Available online at www.sciencedirect.com
ScienceDirect
Journal of Integrative Agriculture
2012, 11(5): 784-791
May 2012
RESEARCH ARTICLE

Structured AJAX Data Extraction Based on Agricultural Ontology

LI Chuan-xi[1, 2], SU Ya-ru[1, 2], WANG Ru-jing[1, 2], WEI Yuan-yuan[1, 2] and HUANG He[1]

[1] *Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, P.R.China*
[2] *School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, P.R.China*

Abstract

More web pages are widely applying AJAX (Asynchronous JavaScript XML) due to the rich interactivity and incremental communication. By observing, it is found that the AJAX contents, which could not be seen by traditional crawler, are well-structured and belong to one specific domain generally. Extracting the structured data from AJAX contents and annotating its semantic are very significant for further applications. In this paper, a structured AJAX data extraction method for agricultural domain based on agricultural ontology was proposed. Firstly, Crawljax, an open AJAX crawling tool, was overridden to explore and retrieve the AJAX contents; secondly, the retrieved contents were partitioned into items and then classified by combining with agricultural ontology. HTML tags and punctuations were used to segment the retrieved contents into entity items. Finally, the entity items were clustered and the semantic annotation was assigned to clustering results according to agricultural ontology. By experimental evaluation, the proposed approach was proved effectively in resource exploring, entity extraction, and semantic annotation.

Key words: information extraction, structured data, AJAX, agricultural ontology, semantic annotation

INTRODUCTION

Along with the Web2.0 development, AJAX (Asynchronous JavaScript XML) was proposed (Garrett 2005). AJAX application is a dynamic web application, which presents all the states by UI (user interface) events on the same URL (uniform resource locator). Many web developers apply this approach to building web application due to the characteristics of rich interactivity and the dynamic content access. More agricultural sites also migrated from the traditional web to AJAX applications. By observation, we found that the contents taking AJAX delivery generally are well-structured for the specific domain. Exploring the structured AJAX contents and annotating the extracted data are significant for further application, such as entity retrieval based on semantic and information integration.

Crawljax (Mesbah *et al*. 2008), an open source tool, is overridden to explore AJAX contents which are omitted by traditional crawler, and the agricultural ontology are employed to filter unrelated contents. HTML tags and punctuations are used to segment the retrieved contents into entity items, which is distinguished with existed approaches, using either HTML tags alone or syntactic analysis to extract entity items. The cluster algorithm DBSCAN is employed to cluster the similar entity items into different semantic sets. The entities and attributes of agricultural ontology supervise the annotation of indicator attributes of entity in item sets.

Ontology-based information extraction has recently

Received 28 June, 2011 Accepted 31 August, 2011
LI Chuan-xi, Mobile: 13515518016, E-mail: Ben04@mail.ustc.edu.cn; Correspondence WANG Ru-jing, Tel: +86-551-5592968, E-mail: rjwang@iim.ac.cn

emerged as a subfield of information extraction. In this method, agricultural ontology (Qian and Zheng 2006; Cui 2009) as domain knowledge is to supervise the structured data extraction and annotation. The constructed agricultural ontology contains agricultural product classification ontology (APCO), administrative division ontology (ADO), agricultural product market ontology (APMO), and agricultural commodity ontology (ACO). The architecture of the agricultural ontology is given in Fig. 1, and for the space limit, only APCO is listed.

The main contributions of this paper are:

(1) overriding Crawljax to exploit and retrieve AJAX contents of web page related to agricultural domain;

(2) using LCS (longest common subsequence) (Rick 2000) to search the repetition pattern containing in dynamic AJAX contents, and employing DBSCAN (Ester *et al*. 1996; Joachims *et al*. 1998) to cluster entity items;

(3) employing the agricultural ontology to supervise the structured information extraction and semantic annotation.

RELATED WORKS

The works mentioned here dedicate to resolve structured AJAX contents extraction and annotation from web pages, which involve AJAX contents exploiting, structured information extraction, and entity semantic annotation. In AJAX contents retrieval, Frey (2007) proposed an AJAX search model, and given a detail description of the procedure, which used state transition graph to represent the model. An open source tool Crawljax was created by Mesbah *et al*. (2008) for crawling dynamic AJAX contents. Mohan (2010) retrieved AJAX page by imitating the action of web browser, and eliminated duplicated states according the similarity of DOM elements. Mesbah and Deursen (2007) presented numerous frameworks for facilitating AJAX application development and gave a detail analysis of the different architecture styles. By comparisons, Mesbah and Deursen (2007) picked out three AJAX frameworks and summarized the three frameworks comprehensively. Marchetto *et al*. (2008) and Roest *et al*. (2010) presented the testing procedure of AJAX application, and analyzed the potential problems of AJAX application.

Information extraction aimed at extracting particular information from free text or web page (Russell and Norvig 2002; Lukose 2012). Related works of infor-

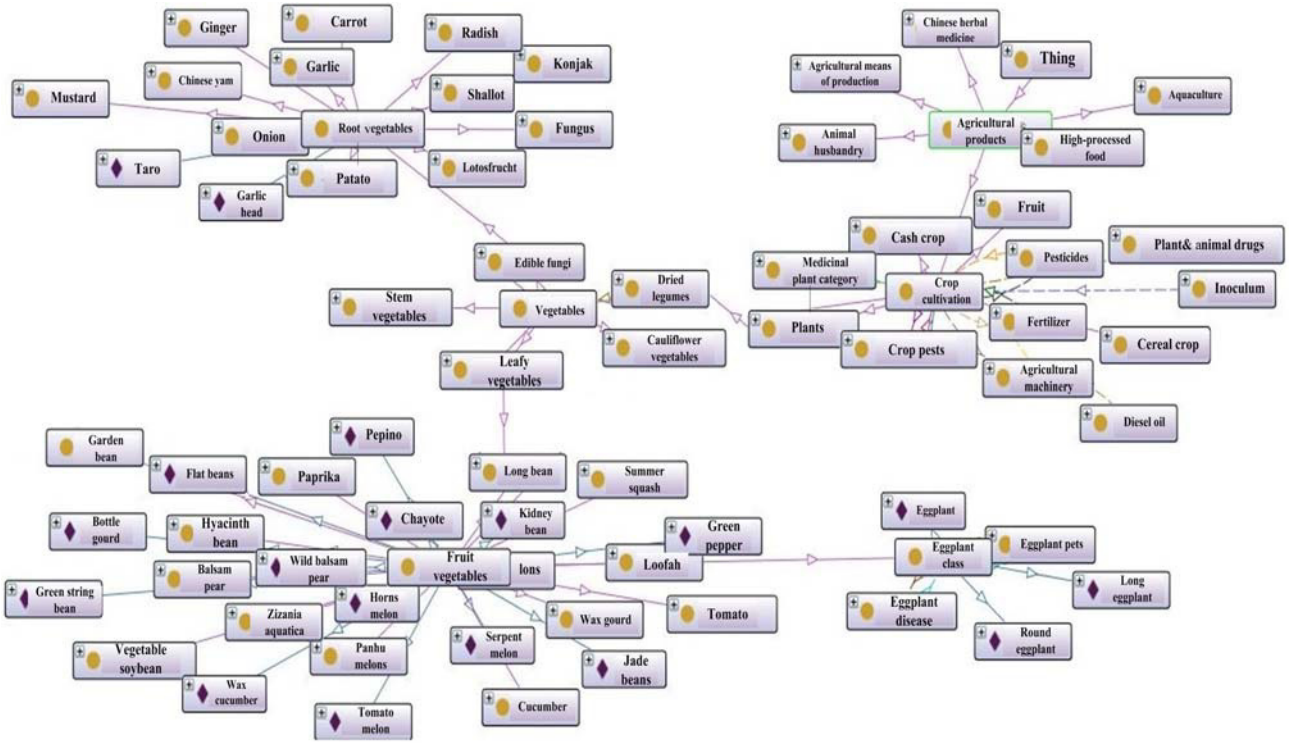

Fig. 1 The architecture of agricultural product classify ontology.

mation extraction had been performed in (Liu *et al*. 2000; Crescenzi *et al*. 2001; Cohen *et al*. 2002). Zhao *et al*. (2006) described an automatic algorithm (MSE) to extract dynamic record sections from search engine result pages and obtained a preferable effect relatively without considering the attributes variation of the entity. An adaptive extraction system was proposed by combining ontology and position-based extraction, and applied on online auctions of decisions support system (Gregg and Walczak 2006). Li and Zhao (2009) developed an agricultural ontology prototype, called AgriOnto, which massive works need to be supplemented before application. Wei *et al*. (2012) proposed a semi-automatic ontology learning method of agricultural professional ontology from web resources. Huang *et al*. (2012) discussed the related theories of building Agricultural Geographic Information Ontology.

A topic crawling method presented in the work of Shchekotykhin *et al*. (2010) exploited existing navigational structure of web page, such as index pages, hierarchical category structures, menu, and site maps, to locate related web pages, but the method ignored the page contents generated by AJAX. Diligenti *et al*. (2000) used context graph to crawl topic information, and utilized anchor text and information surrounding the links to locate related topic, but their method could not retrieve and analyze AJAX contents also. Menczer *et al*. (2001) evaluated several topic crawlers according the metrics of concrete implemented method to build text classifier by independent system or similarity measure. In our method, the inherent characteristics of AJAX contents are combined with domain ontology to perform topic crawling, information extraction, and semantic annotation.

Similar with the proposed method, Tian (2009) extracted structure data from AJAX sites by embed browser to retrieve AJAX contents, but they could not analyze the characteristic of AJAX contents, and only extracted structured data, such as tables. The work (Zhai and Liu 2005), which extract single record data from web pages, heavily relied on the tags structure information, and ignored the other situation, such as well-formed natural language text. The proposed method combines domain ontology with information extraction, using agricultural ontology to guide entity extraction and entity item annotation.

By blocking web page, Song *et al*. (2004) studied the importance of different blocks in AJAX sites, and the domain ontology was used to measure the importance of AJAX contents. Carlson and Schafer (2008) annotated the extracted record from structured web pages by matching fields to domain schema columns, without considering the semantic of the contents and the incomplete fields. Wei *et al*. (2010) used concept model to extract data, utilizing the classify relation of the concepts to trace information blocks, without considering dynamic contents and nested information.

SOBA system (Buitelaar *et al*. 2008) was implemented to extract and integrate information from heterogeneous sources, including structured information, text, and picture title, and annotated this information into knowledge base. However, we not only use ontology to perform extract and annotate information, but also use ontology to exploit dynamic AJAX information of agriculture domain.

FRAMEWORK

Fig. 2 illustrates the framework of the proposed approach which consists of three steps: dynamic AJAX contents exploiting, AJAX contents cleansing, and entity extraction and semantic annotation.

Step one: exploit and retrieve AJAX contents Given the specify URL, Crawljax will retrieve the page content, and judge whether it contains AJAX information, if exists, loading the dynamic AJAX contents and combining agricultural ontology to compute the topic similarity. If similar, Crawljax will execute AJAX calls to extract all the state containing in the site, and complete the topic contents retrieval. After this step finished, a dynamic AJAX contents set is obtained.

Step two: AJAX contents cleansing Due to the AJAX contents retrieved from algorithm 1 contains noise, it will be normalized by removing the irrelevant information, such as, JavaScript codes, comments, and HTML tag attributes. In this step, the longest common subsequence algorithm (LCS) is used to search the common repetition information in AJAX contents, and then removing the noise.

Step three: information extraction and semantic annotation Based on the above steps, the major tasks of this step is to identity the entity and annotate entity items by utilizing agricultural ontology as domain knowl-

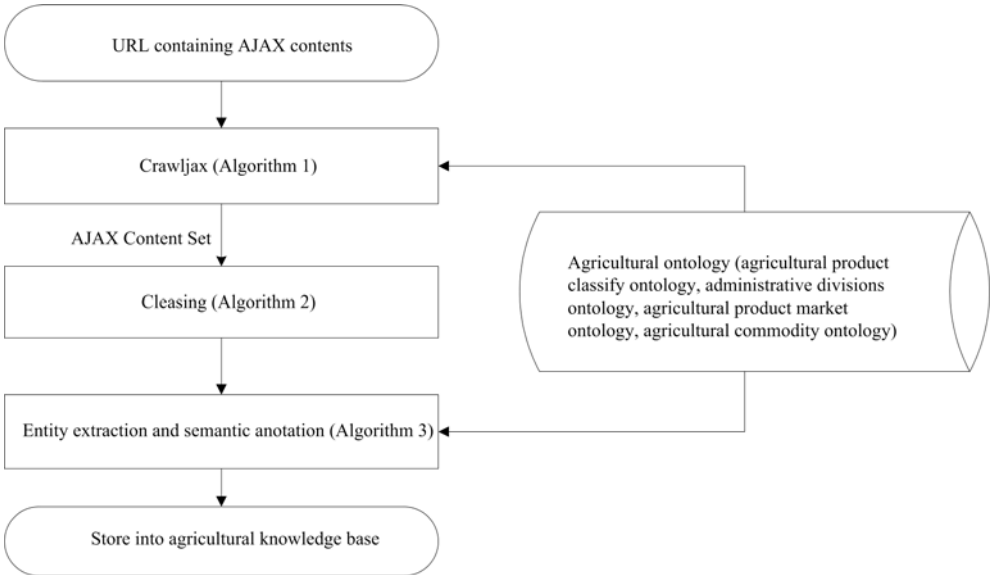

Fig. 2 The framework of structured AJAX data extraction based on agricultural ontology.

edge and employing DBSCAN to cluster the item set. To accomplish entity extraction, the AJAX contents are partition into item set utilizing the tags and punctuation sequences. And then the item set is clustered. The items containing in one cluster that have the nearest distance with agricultural ontology will be regarded as a reference source for entity partition. The entity and set partition is the foundation of semantic annotation. According to the comprehensive entity information, the item set is mapped into agriculture knowledge base.

ALGORITHMS

Before presenting the algorithm, three definitions and one formula are given as follows:

(1) AJAX contents. The contents retrieved by AJAX interaction asynchronous refer to AJAX contents, which do not contain the static page contents.

(2) Entity. Similar definition with record in database, one significant data item set refers to an entity.

(3) Item. Similar definition with record item in database, several items compose one entity (eq.):

$$simTA(wordSet, lexicon)=\frac{|wordSet \cap lexicon|}{|wordSet \cup lexicon|}$$

The algorithm and detail explanations are given as below.

Algorithm of retrieval of AJAX contents

Algorithm 1: AJAXContentExtract (*URL*)

Input: page *URL*, clickable event list:{'next','next page','>','>>'}U{1,2,3…}

Output: Dynamic content set of AJAX page: *pageContent*

1. *pageContent*=Crawljax(*URL*);
2. *pageContentSet*.add(*pageContent*);
3. *eventList*=extract(*pageContent*);
4. while (*eventList* is not empty){
5. *event*=*eventList*.get();
6. *eventList*.remove(*event*);
7. *pageContent*=Crawljax(*event*);
8. If(*pageContentSet*.notContains(*pageContent*))
9. *pageContentSet*.add(*pageContent*);
10. *eventList*=extract(*pageContent*);
11. }

In line 1, the overridden Crawljax is used to loading the AJAX page, in which interface *OnNewStatePlugin* is implemented for retrieving AJAX contents; line 3 analyzes AJAX contents and extracts clickable events list; line 6 is to remove the executed event, ensuring the events submission can terminate eventually; line 7, parameter event contains two types events: the 1st is next button which contains next, >, etc. and the 2nd is digital page number which contains the list of

page numbers, 1, 2, 3, etc. If there exist the 1st type events, ignore the 2nd, or else submit the 2nd type events, repeated this procedure until no new page tuning events existed.

Algorithm of cleansing of AJAX contents set

Algorithm 2: cleansingAJAXContentSet (*pageContentSet*)

Input: *pageContentSet*, threshold parameter *tlcs*

Output: *pageContentSet*

1. for each *pcs*∈*pageContentSet*{
2. JTidy(*pcs*);
3. RemoveNoise(*pcs*);
4. }
5. *longestCommonSubstring*=LCS(*pageContentSet*);
6. If(*longestCommonSubstring*>*tlcs*)
7. *pageContentSet*=RemoveAll(*LongestCommon Substring*, *pcs*), for all *pcs*∈*page ContentSet*

Parameter *tlcs* is used to control the similarity of two strings. Line 2 employs JTidy (2010) to clean up malformed and faulty HTML; line 3 removes all the JavaScript codes, comments, and attributes of HTML Tags; lines 5-7 find all the repeated contents, containing dynamic menu and descriptive contents, which are only for navigation or description.

Algorithm of entity extraction and annotation

Algorithm 3: entity extraction and annotation (*pageContentSet*), total classes applying DBSCAN algorithm to item set are represented as *TC*, and *TCi* represents one of the classes in *TC*. The parameters, *eps* and *minPts* are used in DBSCAN.

Input: ontologies∈{*APCO*, *ADO*, *APMO*, *ACO*}, *pageContentSet*, *MinPts*, *Eps*, *TC*

Output: *sematicAnotatedEntity*

1. *itemSet*=partition(*pageContentSet*)
2. *TC*=DBSCAN(*MinPts*, *Eps*);
3. *TCi*=Argmax{similarity(*TCi*, *onto*)}, where ∀ *TCi*∈*TC*, *onto*∈*ontologies*
4. For *item*∈*TCi*, *pcs*∈*pageContentSet*
5. {
6. *recordSet*.add(extractEntity(item, *pcs*));
7. *sematicRecord*=*Annotation*(*RecordSet*, *onto*); where *onto*∈*ontologies*
8. *sematicAnotatedEntiy*=StoreToKnowledgeBase (*SematicRecord*);
9. }
10. Function similarity(*TCi*, *ontology*){
11. *wordSet*=ictclas (*TCi*);
12. *lexicon*=getAllName(*ontology*);
13. *similarity*=SimTA (*wordSet*, *lexicon*);
14. return similarity;
15. }

Line 1 partition all AJAX contents of *pageContentSet* into *ItemSet*, according to DOM tags tree or punctuations sequences as separator; line 2 applies DBSCAN algorithm to cluster item set, and stores the clustered results into TC. In the worst case, the time complexity of DBSCAN algorithm is o (m^2), where m is the number of point. DBSCAN algorithm requires two parameters: *eps* and *minPts* which is the minimum number of points required to form a cluster; line 3 computes the similarity between the cluster TCi and agricultural ontology, where *TCi*∈*TC*, and four agricultural ontologies, and returns the *TCi* with the maximum similarity, which the similarity computation of *TCi* and agricultural ontology is implemented in lines 10-15; in lines 4-9, using item of single AJAX content in *TCi* to guide entity extraction of the AJAX contents set; line 5 extracts entity lying in one AJAX content. As shown in Fig. 3, the structure of an entity is *HPiDiSiE*, where *H*, *Pi*, *Si*, *E* are empty or the combination of several common field instances, and *Di*∈*TCi*; line 11 calls ictclas (ICTCLAS 2010), a Chinese word partition software, to

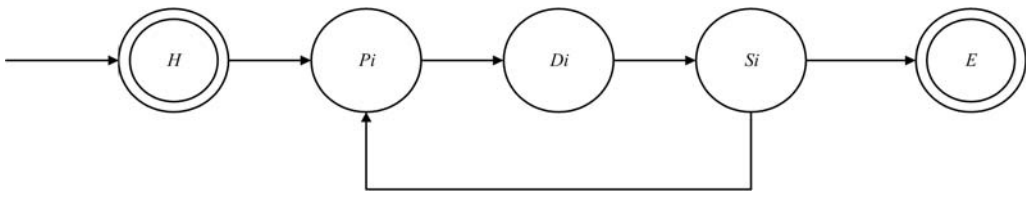

Fig. 3 The process of entity items extraction. *H and E* represent the head and foot of AJAX content, which are optional; *Di* is the item of AJAX content; *Pi* and *Si* are the repeated prefix and suffix of the item *Di*, which are also optional.

retrieve word set, and the vocabulary of agricultural ontology as specific dictionary is added; line 12 gets all the name of classes, instances, and properties containing in ontology; line 13 computes the similarity of two word sets according to the eq.

EXPERIMENTAL EVALUATION

The performance of the proposed approach was evaluated, including: (1) the accuracy of dynamic AJAX contents retrieved from web pages using Crawljax; (2) the performance of DBSCAN algorithm; and (3) the results of entity extraction.

The AJAX contents crawled by Crawljax was checked manually, and unrelated contents, such as AJAX login dialogue and weather forecast, can be filtered with simple rules. In the experiment sites, the implemented Crawljax captured the dynamic AJAX states precisely (Mesbah *et al*. 2008). The paginating symbols which are page turning tags (e.g., 'next', 'next page', '>', '>>', etc.) and page turning numbers (e.g., '1', '2', '3', etc.) were used to turn pages effectively.

Experiment environments

In algorithm 2, cleansing of AJAX contents, the value of *tlcs* was set to 15 for the reason that if *H* and *E* less than 15 (Fig. 3), the meaningful information, such as date, price unit, or the linkman of supply and demand, may be omitted.

DBSCAN algorithm was used to cluster the entity items, which the value of parameters MinPts and Eps were set in 4 and 10, respectively. Taking the inherent characteristics of AJAX content into consideration, which contains well-structured data, the partition of instances belong to different classes was very effectively even if the number of entity items is small. Given the parameters, the entity items were partition into 21 clusters. All the 21 clusters were aggregated into price entity and supply & requirement entity. Semantic annotation combines cluster results with agricultural ontology, which can be annotated mutually. On one hand, using ontology to annotate entity can facilitate the information query and semantic expansion, on the other hand, the extracted entity can be used as complementarities of agricultural ontology (Wimalasuriya and Dou 2010).

The AJAX states of each site were limited no more than 100. For the list of AJAX contents, all the events were clicked and determined by their inter-dependency. The experimental results of the proposed approach are shown in Table. The setting is described below.

520 page states were downloaded from 17 web sites (the states distribution of each site is not balance, take * in Table as an example, it contained more than 3 000 page states, for considerable reason, in this experiment, the crawled page states were limited less than 100 for evaluation). An exploited tool was implemented to mining the AJAX site from our existed agriculture sites library.

Experiment results

The column with header state, extracted state, and entity, and annotated entity lists the actual number of AJAX states on one site, the state identified by Crawljax, the existed entity containing in all states of one site, and the annotated identities by proposed approach respectively. The last two columns list the recall and precision. In the form *x*/*y* of column, *x* represents the correct extraction or annotation, and *y* represents the total extraction or annotation (Table).

Each row presents the performance of the proposed approach on a site category, except for the last row, which presents the performance over all 17 web sites.

It can be seen from Table that the proposed method is generally effective, with the recall of 0.9752 and the precision of 0.9629.

Experiment analysis

By observation, the AJAX contents containing links to non-structured text are prone to produce more irrelevant states, for example, sites 11 and 12. Although the agricultural ontology filters some irrelevant states, the results weaken when the state is very similar with domain entity.

During experiments, the irrelevant states were manually removed to measure the performance of extracting and annotating independently, and utilize MD5 (2010) to remove the duplicated AJAX states (Berson 1993). The duplicated AJAX contents, such as headers and description information, are removed effectively by LCS (longest common subsequence) algorithm. The combination of

Table Performance of the proposed extraction approach

URL	State	Extracted state	Entity	Annotated entity	Recall	Precision
http://price.gdcct.gov.cn/jgxx/*	100	100	1 500	1 500	1	1
http://agri_info.gdcct.gov.cn/	100	100	2 000	2 000	1	1
http://www.cnbajie.cn	4	3/7	38	32	0.8421	1
http://www.xinfadi.com.cn/channel/13168603	100	207	8 862	8 849	0.9981	1
http://www.cdylw.com/	3	3	34	32	0.9411	1
http://b2b.t0001.com/	8	8/10	196	189/210	0.9642	0.9
http://www.51garlic.com/	6	6	48	48	1	1
http://www.huasheng7.com/	8	8	72	72	1	1
http://www.hasc.com.cn/	19	19	77	77	1	1
http://www.hhhtagri.gov.cn/sites/nmy/	13	13/14	187	187/221	0.9950	0.8462
http://www.flowerworld.com.cn/	3	3	26	26	1	1
http://www.sdagri.gov.cn/ServicePlam/page/plam/index.jsp	12	12/18	118	109/155	0.9237	0.7032
http://www.jinzhong.com/	4	4/11	44	43	0.9773	1
http://www.grain.hl.cn/	1	1	24	24	1	1
http://nytg.nmagri.gov.cn/sites/nmgnyjs/	34	34	362	362	1	1
http://www.sdncp.com/	100	100/113	1 319	1302	0.9871	1
http://www.hn-wgagri.gov.cn/sites/MainSite/	5	5/7	507	481/587	0.9487	0.9194
Total	520	625/661	15 414	15 333/15 540	0.9752	0.9629

*, the site contains more than 3 000 page states, for considerable reason, the number of crawled page states was limited less than 100 for evaluation.

HTML tags and punctuations to segment entity items was used unlike the method in MDRII (Zhai and Liu 2005), because many AJAX contents just contain one entity that MDRII could not process effectively. DBSCAN algorithm was used to cluster the partitioned results, which show the concept attributes of ontology for annotating entity items are significant. The agricultural ontology used for filtering the clustering results and annotating the semantic of the entity items. Using agricultural ontology to extract items as indicator attributes of entity and to annotate the items get a promising result.

Compared with Tian (2009), using HTML tags to guide the information extraction, the proposed methods employ HTML tags and punctuations to partition entity, especially in agriculture domain, which is very effectively, for the reason that many entities, such as price and supply&demand entities, partition the items by punctuation rather than HTML tags.

CONCLUSION AND FUTURE WORKS

In this paper, a novel approach was proposed to exploit the structured AJAX contents of agriculture and perform entity extraction and semantic annotation. By applying agricultural ontology to the contents retrieval, most of the irrelevant contents are filtered out. The concepts and their attributes of agricultural ontology are important features when cluster the partitioned items. Agricultural ontology was employed to annotate the extracted entity items and assemble the items into semantic entity. The experiment results showed the proposed method can exploit, extract, and annotate structured information effectively. Although the method in this paper is utilized in agriculture domain, it is also feasible for other domain on condition that the agricultural ontology is replaced by the corresponding domain ontology.

This proposed method mainly works on structured extraction from dynamic AJAX contents. The future works will mainly focus on combing natural language processing with agricultural ontology to extract and annotate entity from free text.

Acknowledgements

This research was supported by the Knowledge Innovation Program of the Chinese Academy of Sciences and the National High-Tech R&D Program of China (2008BAK49B05).

References

Berson T A. 1993. Differential cryptanalysis mod 2^{32} with applications to MD5. In: *Proceedings of the 11th Annual International Conference on Theory and Application of Cryptographic Techniques*. Springer-Verlag Berlin, Heidelber. pp. 71-80.

Buitelaar P, Cimiano P, Frank A, Hartung M, Racioppa S. 2008. Ontology-based information extraction and integration from heterogeneous data sources. *International Journal of Human Computer Studies*, **66**, 759-788.

Carlson A, Schafer C. 2008. Bootstrapping information extraction from semi-structured web pages. *ECML/PKDD*, **5122**, 195-210.

Cohen W W, Hurst M, Jensen L S. 2002. A flexible learning

system for wrapping tables and lists in HTML documents. In: *Proceedings of the 11th International Conference on World Wide Web*. Honolulu, Hawaii, USA. pp. 232-241.

Crescenzi V, Mecca G, Merialdo P. 2001. Road runner: towards automatic data extraction from large web sites. In: *Proceedings of the 27th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. pp. 109-118.

Cui Y P. 2009. *Agricultural Ontology-Based Knowledge Management Key Technologies Research*. China Agricultural Science and Technology Press, Beijing. (in Chinese)

Diligenti M, Coetzee F, Lawrence S, Giles L, Gori M. 2000. Focused crawling using context graphs. In: *Proceedings of 26th International Conference on Very Large Data Bases*. Cairo, Egypt. pp. 527-534.

Ester M, Kriegel H, Sander J, Xu X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *The 2nd International Conference on Knowledge Discovery and Data Mining*. AAAI Press, California, USA. pp. 226-231.

Frey G. 2007. Indexing AJAX web applications. MSc thesis, Institute of Computational Sciences, Switzerland.

Garrett J. 2005. Ajax: a new approach to web applications. [2009-04-10]. http://adaptivepath.com/ideas/ajax-new-approach-web-applications

Gregg D, Walczak S. 2006. Adaptive web information extraction. *Communications of the ACM*, **49**, 78-84.

Huang Y Q, Cui W H, Zhang Y J, Deng G Y. 2012. Research on development of agricultural geographic information ontology. *Journal of Integrative Agriculture*, **11**, 865-877.

ICTCLAS. 2010. [2009-05-10]. http://ictclas.org/

Joachims T, Sander R, Ester M, Kriegel H P, Xu X. 1998. Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery*, **2**, 169-194.

JTidy. 2010. [2010-04-20]. http://jtidy.sourceforge.net

Li D, Zhao C. 2009. Computer and computing technologies in agriculture II. vol. 1. In: *Proceedings of the 2nd IFIP International Conference on Computer and Computing Technologies in Agriculture*. Springer, Beijing, China.

Liu L, Pu C, Han W. 2000. XWRAP: an XML-enabled wrapper construction system for web information sources. In: *Proceedings of the 16th International Conference on Data Engineering*. SanDiego, CA, USA. pp. 611-621.

Lukose D. 2012. World-wide semantic web of agriculture knowledge. *Journal of Integrative Agriculture*, **11**, 769-774.

Marchetto A, Tonella P, Ricca F. 2008. State-based testing of Ajax web applications. In: *Proceedings of the 1st IEEE International Conference on Software Testing, Verification and Validation*. Lillehammer, Norway. pp. 121-130.

MD5. 2010. [2010-02-10]. http://en.wikipedia.org/wiki/MD5

Menczer F, Pant G, Srinivasan P, Ruiz M E. 2001. Evaluating topic-driven web crawlers. In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New Orleans, Louisiana, USA. pp. 241-249.

Mesbah A, Bozdag E, Deursen A V. 2008. Crawling AJAX by inferring user interface state changes. In: *Proceedings of the 8th International Conference on Web Engineering*. New Jersey, USA. pp. 122-134.

Mesbah A, Deursen A V. 2007. An architectural style for Ajax. In: *Proceedings of the 6th Working IEEE/IFIP Conference on Software Architecture*. Mumbai, India. p. 9.

Mohan S. 2010. Indexing Web 2.0 applications. MSc thesis, Oregon State University, USA.

Qian P, Zheng Y L. 2006. *Agricultural Ontology Research and Application (fine)*. China Agricultural Science and Technology Press, Beijing. (in Chinese)

Rick C. 2000. Efficient computation of all longest common subsequences. In: *Proceedings of the 7th Scandinavian Workshop on Algorithm Theory Springer-Verlag*. London, UK. pp. 407-418.

Roest D, Mesbah A, Deursen A V. 2010. Regression testing ajax applications: coping with dynamism. In: *Proceedings of the 3rd International Conference on Software Testing, Verification and Validation*. Paris, French. pp. 127-136.

Russell S, Norvig P. 2002. *Artificial Intelligence: A Modern Approach*. Prentice Hall. New Jersey, USA.

Shchekotykhin K, Jannach D, Friedrich G. 2010. xCrawl: a high-recall crawling method for web mining. *Knowledge and Information Systems*, **25**, 303-326.

Song R, Liu H, Wen J, Ma W. 2004. Learning block importance models for web pages. In: *Proceedings of the 13th International Conference on World Wide Web*. New York, USA. pp. 203-211.

Tian X. 2009. Extracting structured data from Ajax site. In: *Proceedings of the 1st International Workshop on Database Technology and Applications*. Wuhan, China. pp. 259-262.

Yi W G, Yan L W, Liu Y Q, Liu Z. 2010. An ontology-based web information extraction approach. In: *Proceedings of the 2nd International Conference on Future Computer and Communication*. Wuhan, China. p. 1.

Wei Y Y, Wang R J, Hu Y M, Wang X. 2012. From web resources to agricultural ontology: a method for semi-automatic construction. *Journal of Integrative Agriculture*, **11**, 775-783.

Wimalasuriya D C, Dou D. 2010. Ontology-based information extraction: an introduction and a survey of current approaches. *Journal of Information Science*, **36**, 306-323.

Zhai Y, Liu B. 2005. Web data extraction based on partial tree alignment. In: *Proceedings of the 14th International Conference on World Wide Web*. Chiba, Japan. pp. 76-85.

Zhao H, Meng W, Yu C. 2006. Automatic extraction of dynamic record sections from search engine result pages. In: *Proceedings of the 32nd International Conference on Very Large Data Bases*. Seoul, Korea. pp. 989-1000.

(Managing editor WANG Ning)