2005 Society for Design and Process Science
Printed in the United States of America

AUTOMATIC DOMAIN ONTOLOGY GENERATION FROM WEB SITES*

Tak-Lam Wong
Wai Lam
Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong

Enhong Chen
Department of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, 230027, P.R. China

Ontology plays an important role in semantic Web technology since it can effectively represent the domain knowledge. We develop a novel framework for automatically generating the domain knowledge by analyzing different Web sites in a given domain. The idea of our approach is to consider two kinds of information from the Web sites. The first kind of information is the text fragments corresponding to the concepts in the ontology. The other kind of information is the header labels corresponding to the concepts. We design a method for generating the domain ontology by measuring the similarity between the concepts in different Web sites. We have conducted extensive experiments to demonstrate the effectiveness of our approach.

Keywords: *ontology mining, Web mining, semantic Web.*

1. Introduction

Semantic Web is considered as the next generation of the World Wide Web. The goal of semantic Web is to describe Web resources with well-defined and machine-understandable meaning. Ontology plays a vital role in semantic Web because it defines the conceptual meanings and their relationship in a given domain. Previously, ontology has been adopted in various research areas such as bioinformatics (Stevens *et al.*, 2002). Due to the terminological discrepancies between different biological repositories, data cannot be shared easily. Gene Ontology attempts to solve this problem by introducing a set of controlled vocabularies for annotating a gene product with its molecular functions, the biological process in which it is involved, and the cellular locations in which it is found (The Gene Ontology Consortium, 2000).

Ontology is regarded as a conceptual hierarchy of the domain since it is normally expressed as a tree-like structure. For example, Figure 1 shows a sample of a Web page from a Web site about book catalog1. There are several book records in this page. Each book record contains concepts such as

*The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Nos: CUHK 4179/03E and CUHK4193/04E), the Direct Grant of the Faculty of Engineering, CUHK (Project Code: 2050363), and CUHK Strategic Grant (No: 4410001). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

"title", "author", "published", "list price", "you save", and "our price". Figure 2 depicts a sample ontology representing a book in this Web site. In this ontology, there is a root node called "book". The internal nodes, such as "title", represent different concepts associated with a book. "List price", "you save", and "our price" are the sub-concepts of the concept "price".

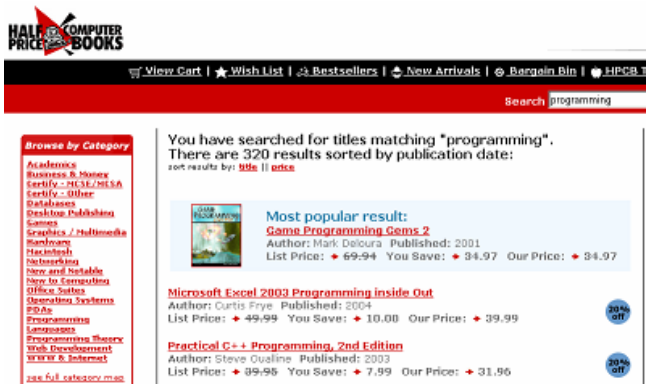

Fig. 1 An example of Web page about book catalog.

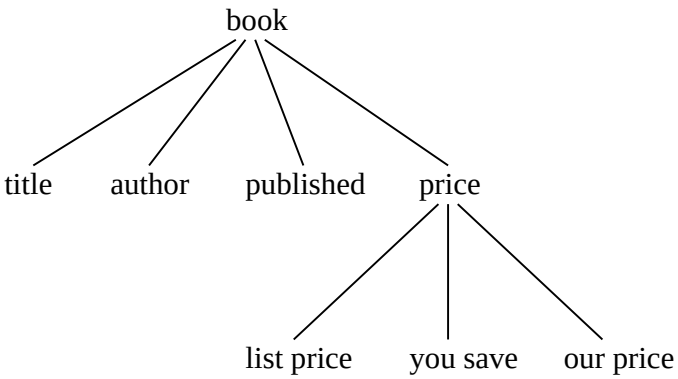

Fig. 2 The ontology describing the relationship between the concepts of a book in the Web page shown in Figure 1.

Manual effort is required to construct a specific ontology for a given site. This task becomes tedious, error-prone, and requires a high level of expertise, when a large number of different sites must be dealt with. Recently, several research groups attempted to apply machine learning techniques for ontology learning, in order to reduce human effort in ontology construction (Maedche *et al.*, 2001, Navigli *et al.*, 2003, Tijerino *et al.*, 2003). Some of these techniques are semi-automatic and require interactions with the users during the construction process. Some methods can only handle data in a specified format such as tables in Web pages. This poses limitations in current ontology learning techniques. Moreover, the ontology constructed for a particular Web site may not effectively apply to another Web site, even in the same domain. Consider the Web page shown in Figure 32. It is collected from a Web site different from the one shown in Figure 1. Figure 4 depicts the ontology describing the concepts of a book in this Web site. Although both ontologies in Figures 2 and 4 define a book, there are several differences. First, some concepts such as "published" and "ISBN" are present in only one of

the ontologies. Second, "list price" in Figure 2 and "MSRP" in Figure 4 both refer to the same concept, but in different terminology. Therefore, the ontology constructed for one Web site typically cannot be reused in another Web site. A separate effort is required to construct a specific ontology for the new site. One possible solution to this problem is to construct a general domain ontology which contains all the concepts and the associated terminologies. For instance, Figure 5 depicts the ontology which can describe the books in both Figures 1 and 3. However, the construction of such general ontology becomes difficult when there are numerous individual ontologies from different Web sites.

Fig. 3 An example of Web page about book catalog collected from a different Web site shown in Figure 1.

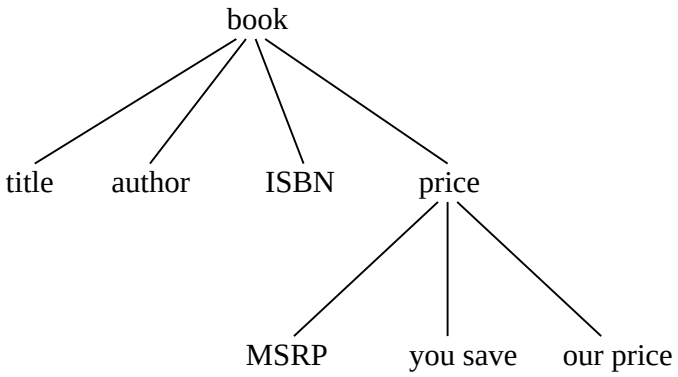

Fig. 4 The ontology describing the relationship between the concepts of a book in the Web page shown in Figure 3.

We develop a framework in which users can provide training examples from one particular source Web site and automatically construct the general domain ontology in a domain. For example, suppose the source Web site is the one shown in Figure 1, which is associated with the ontology shown in Figure 2. Our framework can automatically refine this existing ontology to suit the new site as shown in Figure 3. The resulting ontology after the refinement will be the one depicted in Figure 4. Next, the

two ontologies in Figures 2 and 4 are considered to generate the domain ontology depicted in Figure 5. The generation of domain ontology is achieved by considering two kinds of information in the Web sites. The first kind of information is the text fragments corresponding to the content of the concepts in the ontologies. For example, text fragment samples corresponding to the content of the concept "title" on the Web page in Figure 1 include "Game Programming Gems 2", "Microsoft Excel 2003 Programming Inside Out", and "Practical C++ Programming 2nd Edition". These text fragments can be easily collected or extracted by using automatic information extraction methods such as wrappers (Cohen, *et al.*, 2002, Kushmerick and Thomas, 2002, Wong and Lam, 2004). Since text fragments from the same concept in different Web sites contain similar characteristics, they are useful in generating the concepts contained in the domain ontology. The second kind of information is the header labels associated with the concepts in the Web sites. For instance, the text fragments "You save" and "Our price" are the header labels in the Web sites shown in Figures 1 and 3. These header labels provide a very useful clue in identifying the concept in the domain ontology, because the same concept in different ontologies is normally associated with similar header labels in different Web sites. Our framework analyzes this information to construct the domain ontology using different individual ontologies from different Web sites.

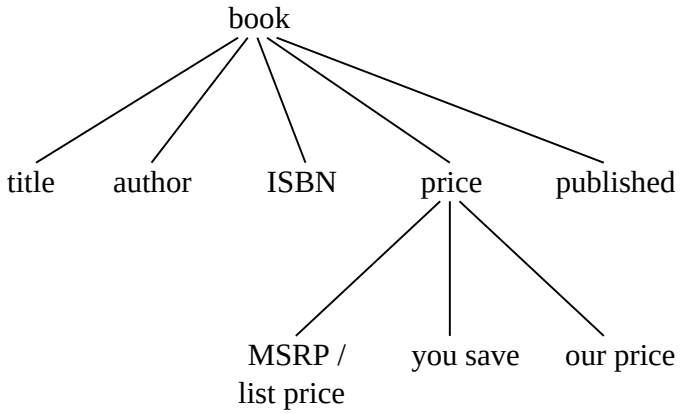

Fig. 5 The general ontology describing the relationship between the concepts of a book in the Web pages shown in Figures 1 and 3.

2. Related Work

Ontology plays an important role in semantic Web (World Wide Web Consortium, 2001) since it provides a way to express the meaning and knowledge contained in the Web resources, such as Web pages. With the semantic Web, software agents can then share knowledge in the Web. Another application of ontology is in the area of bioinformatics. Gene Ontology and RiboWeb are two examples for applying ontologies to semantically describe the knowledge in bioinformatics resources (Altman *et al.*, 1999, The Gene Ontology Consortium, 2000). Stevens *et al.* proposed to use the ontology language DAML+OIL to construct the bioinformatics concepts (Stevens and Goble, 2002). The constructed ontology can support inference and be shared in the semantic Web.

Various semi-automatic methods are proposed to reduce the human work in ontology construction (Maedche and Staab 2001, Navigli *et al.*, 2003, van der Vet and Mars, 1998). However, they all require user interaction during the construction process. A system known as TANGO (Tijerion *et al.*, 2003) attempts to semi-automatically generate the ontology from data in table format. In TANGO, an ontology engineer first constructs a seed ontology within the system. Then, the system analyzes the

content of each table, such as its caption, and designates value pairs in the Web page to form a mini-ontology. Next, the set of mini-ontologies will be integrated into the seed ontology. One limitation of TANGO is that the data must be in table format.

Another common shortcoming for the above approache is that the ontology constructed can only represent a particular Web site. If we want to construct the ontology for a new Web site, a separate effort is required. Maedche *et al.* investigated the problem of ontology reuse (Maedche *et al.*, 2003). They discovered that the ontology defined for one Web site may not be applicable to another site. They proposed a framework to solve this problem by providing a mathematical model to retain the consistency in the reused ontology. They also developed a tool for managing the ontologies from different Web sites. However, their approach still requires a considerable amount of human effort in practice. Doan *et al.* proposed a method to solve the ontology matching problem which aims at matching the concepts of two ontologies (Doan *et al.*, 2003). For example, the concept "associate professor" in one ontology may be equivalent to the concept "senior lecturer" in another ontology. However their approach requires human effort to prepare training documents in each concept. Their objective is also different from ontology construction or refinement.

3. Domain Ontology Generation

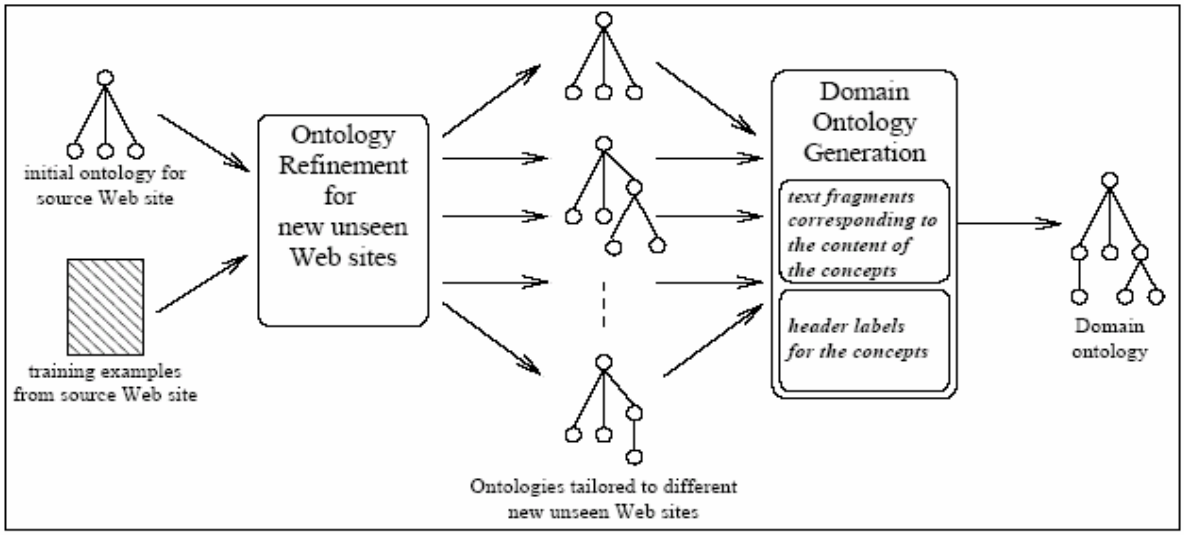

Fig. 6 An overview of our method of generating domain ontology.

The objective of our approach is to generate the desired domain ontology based on an initial ontology and training examples from a particular source Web site. The idea behind our approach is to first refine the ontology from the source Web site to obtain different ontologies, tailored for other new sites in the given domain. Next, individual ontologies from different Web sites are analyzed and two kinds of information from the Web sites are considered to generate the domain ontology. The first kind of information is the text fragments corresponding to the content of the concepts in the ontologies. Since the text fragments corresponding to the same concept share some common characteristics in different Web sites, we can analyze the content of these text fragments and decide whether the nodes from different ontologies refer to the same concept. The second kind of information is the header labels of the concepts in different Web sites. The header labels collected from different Web sites are similar

if they are related to the same concept. Therefore, the header labels provide very important information in domain ontology generation. We develop a two phase approach for generating the domain ontology. Figure 6 shows the overview of our approach. The first phase is the ontology refinement phase based on our previous work (Wong and Lam, 2005). The goals of this phase are to analyze the ontology and the text fragments associated with concepts from the source Web site, and to generate the refined ontologies tailored to the new unseen sites. This phase considers the clues from the text fragments and corresponding concepts in the ontology, the header labels, and the visual layout of the Web sites. The refined ontologies are tailored for the corresponding new Web sites and can precisely represent the information of the concepts in the new sites. In this paper, we focus on the second phase of our approach. Readers can refer to (Wong and Lam, 2005) for the details of the first phase, which is ontology refinement.

The second phase is the domain ontology generation phase. The goal of this phase is to construct the domain ontology based on the ontologies from different Web sites in the same domain. This phase considers the information from the content of the text fragment corresponding to the concepts and the information from the header labels. The goal of this phase is to compute the similarity between the concepts in the ontologies. If the concepts from different ontologies are similar, they are regarded as the same concept and become a single concept in the final domain ontology. We design a two-level edit distance between two text fragments. A scoring method is then developed, based on the two-level edit distance, to determine the similarity between the concepts in the ontologies.

As mentioned above, the first phase of our approach generates the tailored ontology for each of the hypothetical Web sites. A set of text fragments and header labels corresponding to each of the concepts in the ontology will also be automatically extracted in this phase. For example, the text fragments "Scott Urman" and "Author" are a sample text fragment and a sample header label corresponding to the concept "author" in the ontology shown in Figure 4. The domain ontology is then constructed based on these text fragments and header labels from different Web sites. Our method is designed based on the two-level edit distance between two text fragments. We define the two-level edit distance as follows:

Definition 3.1 Suppose there are two tokens t_1 and t_2 containing p and q characters respectively, the character-level edit distance between the tokens is the minimum cost of transforming t_1 to t_2 by inserting, deleting or modifying the characters in t_1, where the cost of insertion, deletion, and modification of a character is 1. The normalized character-level edit distance is the character-level edit distance divided by $\max(p,q)$.

Definition 3.2 Suppose f^1 is a text fragment containing the sequence of tokens $t^1_1, t^1_2, \ldots, t^1_m$ and f^2 is another text fragment containing the sequence of tokens $t^2_1, t^2_2, \ldots, t^2_n$. We define the token-level edit distance, $D(f^1,f^2)$, between the text fragments f^1 and f^2 as the minimum cost of transforming f^1 to f^2 by inserting, deleting or modifying the tokens in f^1, where the cost of insertion, deletion of a token is 1, and the cost of modification of a token t^1_i to t^2_j is the normalized character-level edit distance between t^1_i and t^2_j. The normalized token-level edit distance is the token-level edit distance divided by $\max(m,n)$.

Both the character-level, and token-level edit distance, can be computed efficiently using a dynamic programming technique (Gusfield, 1997). This token-level edit distance can effectively represent the similarity between the two text fragments.

Figures 7 and 8 depict the outline of our automatic domain ontology generation algorithm and the associated score function, respectively. In the automatic domain ontology generation algorithm, we first randomly select one Web site from the set of K Web sites. The ontology associated with the selected Web site is regarded as the seed ontology. This seed ontology will "grow" and become the

domain ontology. Next, the distance between a particular concept, *c*, in the seed ontology and another concept, *c′*, in the ontologies from the remaining sites is computed using the *score* function depicted in Figure 8. If the distance is smaller than a pre-defined threshold θ, then *c* and *c′* are regarded as the same concept and are merged by the *merge* function. The *merge* function effectively updates the set of text fragments corresponding to the concept *c* by adding the text fragments corresponding to the concept *c′*. If the distance is larger than θ, then *c′* is regarded as a different concept from *c*, and it will be added to the seed ontology. Subsequently, *c′* will be removed from the associated ontology. After processing all the concepts in the ontologies, the "grown" seed ontology is returned as the domain ontology.

 # Automatic domain ontology generation algorithm
Input: Ontologies from K Web sites
Text fragments corresponding to the concepts in the ontologies collected in the Web sites
Header labels corresponding to the concepts in the ontologies collected in the Web sites
Output: The desired domain ontology

Algorithm
1. Randomly select one Web site ω from the K Web sites
2. foreach concept c in the ontology $O(\omega)$ associated with the Web site ω
3. foreach ontology $O(\overline{\omega})$ associated with the remaining Web sites
4. $dist = \min_{c' \in C(O(\overline{\omega}))} \text{score}(c, c')$
where $C(O(\overline{\omega}))$ is the set of concepts contained in the ontology $O(\overline{\omega})$
5. $\hat{c} = \arg\min_{c' \in C(O(\overline{\omega}))} \text{score}(c, c')$
6. if $(dist < \theta)$
7. merge($\hat{c}$, c, $O(\omega)$)
8. else
9. add($\hat{c}$, $O(\omega)$)
10. delete($\hat{c}$, $O(\overline{\omega})$)
11. return $O(\omega)$

Figure. 7 The outline of the automatic domain ontology generation algorithm.

 # Score function used in automatic domain ontology generation
Input: Concepts c and c' from two different ontologies
The set of text fragments F_c corresponding to the concept c
The set of text fragments $F_{c'}$ corresponding to the concept c'
The header labels h_c corresponding to the concept c
The header labels $h_{c'}$ corresponding to the concept c
Output: A score reflecting the similarity between the concepts c and c'

Algorithm
1. $dist_1 = \frac{1}{|F_c|} \sum_{f_c \in F_c} \min_{f_{c'} \in F_{c'}} D(f_c, f_{c'})$
2. $dist_2 = D(h_c, h_{c'})$
3. $dist = \alpha dist_1 + (1 - \alpha) dist_2$
4. return $dist$

Figure. 8 The outline of the score function used in automatic domain ontology generation.

The *score* function takes into account two pieces of information to determine the distance between the two ontologies. As mentioned before, each concept is associated with a set of text fragments and header labels corresponding to that concept. One piece of data considered in the *score* function is the

text fragments corresponding to the concepts c and c'. Suppose F_c and $F_{c'}$ are sets of text fragments corresponding to the concept c and c', respectively. For each element in F_c, the *score* function first finds the nearest neighbor in $F_{c'}$ using the normalized token-level edit distance as the distance measure. Also, $dist_1$ is the average distance between the elements in F_c and their nearest neighbor in $F_{c'}$. Other information considered is the header labels corresponding to the concepts c and c', where $dist_2$ is the normalized token-level edit distance between the header labels h_c and $h_{c'}$ associated with c and c', respectively. Finally, the distance between the two concepts is the weighted sum of $dist_1$ and $dist_2$ with the pre-defined weight α.

4. Experimental Results

	Web site (URL)
S1	Half Price Computer Books (http://www.halfpricecomputerbooks.com)
S2	Discount-PCBooks.com (http://www.discount-pcbooks.com)
S3	mmistore.com (http://www.mmistore.com)
S4	Amazon.com (http://www.amazon.com)
S5	1Bookstreet.com (http://www.1bookstreet.com)
S6	Barnes & Noble.com (http://www.barnesandnoble.com)
S7	bookpool.com (http://www.bookpool.com)
S8	half.com (http://half.ebay.com)
S9	DigitalGuru Technical Bookshops (http://www.digitalguru.com)

Table. 1 Web sites in the book catalog domain collected for experiments.

We conducted experiments on several real-world Web sites in the book catalog domain to demonstrate the performance of our automatic ontology generation framework. Table 1 shows the Web sites used in our experiment. The first column shows the Web site label, and the second column shows the name of the Web sites and the corresponding URL.

To evaluate the performance of our ontology refinement framework, we first manually construct the domain ontology of the book catalog domain. This manually constructed ontology is considered the gold standard for evaluation. Also, our automatic ontology generation method was conducted to generate the domain ontology, using the provision of ontology and training examples from a source Web site. For example, the user provides the ontology and the text fragments corresponding to the concepts from S1, and our automatic ontology generation method is applied to generate the domain ontology. The resulting ontology is then compared with the manually constructed domain ontology for evaluation. We evaluate the performance by calculating the *tree edit distance* between the automatic generated ontology and the manually constructed ontology. The tree edit distance is defined as the minimum cost of an edit operation sequence that transforms one tree to the other. There are three kinds of edit operations. The first operation is to change the label of a node φ. The second operation is to delete a node φ, and make its children become the children of the original parent of φ. The third operation is to insert a node φ as the child of another node φ, and make any child become the child of φ. We fix the costs of all these edit operations to *1*. The smaller the tree edit distance between the two

ontologies, the greater their similarity is. Readers can refer to (Shasha and Zhang, 1997) for the details of the tree edit distance.

Table 2 shows the results of comparing the automatic generated ontology with the manually constructed domain ontology. The first column shows the Web sites (source sites) from which the ontologies and training examples are provided. The second and third column depict the edit distance between the automatic generated ontology and the manually constructed ontology (ε), and the edit distance between the automatic generated ontology and the manually constructed ontology, normalized by the total number of concepts (ε'). Note that the smaller the distance, the better is the performance. The results indicate that our framework achieves a very satisfactory result in discovering the structure of the ontology.

	ϵ	ϵ'
S1	1.00	0.08
S2	1.00	0.08
S3	0.00	0.00
S4	1.00	0.08
S5	1.00	0.08
S6	0.00	0.00
S7	1.00	0.08
S8	0.00	0.00
S9	0.00	0.00

Table. 2 Performance of our automatic ontology generation framework on generating the domain ontology in the book catalog domain. ε refers to the tree edit distance between the automatic generated ontology and the manually constructed ontology in the domain. ε' refers to the tree edit distance between the automatic generated ontology and the manually constructed ontology normalized by the total number of concept in the domain. (Note that the smaller the distance, the better is the performance.)

5. Conclusions and Future Work

We have developed a two phase framework for generating the domain ontology with the provision of an initial ontology and training examples from a particular source Web site. The first phase of our framework can effectively generate the ontologies tailored for different Web sites. These ontologies from different Web sites are then analyzed in the second phase to automatically generate the domain ontology. This phase mainly considers two kinds of information. One kind of information is the text fragment corresponding to the concepts in the ontology. Since the text fragments corresponding to the same concept from different Web sites share some similar characteristics, they can help identify the concepts contained in the domain ontology. Another kind of information is the header labels corresponding to the concepts. The same concept from different Web sites is normally associated with similar header labels. This provides a very useful clue to identifying concepts. We have designed a two-level edit distance to express the similarity between two text fragments. The domain ontology is generated based on the two-level edit distance technique. We have conducted extensive experiments to demonstrate the effectiveness of our approach. The results demonstrate that our approach can effectively reduce the human effort involved in constructing the domain ontology.

We intend to extend our framework in several directions. One possible direction is to incorporate the domain knowledge of the users. Oftentimes, users have some knowledge about the ontology, such as some constraints between concepts. This domain knowledge is useful in constructing a precise and

expressive ontology. Another possible direction is to apply the automatically generated ontology in multi-agent system for querying between agents. Since the ontology contains the domain knowledge, it is helpful in the query system.

6. References

R. Altman, M. Bada, X. Chai, M. Carillo, R. Chen, and N. Abernethy, 1999, "Riboweb: An ontology-based system for collaborative molecular biology," IEEE Transactions on Intelligent Systems, Vol. 14, No. 5, pp. 68-76.

W. Cohen, M. Hurst, and L. Jensen, 2002, "A flexible learning system for wrapping tables and lists in HTML documents," Proceedings of the Eleventh International World Wide Web Conference, Budapest, Hungary, May 7-11, pp. 232-241.

A. Doan, J. Madhavan, R. Dhamanker, P. Domingos, and A. Halevy, 2003, "Learning to match ontologies on the semantic web," The VLDB Journal, Vol. 12, No. 4, pp. 303-319.

D. Gusfield, 1997, Algorithms on Strings, Trees, and Sequences, Cambridge University Press.

N. Kushmerick and B. Thomas, 2002, "Adaptive information extraction: Core technologies for information agents," Intelligent Information Agents R&D in Europe: An AgentLink Perspective, pp. 79-103.

A. Maedche, B. Motik, and L. Stojanovic, 2003, "Managing multiple and distributed ontologies on the semantic web," The VLDB Journal, Vol. 12, No. 4, pp. 286-302.

A. Maedche and S. Staab, 2001, "Ontology learning for the semantic web," IEEE Intelligent Systems, Vol. 16 No. 2, pp. 72-79.

R. Navigli, P. Velardi, and A. Gangemi, 2003, "Ontology learning and its application to automated terminology translation," IEEE Intelligent Systems, Vol. 18, No. 1, pp. 22-31.

D. Shasha and K. Zhang, 1997, Pattern Matching Algorithms; Oxford University Press.

R. Stevens, C. Goble, I. Horrocks, and S. Bechhofer, 2002, "OILing the way to machine understandable bioinformatics resources," IEEE Transactions on Information Technology in Biomedicine, Vol. 6, No. 2, pp. 129-134.

The Gene Ontology Consortium, 2000, "Gene ontology: Tool for the unification of biology," Nature Genetics, Vol. 25, No. 1, pp. 25-29.

Y. Tijerino, D. Embley, D. Lonsdale, and G. Nagy, 2003, "Ontology generation from tables," Proceedings of the Forth International Conference on Web Information Systems Engineering, Rome, Italy, pp. 242-249.

P. van der Vet and N. Mars, 1998, "Bottom-up construction of ontologies," IEEE Transaction on Knowledge and Data Engineering, Vol. 10, No. 4, pp. 513-525.

T. L. Wong and W. Lam, 2004, "A probabilistic approach for adapting information extraction wrappers and discovering new attributes," Proceedings of the Fourth 2004 IEEE International Conference on Data Mining, Brighton, United Kingdom, November 1-4, pp. 257-264.

T. L. Wong and W. Lam, 2005, "Learning to refine ontology for a new web site using a Bayesian approach," Proceedings of the 2005 SIAM International Conference on Data Mining (SDM-2005), Newport Beach, California, pp. 298-309.

World Wide Web Consortium (W3C), 2001, "Semantic web," In http://www.w3.org/2001/sw/.