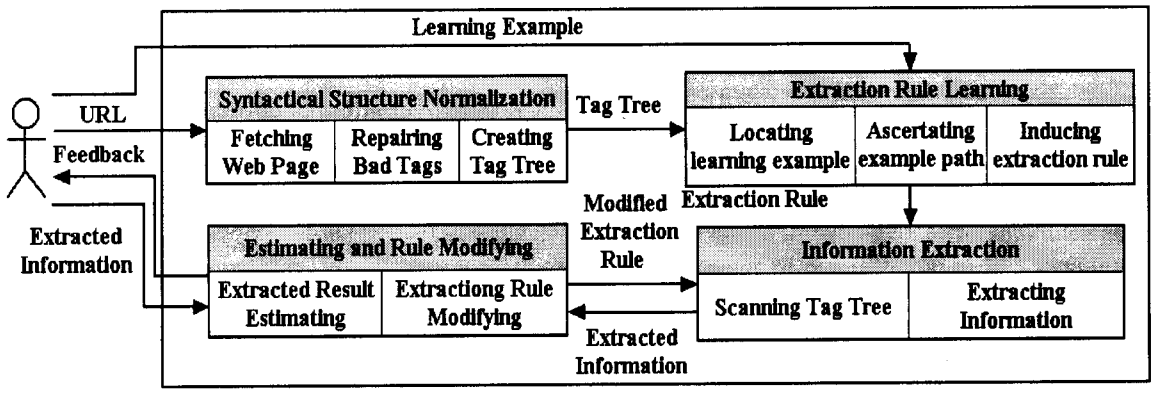


Fig. 1: Approach architecture

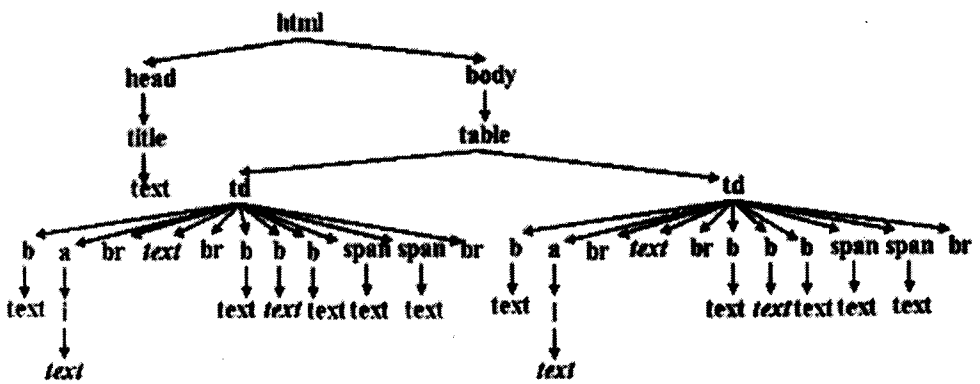


Fig. 2: Tag tree of sample page at http://www.amazon.com

1. Make f = the root of tag tree T;
2. Pop out the first two elements from Rule Stack R and make
   t = the tag element,
   l = the location element;
3. If (l != "*")
   Scan T and go to the node that is the l^{th} child of f and named t;
4. Else
   Scan T and extract nodes that are children of f and named t;
5. Make f = t;
6. If R is not empty
   Go to 2;
7. End.

Fig. 3: Procedure of Information Extraction Phase

Compared with traditional structure analysis based approach, our approach only needs non-expert user affords learning examples to a friendly user interface to help analyze the structure of web pages. The user's participation also shortens the learning time of our approach.

Syntactical Structure Normalization: This phase is dedicated to preparing the web document for extraction. It takes a URL from user, and performs three tasks: fetching web page, repairing bad tags, and creating tag tree.

Most web documents are hypertext documents that are written according to Document Type Definition (DTD) and include plain text and tags. Most of the tags are twinned, that is a pair of tags consists of a start tag (e.g.<body>) and an end tag (e.g.</body>, except for a few tags like
, <meta> etc.). But the nonstandard character of HTML always makes programmers omit the end tag of twinned tags, which brings great difficulty to structure analysis. Therefore we must repair those unpaired tags and ignore comment tags (e.g.<!>) after fetching a web page to facilitate the tree creating process afterwards. Here we use a standardization class named Tidy (http://www.w3.org) to fulfill the normalization task. Furthermore, a remedy procedure is also applied to deal with the mistakes during the conversion of special tokens (e.g. "&deg") by Tidy.
Considering of the twinned and nested tags of HTML, we create a strict tag tree to show the hierarchy of web documents after repairing bad tags. A node N in the tag tree T consists of seven elements, in which the TagName and Content elements represent the basic information of this node, and the Parent, LeftChild, RightBrother, ChildNum elements represent the node location. Moreover, the Pass element represents whether this node has been scanned in the extraction phase, through which we can improve the searching efficiency. Fig. 2 shows the tag tree of part of a sample page at http://www.amazon.com. In this figure, the TagName of the plain text node is "text" and the Content of the node is the plain text.

Learning of Extraction Rules: After normalizing the web page, the task of the extraction-rule learning phase is to explore the structure of the target information in a declarative extraction rule language by machine learning with few user-offering examples and little user's participation. For a HTML document, this phase takes a tag tree created by the syntactical structure normalization component as input. Then it scans the tag tree to locate the examples supplied by user and gets the paths of examples. At last it induces extraction rules through the example paths.
To alleviate user burden of structure analysis, we use a friendly interface to help user provide learning examples and set extra parameters. To offer learning examples, user only needs to copy the objects that they want from the web page to this interface. Moreover, user can also offer the sequence that the example occurs to orient user examples more exactly.
The first step of this phase is locating learning examples and getting example path. The italic texts of Fig. 2 are two learning examples, each of which has three elements: name, author and price. In order to locate these learning examples, we must scan the whole tag tree and note the ancestor nodes of these texts. Then we use a special format, in which the ancestor nodes and the target node are arranged with up-to-down sequence, to express learning example path. Each node of the path is composed of two elements: tag and location, in which the tag element is the TagName of this node and the location element is the ChildNum of this node, which shows which number of child it is of its father. For example, the example path of the most left italic text node is:

<html>0<body>2<table>1<td>1<a>2<i>1<text>1

And the example path of the most right italic text node is:

<html>0<body>2<table>1<td>2<b>7<text>1

After getting the paths of user-offering examples, we are going to induce extraction rules through these paths. If we number the italic nodes in Fig. 2 from left to right, we will find out that node 1 and node 4, node 2 and node 5, node 3 and node 6 are three parts that show the name, author and price of the messages respectively. Looking at the example paths of node 1 and node 4, we can find out that the tag elements of these two paths are the same but the location elements have little difference.

Example path of node 1: <html>0<body>2<table>1<td>**1**<a>2<i>1<text>1
Example path of node 4: <html>0<body>2<table>1<td>**2**<a>2<i>1<text>1

This interesting property suggests us to induce the example paths to constitute extraction rules. Comparing the example paths of node 1 and node 4, we remain the same tag and location elements and change the different location element following the same tag element to a wildcard " *". Then we can induce the extraction rule of the name part as following:

Extraction rule of name part: <html>0<body>2<table>1<td>*<a>2<i>1<text>1

Analogously, we can induce the extraction rules of the author and price parts as following:

```
1. Make f = the root of tag tree T;
2. for i1 = 1 to n1
     for i2 = 1 to n2
       .....
         for im = 1 to nm
           for j = 1 to m
2.1.           Assign ij to the jth "*" in R;
2.2.           Pop out the first two elements from Rule Stack R and make
                 t = the tag element,
                 l = the location element;
2.3.           If (l != "*")
                 Scan T and go to the node that is the lth child of f and named t;
2.4.           Else
                 Scan T and extract nodes that are children of f and named t;
2.5.           Make f = t;
2.6.           If R is not empty
                 Go to 2.2;
2.".           Else
                 Refresh R;
3. End.
```

Fig. 4: A Modified procedure of information extraction phase

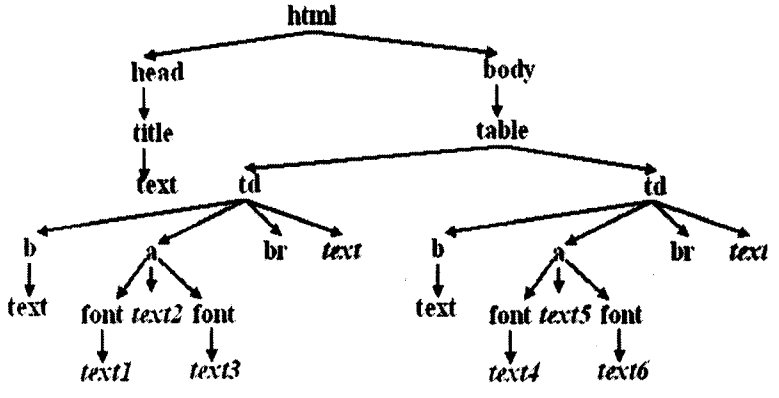
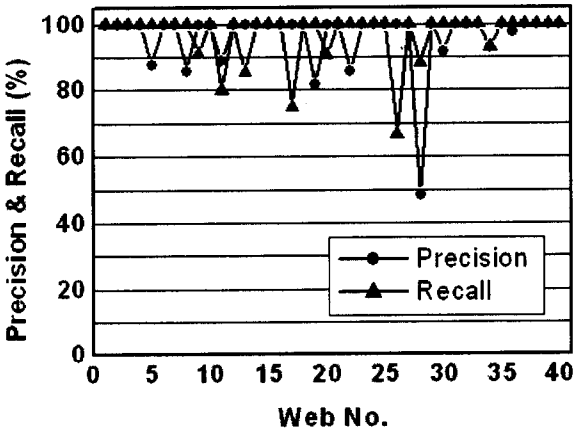
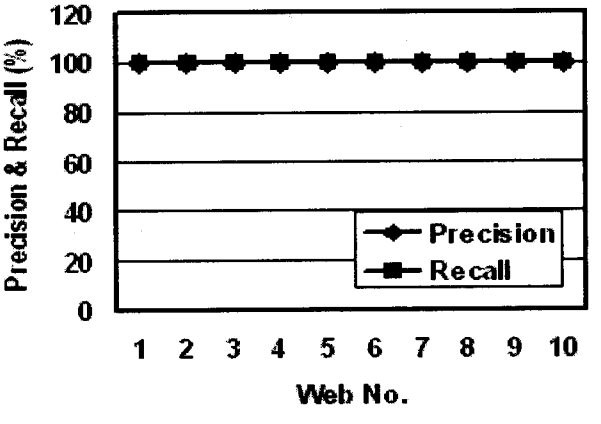


Fig. 5: Tag tree of incoherent information

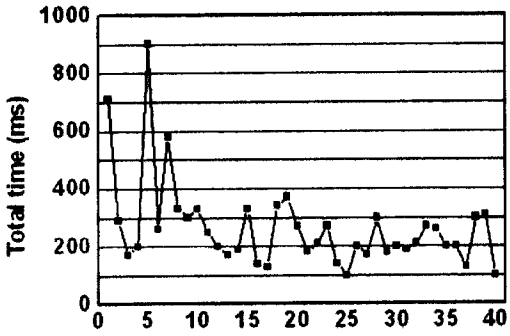


(a) Static web page

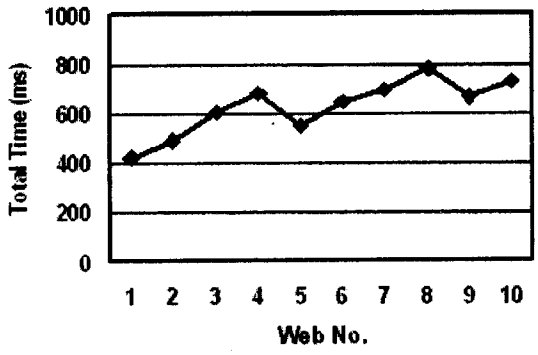


(b) Dynamic web page

Fig. 6: Results of precision and recall



(a) Static Web page



(b) Dynamic web page

Fig. 7: Results of total working time

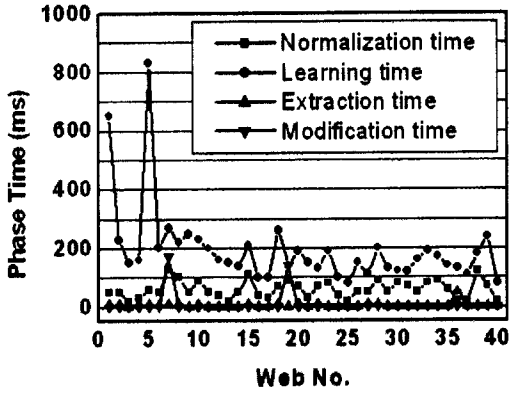
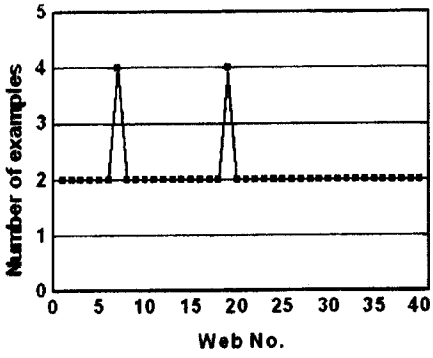
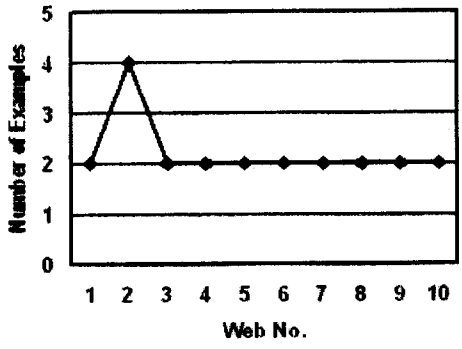


Fig. 8: Working time of four phase for static web page



(a) Static web page



(b) Dynamic web page

Fig. 9: Results of the number of learning examples

Extraction rule of author part: <html>0<body>2<table>1<td>*<text>4
Extraction rule of price part: <html>0<body>2<table>1<td>*<a>2<i>1<text>1

In order to improve the precision of information extraction, we modify extraction rules according to user's feedback information. This task is usually executed in the instance that there are different types of layout in one site. For example, if a product in a web site has different layout to that of other products, it will be lost in the information extraction phase. After examining the results returned by phase 3, user indicates the lost of this product and applies this product as a modify example to phase 4. Then phase 4 induces a new extraction rule from the modify examples in the procedure of phase 2, and adds this rule to existent extraction rules. Phase 3 extracts information with modified rules and returns results back to user. Such process has to be repeated until user is

satisfied with extracted messages. Experimental results show that the time consumed by this phase is very short compared with the total working time (Fig. 8), which prove that this work will not affect the overall efficiency.

Information Extraction: With the extraction rules induced in phase 2, the task of information extraction phase is to scan the tag tree along the rules and extract information at certain location. Here we use a stack *R* to save the extraction rule, in which the first element of this rule is on the top of the stack.
Fig. 3 shows the procedure of this phase, which includes seven steps. At the fourth step, we scan the tag tree and extract nodes that are children of *f* and named *t*. This procedure is successful for extracting information when there is only one " *". But if there are several " *" nodes in one extraction rule, it is difficult to extract all the target nodes with this simple procedure. So we use another modified procedure (Fig. 4) to accomplish the extraction task when there is more than one " *" node in the extraction rule. The variant *m* in Fig. 4 represents the total number of " *" nodes. And the variants $n_1 \ldots n_m$ means the number of children of the m^{th} "*" node. Experimental results show that almost all the target information can be extracted from tag tree through the modified procedure of information extraction phase (Fig. 6) successfully.
Fig. 4 shows a successful extraction process. But the time complexity of this procedure is $O(n_1 * n_2 * \ldots * n_m * m)$, which is too slow for online information extraction. Therefore, how to reduce the time complexity becomes a key question.
We use a heuristic rule to solve this problem. Every node of tag tree has seven elements, among which the Pass element is used to fulfill the heuristic search. If one node has been extracted, the Pass element of this node is changed to true, which means that we do not need to scan this node again when extracting other information. Using this rule, our approach needs to scan one node in the tag tree only once and the complexity of information extraction phase drops to $O(n)$ (here n is the total number of nodes in tag tree). This technique is also used in the example path-ascertaining step and the complexity of that step is $O(\log n)$.

Incoherent Information Extraction: It is worth noticing that when information to be extracted is not coherent, we cannot make sure the location of target information with only one wildcard "*". For example, the target information is the patchwork of three nodes text1, text2, text3 and the patchwork of text4, text5, text6 (Fig. 5). To solve this problem, we use another wildcard " -2" to differ this type of information from the ordinary ones. We assign " -2" as the location element to the node that is the common ancestor of target nodes and ignore the child nodes of this ancestor. For example, the example paths of the target information in Fig. 5 are:

<html>0<body>2<table>1<td>1<a>-2
<html>0<body>2<table>1<td>2<a>-2

Adopting the induction technique, we can get the extraction of target information as:
<html>0<body>2<table>1<td>*<a>-2

To extract incoherent information accurately and avoid extracting those irrelevant nodes, we only need to extract all the plain text nodes that are children of *f* and connect them together with left-to-right sequence in step 2.4 (Fig. 4).

Experimental Settings: To evaluate our hybrid machine learning approach, an information extraction tool based on this approach has been developed. This tool is composed of two parts: rule wrapper and information extractor. The rule wrapper first learns extraction rules from the representative web page and modifies the rules with user's participation. Then the information extractor runs in background and extracts target information from web sites according to the extraction rules.
Using this extraction tool, two kinds of experiments have been employed. One is extracting information from static web pages. We chose forty U.S. newspapers from the NewspaperLinks.com (www.newspaperlinks.com) randomly and extracted the name of news or headlines of each site. The other is information extraction from dynamic web pages. We first search predefined products from the search engines of ten randomly selected Internet shopping markets. Then we extract the name and price of those products from the web pages that are returned by the search engines. In these experiments, we first download the first page of each site as learning page and run the rule wrapper offline, and then extract target information from each site online. All these experiments were performed by a non-specialist extractor (i.e. a non-computer-major student).

Results and Discussion

In practical application, accuracy, working time and user burden are believed the most important factors to evaluate a system. So our experiments mostly concerns about them. Here accuracy is represented by precision and recall;

working time is composed of the runtime of four phases mentioned above; and user burden is expressed by the number of learning examples that user must offer to the wrapper.

Fig. 6 shows the precision and recall of our approach. We can see that for static web pages, it exhibits the average precision of 96.5% and average recall of 96.7%. And for dynamic web pages, the average precision and average recall, as high as 100%, can be achieved. These results indicate that our approach can deal with most of the testing sites successfully, which is as good as that of structure analysis based approach (David Butter *et al.*, 2001), due to two reasons: first, our approach combines structure analysis and inherits the high-performance merit; second, we use user's modification about extraction rules in Phase 4 and it improves the performance successfully.

The total working time of our approach is shown in Fig. 7. We can see that for static web pages, the average total run time is 264.9 millisecond, and for 92.5% experimental sites the working time is lower than 0.4 second. For dynamic web pages, the working time is less than 800 millisecond although there is more information in each page. It indicates that adding user participation to learning process can shorten the working time remarkably.

The distribution of working time in different phases is shown in Fig. 8. It exhibits that the working time is mainly distributed in preprocessing and learning phase (phase 1 and phase 2), while the extraction phase (phase 3) and rule modification phase (phase 4) only consume little time. It shows that user' s modification about extraction rules does not affect overall efficiency but improves the extracting precision effectively.

Moreover, Fig. 9 shows the results of the number of learning examples in both static web pages and dynamic web pages. Only two learning examples are required by more than 90% testing sites. The average number of examples needed for static page by our approach is 2.1, and the number for dynamic page is 2.2. This result is closed to that of the traditional machine learning based approach (N. Kushmerick, 2000), which is 2 on 30 randomly selected Internet sites, and are believed will not bring any obstruction to the practical application.

Conclusion

We have proposed a hybrid machine learning approach, which combines structure analysis with machine learning, to automatically extract information from WWW. This approach has been tested and evaluated with both static and dynamic web pages. According to these primary experimental results, high precision and high recall can be achieved by our hybrid machine learning approach with short working time and little user participation. It proves that our approach can meet the convenient, fast and high-accuracy requirements of practical applications.

While, there are still some problems need to be investigated in the future. For example, the performance of our approach is not very good in some special sites. There are two possible reasons. First, it only uses positive learning examples in learning phase so that it cannot distinguish those wrong captions that have the same format of target information successfully. Second, we use a strict tag tree in structure analysis, but some sites have pictures besides the messages that destroy the repeating structure. We plan to address these questions in our future work. We consider adding some location symbols to help find the boundary of a group of information, using a more flexible tag tree and adding negative learning examples to address these problems. Our future work will focus on the fulfillment of these plans.

Acknowledgements

This research is funded in part by National Natural Science Foundation of China (No. 70171052, No. 60075015).

References

Laender, A. H. F. ,B. A. Ribeiro-Neto, A. S. da Silva, J. S. Teixeira, 2002. A brief survey of web data extraction tools. ACM SIGMOD Record. 31: 84-93

Sahuguet, A., F. Azavant, 1999. Wysiwyg Web Wrapper Factory (W4F). Unpublished. PENN Database Research Group, University of Pennsylvania

Alon Y. Levy, S. Daniel, 2000. Weld: Intelligent Internet Systems. Artificial Intelligence, 118: 1-14

Hgu, C., M. Dung, 1998. Generating finite-state transducers for semi-structured data extraction from the Web. Information Systems 23: 521-538

David Buttler, Ling Liu, Calton Pu, 2001. A fully automated object extraction system for the World Wide Web. In: Proceedings of the International Conference on Distributed Computing Systems. IEEE Computer Society. Phoenix, Arizona, 611-621

Embley, D. W., Y. S. Jiang, Y. K. Ng, 1999. Record-Boundary Discovery in Web Documents. In: Alex Delis, Christos Faloutsos, Shagram Ghandeharizadeh (ed.): Proceedings of the 1999 ACM SIGMOD. Philadelphia, Pennsylvania, 467-478

Florescu, D., A. Y. Levy, A. O. Mendelzon, 1998. Database techniques for the World-Wide Web: A survey. SIGMOD Record 27: 59-74

Ion Muslea, Steve Minton, Craig Knoblock, 1999. A hierarchical approach to Wrapper induction. In: Proceedings of the Third International Conference Autonomous Agents. ACM Press. Seattle, WA, USA, 190-197

Hammer, J., H. Garcia-Molina, J. Cho, R. Aranha, A. Crespo, 1997. Extracting semi structured information from the web. In: Joan Peckham (ed.): Proceedings of Workshop on Management of Semi-structured Data. SIGMOD Record, Vol. 26. Tucson, Arizona, 18-25

Ling Liu, Calton Pu, Wei Han, 2001. An XML-enabled data extraction toolkit for web sources. Information Systems 26: 563-583

Ling Liu, Calton Pu, Wei Han: Xwrap, 2000. An XML-enabled wrapper construction system for web information sources. In: Proceedings of the International Conference on Data Engineering. IEEE Computer Society. San Diego, California, 611-621

Mathias Bauer, Dietmar Dengler, Gabriele Paul, 2000. Instructible information agents for Web mining. In: Proceedings of the 2000 Conference on Intelligent User Interfaces. ACM Press. New Orleans, Louisiana, USA, 21-28

Mike Perkowitz, Robert B. Doorenbos, Oren Etzioni, Daniel S. Weld, 1997. Learning to understand information on the Internet: An example-based approach. Journal of Intelligent Information Systems, 8: 133-153

Kushmerick, N., 2000. Wrapper Induction: Efficiency and expressiveness. Artificial Intelligence 118: 15-68

Kushmerick, N., B. Thomas, 2003. Adaptive information extraction: Core technologies for information agents. In: Klusch, Bergamaschi, Edwards & Petta (eds.): Intelligent Information Agents R&D in Europe: An AgentLink perspective. Lecture Notes in Computer Science, Springer 2586-2610

Atzeni, P., G. Mecca: Cut and Paste, 1997. Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS '97). ACM Press. Tucson, Arizona, USA, 144-153

Robert, B. Doorenbos, Oren Etzioni, Daniel S. Weld, 1997. A Scalable Comparison-Shopping Agent for the World-Wide Web. In: W. L. Johnson and B. Hayes-Roth (ed.): Proceedings of the First International Conference on Autonomous Agents (Agents'97). Marina del Rey. CA, USA, 39-48

Gaizauskas, R., Y. Wilks, 1998. Information Extraction: Beyond Document Retrieval. Journal of Documentation 54: 70-105