

# Learning to Refine Ontology for a New Web Site Using a Bayesian Approach\*

Tak-Lam Wong and Wai Lam

Department of Systems Engineering And Engineering Management

The Chinese University of Hong Kong

Hong Kong

{wongtl,wlam}@se.cuhk.edu.hk

**Keywords:** Text Mining, Knowledge Adaptation, Ontology Refinement

## Abstract

We develop a probabilistic framework which can refine an existing ontology from a source Web site to new unseen sites. One characteristic of our framework is to consider several clues related to how an ontology influences the text content and the visual layout of the Web pages. The first clue is the text fragments regarding the content of the concepts previously collected or extracted from the source Web site. The second clue is the text fragments regarding the header labels of the concepts in the unseen site. To harness the uncertainty involved in a rigorous manner, we formalize these clues by a generative model to represent the generation of text fragments regarding the concepts and the ontology corresponding to the Web page. Bayesian learning technique and expectation-maximization (EM) algorithm are employed to accomplish the task. Extensive experiments on several real-world Web sites from two different domains have been conducted to demonstrate the effectiveness of our framework.

## 1 Introduction

Recently, ontology has become one of the important research topics in Web technology. This is mainly due to the emergence of semantic Web which attempts to give well-defined and machine-understandable meaning to Web resources for a specified domain [20]. Ontology is also employed in various applications. For example, it can be used to enhance the performance of Web personalization [6]. Web personalization is to intelligently provide users with tailor-made information. By annotating the Web page content semantically with a conceptual hierarchy, precise Web content can be processed in a flexible manner. Ontology is also used extensively in bioinformatics [13]. Due to the terminological difference in different biological repositories, data cannot

\*The work described in this paper was substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Nos: CUHK 4187/01E, CUHK 4179/03E, and CUHK 4193/04E) and CUHK Strategic Grant (No: 4410001).

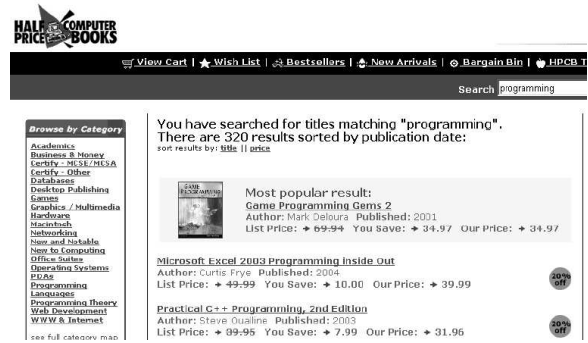


Figure 1: An example of Web page about book catalog.

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 [14].

Ontology defines the concepts and their relationships in a given domain. It is also 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 catalog<sup>1</sup>. There are several book records in this page. Each book record contains concepts such as “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 concepts associated with a book. “List price”, “you save”, and “our price” are the subconcepts of the concept “price”.

Manual effort is required to construct an ontology for a given site. However, if we need to deal with a large number of different sites, this task becomes tedious, error-prone, and requires high level of expertise. Recently, several research groups attempted to reduce human effort in ontology construction by applying machine

<sup>1</sup>The URL of the Web site is <http://www.halfpricecomputerbooks.html>.

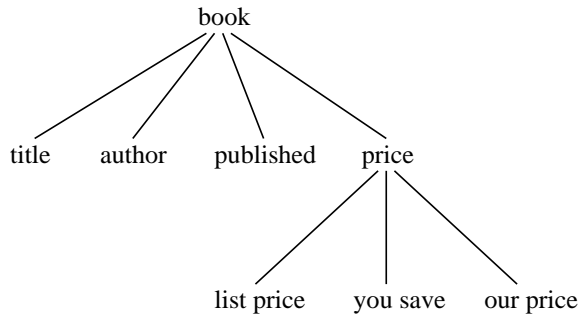


Figure 2: The ontology describing the relationship between the concepts of a book in the Web page shown in Figure 1



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

learning techniques for ontology learning [9, 11, 15]. Of these proposed methods, some of them require interactions with the users during the construction process. Some methods require that the raw text data is organized 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 be able to effectively apply to another Web site even in the same domain. Consider the Web page shown in Figure 3<sup>2</sup>. 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 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 an ontology for the new site.

We develop a probabilistic framework which can

<sup>2</sup>The URL of the Web site is <http://www.discount.com>.

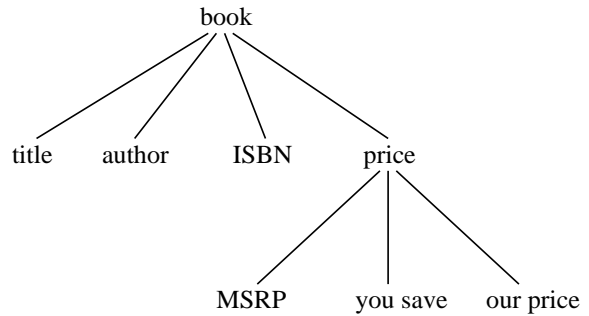


Figure 4: The ontology describing the relationship between the concepts of a book in the Web page shown in Figure 3

automatically refine an existing ontology tailored to a new unseen Web site in the same 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 unseen site as shown in Figure 3. The resulting ontology after the refinement will be the one depicted in Figure 4. This is achieved by considering several clues related to how the ontology (concept and structure) influences the text content and visual layout of the Web pages. The first clue is the text fragments corresponding to the content of the concepts previously collected or extracted from the source Web sites. 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 [3, 7, 17]. Our framework analyzes the characteristics of these text fragments. As the Web sites belong to the same domain, the content of the text fragments regarding the same concept shares some characteristics and provides useful information to identify the same concept in the new unseen site.

The second clue is the text fragments regarding the header labels of the concepts. The text fragments “Author”, “List Price”, and “You Save” in Figure 1 are examples of header labels of the concepts “author”, “list price”, and “you save” respectively. The header labels are useful because their semantic meaning and orthographic features can indicate the relationship between the concepts. For example, the words “Save” and “Price” are semantically related. Hence, it is likely that the concepts “list price” and “you save” are subconcepts of the concept “price”.

The third clue we consider is the visual layout of the text fragments regarding the content of concepts and the header labels of the concepts. The visual layout

provides information in two aspects. One aspect is the association between the content and the header label of a particular concept. For example, the header label “Author” in Figure 3 is bolded and is located on the left to the text fragments regarding the content of “author”. We can infer that similar patterns in the Web page are probably an indication of new or existing concepts. Another aspect is the location of the concepts relative to each other. We observe that concepts related to each other are normally located in a nearby position. For example, the concepts “list price”, “you save”, and “our price” in Figure 1 are located within the same row.

The clues described above involve uncertainty. To cope with the problem in a formal manner, we develop a generative model to represent the generation of text fragments regarding the concepts and the ontology corresponding to the Web page. Bayesian learning technique and expectation-maximization (EM) algorithm are then employed to achieve this task.

## 2 Related Work

Ontology plays an important role in semantic Web [20] 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 [1, 14]. Stevens et al. proposed to use the ontology language DAML+OIL to construct the bioinformatics concepts [13]. The constructed ontology can support inference and be shared in the semantic Web.

In spite of the increasing popularity of ontology in expressing knowledge, ontology construction is a tedious task and requires high level of expertise. Some methods have been proposed to reduce the human effort in constructing ontology. van der Vet and Mars proposed a bottom-up methodology for ontology construction [16]. Their idea is to perform generalization to form some concepts from the primitive concepts. Maedche and Staab developed a system with graphical user interface for users to easily construct ontologies [9]. OntoLearn [11] is another system to assist ontology construction. It makes use of natural language processing and machine learning techniques to extract terms in domain texts. The semantic meaning of the extracted term is interpreted by WordNet and the ontology is enriched by these extracted terms. The resulting ontology can then be edited, validated by other ontology management tools.

All the methods proposed above aim at reducing

the human effort in ontology construction. However, they all require user interaction during the construction process. A system known as TANGO [15] attempts to generate the ontology from data in table format in a semi-automatic fashion. In TANGO, an ontology engineer first constructs a kernel ontology to the system. Then, the system analyzes the content of each table such as its caption, attribute-value pairs in the Web page to form a mini-ontology. Next, the set of mini-ontologies will be integrated into the kernel ontology. One limitation of TANGO is that the data must be in table format.

Another common shortcoming for the above approaches 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 [8]. They found out 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 [5]. 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 Overview of Our Framework

The rationale of our framework is to consider several clues concerned with how the ontology affects the text content and visual layout of the Web pages. The first clue is the text fragments regarding the content of the concepts. For example, “Stephen Randy Davis” is a text fragment regarding the concept “author” in the Web page shown in Figure 1. This text fragment possesses characteristics of the content such as the first character in each word is capitalized. These characteristics help identify the text fragments corresponding to the content of similar concepts in the new unseen Web page. The second clue is the text fragments regarding the header label of the concepts. For instance, “You Save” is an example of a header label in Figure 1. The word “save” indicates that this header label is related to the concept “price” of a book. Similarly, the header labels “List Price” and “Our Price” of the concepts “list price” and “our price” respectively are also related to the concept

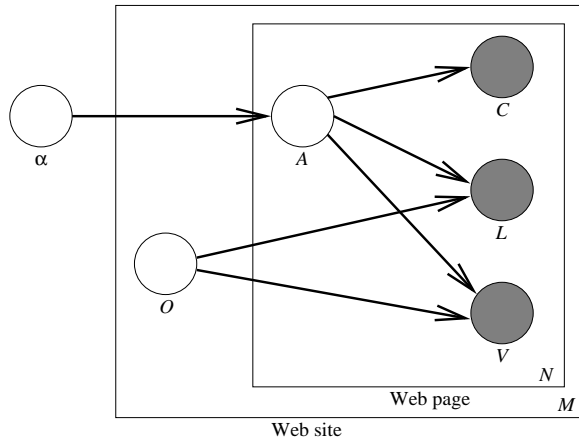


Figure 5: The generative model for generation of text fragments regarding the concepts from a domain ontology.  $\alpha$  denotes the concept generation parameter.  $A$  and  $O$  denote the concept and the ontology respectively.  $C$  and  $L$  denote the content and the header label of a concept respectively.  $V$  denotes visual layout information.  $N$  is the number of text fragments regarding the concepts within the Web page and  $M$  is the number of Web pages in the Web site.

“price”. Therefore, it increases the likelihood that these concepts should be the subconcepts of the concept “price”. The third clue we consider is the visual layout of the text fragments regarding the content of concepts and the header labels of the concepts. One kind of visual layout information is the association between the content and the header label of a particular concept. The second kind of visual layout information is the location of the concepts relative to each other. Visual layout information helps discover new concepts in the unseen site. For example, suppose we wish to induce an ontology for the new Web page in Figure 3 given an existing concept “title” and some book title names previously collected or extracted from a site such as the one in Figure 1. Based on this evidence, the book title content on the new Web page can be identified. Therefore, the concept “title” can be induced for the new site. Similarly the existing concept “author” can be induced for the new site. Since the header label “Author” of the concept “author” on the new Web page makes use of certain font and boldness, it can be inferred that the text fragments “ISBN” on the new Web page is likely to be the header label of a new concept. Another example is that the concepts “MSRP”, “your price”, and “you save” are located in the same row in Figure 3. It increases the likelihood that these three concepts are the subconcepts of the concept “price”.

To harness the uncertainty involved in a rigorous manner, we investigate a generative model for the generation of text fragments regarding the concepts in a Web page. Figure 5 illustrates the graphical representation of our model. Shaded nodes and unshaded nodes

refer to observable variables and unobservable variables respectively. The arrows represent the dependence of the variables. Precisely, the variables at the head depend on the variables at the tail. In a particular domain, there is a concept generation parameter called  $\alpha$ . This parameter is domain dependent and site invariant because the Web sites collected in the same domain contain similar kinds of concepts. This parameter controls the concepts, denoted by  $A$ , contained in the Web pages. For each of the  $N$  Web pages of the  $M$  Web sites, the concepts contained in the Web pages are generated by a certain probability distribution dependent on  $\alpha$ , i.e.,  $P(A; \alpha)$ . For example, the Web page in Figure 1 contains the concepts “title”, “author”, “published”, “list price”, “you save”, and “our price”. The content of a concept, denoted by  $C$ , is then generated according to the probability distribution  $P(C|A)$ . For instance, the content of “title” in the first record in Figure 1 is “Game Programming Gems 2”, which is generated from  $P(C|A = \text{“title”})$ . Since  $C$  is dependent on  $A$  which in turn depends on  $\alpha$ ,  $C$  is domain dependent and site invariant. Therefore, the text fragments regarding the content of the concepts are similar in the Web pages of the Web sites from the same domain. Another parameter in our model is the ontology associated with the Web site and it is denoted by  $O$  in Figure 5. This parameter is site dependent and influences the header label and the visual layout of a concept in the Web pages of a particular Web site.

The header label and the visual layout are denoted by  $L$  and  $V$  in Figure 5 respectively. The header labels are generated according to the probability distribution  $P(L|A; O)$ . The ontology also affects the visual layout of a Web page. For example, the concepts “list price”, “you save”, and “our price” are likely to be the subconcepts of “price” and they are located in the same row in the Web page. The visual layout of the concepts is generated by the probability distribution  $P(V|A; O)$ .

Based on this generative model, we can derive the following joint probability:

$$P(C, L, V, A; \alpha, O) = P(C|A)P(L|A; O)P(V|A; O)P(A; \alpha) \quad (3.1)$$

Using the above notation, a text fragment within a Web page can be represented by the attributes  $C$ ,  $L$ , and  $V$ . To predict the concept of a particular text fragment, we compute the probability  $P(A|C, L, V; \alpha, O)$ . From Equation 3.1, we can derive that:

$$P(A|C, L, V; \alpha, O) = \frac{P(C|A)P(L|A; O)P(V|A; O)P(A|\alpha)}{\sum_A P(C|A)P(L|A; O)P(V|A; O)P(A|\alpha)} \quad (3.2)$$

In practice, the text fragments regarding to the con-

tent of concepts in the new unseen site are required to be identified in advance. We called it *informative text fragment*. We develop an information-theoretic approach to analyze the Document Object Model<sup>3</sup>(DOM) structure of the Web pages of the unseen Web site. The informative nodes in the DOM structure can be identified. The text fragments within the informative nodes become the informative text fragments.

The parameters of our probabilistic model is inferred from a set of informative text fragments regarding the concepts collected from the source Web site and the new unseen site. The text fragments collected from the source Web site are useful because their content share some characteristics with those text fragments in the new unseen site. Our framework can conduct this inference process to identify the concepts contained in the unseen site. For the informative text fragments collected in the new unseen site, their concepts are uncertain and cannot be used for inference directly. By modeling a concept ( $A$ ) by an unobservable variable in the generative model, we can employ expectation-maximization (EM) algorithm in the inference of the parameters  $\alpha$  and  $O$ . We derive several content features and make use of the text fragments collected from both of the source Web sites and the new unseen sites to estimate  $P(C|A)$  in Equation 3.2. We can estimate  $P(L|A; O)$  and  $P(V|A; O)$  from the text fragments collected from the new unseen site since  $O$  is site dependent. However, the parameter  $O$  cannot be directly obtained because the text fragments collected do not come with a complete and correct ontology for the new unseen site. Potential ontology candidates are constructed by considering the concepts of the text fragments, the semantic meaning, and the orthographic features of the header labels. We design a metric to estimate  $P(L|A; O)$  and  $P(V|A; O)$  based on the relationship between each of the potential ontology candidates, the conditional probability  $P(L|A)$ , and the relationship between  $P(V|A)$ . The ontology can then be refined iteratively to adapt to the new unseen site in our EM framework. The learned model and the refined ontology can then be applied to compute the probability  $P(A|C, L, V; \alpha, O)$  of each text fragment in the new unseen site.

#### 4 Bayesian Learning Approach with EM

In our framework, the informative text fragments in the new unseen Web site under consideration belong to one of the concepts, either an existing concept from the source Web site or a new concept. In essence, the joint

probability in Equation 3.1 represents the probability for generating the text fragment  $(C, L, V)$  which belongs to the concept  $A$ . Suppose there are  $N$  text fragments in the Web page and let  $\Lambda$  be this set of  $N$  text fragments. The joint probability for generating  $\Lambda$  is as follows:

$$P(\Lambda; \alpha, O) = \prod_{i=1}^N P(C_i, L_i, V_i, A_i; \alpha, O) \quad (4.3)$$

As the concept of the text fragment is uncertain in the new unseen site, Equation 4.3 can be rewritten as follows:

$$P(\Lambda; \alpha, O) = \prod_{i=1}^N \sum_j P(C_i, L_i, V_i, A_{ij}; \alpha, O) \quad (4.4)$$

where  $A_{ij}$  represents the  $i$ -th text fragment belonging to  $j$ -th concept. Combining Equations 3.1 and 4.4, we can derive the log likelihood function  $L(\alpha, O; \Lambda)$  as follows:

$$L(\alpha, O; \Lambda) = \sum_{i=1}^N \log \sum_j P(C|A)P(L|A; O)P(V|A; O)P(A|\alpha) \quad (4.5)$$

According to the principle of maximum likelihood estimation of parameters, our goal is to find the set of parameters  $(\alpha, O)$  which maximizes the function depicted in Equation 4.5. However, the summation inside the logarithm makes such inference intractable. As  $A_{ij}$  is an unobservable variable in Equation 4.5, we can derive the following expected log likelihood function:

$$L'(\alpha, O; \Lambda) = \sum_{i=1}^N \sum_j P(A|\alpha) \log P(C|A)P(L|A; O)P(V|A; O) \quad (4.6)$$

which is tractable. By Jensen's inequality and the concavity of the logarithmic function, it can be proved that  $L(\alpha, O; \Lambda)$  is bounded below by  $L'(\alpha, O; \Lambda)$  [10]. Therefore, maximizing  $L'(\alpha, O; \Lambda)$  is equivalent to maximizing  $L(\alpha, O; \Lambda)$ . The EM algorithm [4] is employed to iteratively increase  $L'(\alpha, O; \Lambda)$  until convergence is reached. The E-Step and M-Step are as follows:

E-Step:

$$P(A|C, L, V; \alpha_t, O_t) \propto P(C|A)P(L|A; O_t)P(V|A; O_t)P(A|\alpha_t)$$

M-Step:

$$(\alpha_{t+1}, O_{t+1}) = \arg \max_{\alpha, O} L'(\alpha, O; \Lambda, \alpha_t, O_t)$$

We can express the terms  $P(C|A)$  with probabilities related to the set of features related  $C$ . Assume that these features are independent to each other,  $P(C|A)$

<sup>3</sup>The details of the document object model can be found in <http://www.w3.org/DOM/>.

can be expanded as follows:

$$P(C|A) = \prod_{k=1}^{Z_c} P(f_k^c(C)|A) \quad (4.7)$$

where  $Z_c$  is the number of content features and  $f_k^c(C)$  is the  $k$ -th feature of the content  $C$ . We design the following features to characterize the content of a concept.

- $f_1^c$ : the number of characters in the content
- $f_2^c$ : the number of tokens in the content
- $f_3^c$ : the average number of characters per token in the content
- $f_4^c$ : the number of digits in the content
- $f_5^c$ : the number of floating points in the content
- $f_6^c$ : the number of alphabets in the content
- $f_7^c$ : the number of upper case alphabets in the content
- $f_8^c$ : the number of lower case alphabets in the content
- $f_9^c$ : the number of punctuations in the content
- $f_{10}^c$ : the number of HTML tags in the content
- $f_{11}^c$ : the number of tokens starting with capital letter in the content

We assume that each of the above features is normally distributed with mean ( $\mu_k^c$ ) and variance ( $\sigma_k^c$ ) for the  $k$ -th content feature. In the EM framework, they are calculated as follows:

$$\mu_k^c = \frac{\sum_i P(A|C, L, V; \alpha_t, O_t) f_k^c(C)}{\sum_i P(A|C, L, V; \alpha_t, O_t)} \quad (4.8)$$

$$\sigma_k^c = \frac{\sum_i P(A|C, L, V; \alpha_t, O_t) (f_k^c(C) - \mu_k^c)^2}{\sum_i P(A|C, L, V; \alpha_t, O_t)} \quad (4.9)$$

Similarly, we can expand the terms  $P(L|A; O)$  and  $P(V|A; O)$ . However, since the text fragments under consideration do not come with the ontology of the new unseen Web site,  $P(L|A; O)$  and  $P(V|A; O)$  cannot be found directly. We propose a metric ( $\lambda$ ) to estimate  $P(L|A; O)$  and  $P(V|A; O)$  from  $P(L|A)$  and  $P(V|A)$ . The idea is to model the observation that concepts with a common parent are generally located in a nearby position and their header labels are related. Therefore,  $P(L|A; O)$  and  $P(V|A; O)$  are expanded as follows:

$$P(L|A; O) = \lambda_l(A, L, O) \prod_{k=1}^{Z_l} P(f_k^l(L)|A) \quad (4.10)$$

$$P(V|A; O) = \lambda_v(A, V, O) \prod_{k=1}^{Z_v} P(f_k^v(V)|A) \quad (4.11)$$

where  $Z_l$  and  $Z_v$  are the number of features for header label and visual layout respectively.  $f_k^l(L)$  and  $f_k^v(V)$  are the  $k$ -th header label feature and the  $k$ -th visual layout feature respectively.  $\lambda$  will be discussed in the later part of this section. Similarly we derive several features related to header labels and the visual layout. Unlike the content features, some of the features derived for header labels and the visual layout such as the color, boldness, and the occurrence of a certain word are discrete. Instead of calculating the mean and variance,

we can compute the  $P(f_k^l(L)|A; O)$ ,  $P(f_k^v(V)|A; O)$  for the discrete features as follows:

$$P(f_k^l(L) = \gamma|A) = \frac{1 + \sum_{i=1}^N \#(f_k^l(L) = \gamma) P(A|C, L, V; \alpha_t, O_t)}{|f_k^l| + \sum_{s \in f_k^l} \sum_{i=1}^N \#(f_k^l(L) = s) P(A|C, L, V; \alpha_t, O_t)} \quad (4.12)$$

$$P(f_k^v(V) = \omega|A) = \frac{1 + \sum_{i=1}^N \#(f_k^v(V) = \omega) P(A|C, L, V; \alpha_t, O_t)}{|f_k^v| + \sum_{s \in f_k^v} \sum_{i=1}^N \#(f_k^v(V) = s) P(A|C, L, V; \alpha_t, O_t)} \quad (4.13)$$

where  $|f_k^l|$  and  $|f_k^v|$  are the number of possible values of the  $k$ -th header label feature and the  $k$ -th visual layout feature respectively,  $\#(f_k^l(L) = \gamma)$  and  $\#(f_k^v(V) = \omega)$  are the count of the number of times  $f_k^l(L)$  equals to  $\gamma$  and the count of the number of times  $f_k^v(V)$  equals to  $\omega$  respectively.

The term  $P(A; \alpha)$  in Equation 3.2 can be estimated by:

$$P(A = \psi; \alpha_t, O_t) = \frac{1 + \sum_{i=1}^N P(A = \psi|C, L, V; \alpha_t, O_t)}{|A| + N} \quad (4.14)$$

where  $|A|$  is the number of possible values of  $A$ .

Recall that concepts with common parents are likely to have similar tokens in their header labels. The term  $\lambda_l(A, L, O)$  in Equation 4.10 is proposed to model this observation.  $\lambda_l(A, L, O)$  is defined as follows:

$$\lambda_l(A, L, O) = \frac{1}{L} \max_H \lambda(\tau^{\delta_O(A, H)}, D'(L, L_H)) \quad (4.15)$$

where  $L$  is a normalizing factor;  $H$  is a concept in the ontology  $O$ ;  $\tau$  is a factor between 0 and 1;  $L_H$  is the header label for concept  $H$ ;  $\delta_O(A, H)$  is the distance between the concept  $A$  and  $H$  in the ontology  $O$ ; and  $D'(L, L_H)$  is the modified edit distance defined in Section 5.1;  $\lambda(x, y)$  is a function whose image is from 0 to 1 if  $x$  and  $y$  are between 0 and 1. The definition of  $\delta_O(A, H)$  is the minimum depth among  $A$  and  $H$  in the subtree rooted from their common parent in  $O$ . For example, in the ontology depicted in Figure 3,  $\delta_O(\text{"you save"}, \text{"our price"})$  is 1, while  $\delta_O(\text{"you save"}, \text{"ISBN"})$  is 2.  $\lambda(x, y)$  is a function defined as follows:

$$\lambda(x, y) = \frac{2xy}{x^2 + y^2} \quad (4.16)$$

The property of this function is that if the values of  $x$  and  $y$  are more similar, the value of the function is higher. In essence,  $\lambda$  returns a large value if the two concepts under consideration are close in the ontology and their labels are similar. As  $\lambda_l(A, L, O)$  is interpreted as probability, a normalizing factor  $L$  is involved in Equation 4.15.  $\lambda_v(A, V, O)$  in Equation 4.11 is to model the observation that concepts with common parents are likely to be located in a nearby position. The definition of  $\lambda_v(A, V, O)$  is similar to the definition of  $\lambda_l(A, L, O)$  shown in Equation 4.15, but replacing the them  $D'(L, L_H)$  with the inverse of the distance between the two concepts within the Web page.

## 5 Adapting Ontology to the New Unseen Site

Our ontology refinement framework mainly consists of three components, namely, existing concepts discovery

component, new concepts discovery component and ontology refinement component. These components are designed based on the EM algorithm described above.

**5.1 Discovering Existing Concepts** The first component of our framework is to discover existing concepts in the new unseen Web site. This is achieved by calculating  $P(A|C, L, V; \alpha, O)$  through our EM algorithm. To initialize the EM algorithm,  $P(A|C, L, V; \alpha_0, O_0)$  needs to be computed. Recall that the contents of a concept share similar characteristics in both of the source Web site and the new unseen site. Therefore, we can make use of the content of the existing concepts collected in the source Web sites to obtain the initial estimation of  $P(A|C, L, V; \alpha_0, O_0)$  for the same concept in the new unseen site. We design a modified edit distance approach to determine the initial estimation denoted by  $P_0(A|C, L, V; \alpha_0, O_0)$ . The details of the modified edit distance algorithm can be found in our previous work [19]. Suppose  $D(C, l_i^j)$  is the distance between the content  $C$  and the  $i$ -th entry of the content for the  $j$ -th concept collected in the source Web site.  $P_0(A|C, L, V; \alpha_0, O_0)$  is then estimated by:

$$P_0(A|C, L, V; \alpha_0, O_0) \sim \frac{1}{K} (\max_i \{D'(C, l_i^j)\}) \quad (5.17)$$

where  $D'(C, l_i^j) = 1 - D(C, l_i^j)$  and  $K$  is a normalization factor. With this initial estimation, the EM algorithm can iteratively estimate the parameter  $\alpha$  until convergence. To discover the existing concepts in the new unseen site, we fix  $O$  as the existing ontology from the source Web site. After obtaining the parameters,  $P(A|C, L, V; \alpha, O)$  can be calculated according to Equation 3.2. For each of the existing concepts, those informative text fragments with probability of belonging to this concept higher than a threshold will be classified as the corresponding concept in the new unseen site. It is possible that no text fragments in the new unseen site belong to a particular existing concept in the source Web site. This kind of concept will then be removed from the existing ontology.

**5.2 Discovering New Concepts** As described in the previous section, those informative text fragments which are not classified as any existing concepts will be considered for some new concepts in the unseen site. We attempt to label these new concepts by discovering their header labels from the Web page. We make use of the classified text fragments and their header labels to accomplish this task. The idea is to train a binary classifier to predict if a certain text fragment in the new unseen site is a header label of the new concept.

The header label of a concept is one of the surrounding texts of the text fragments regarding the content of the concept. We can construct a set of training examples by pairing the classified text fragments and each of their surrounding texts. If the surrounding text is

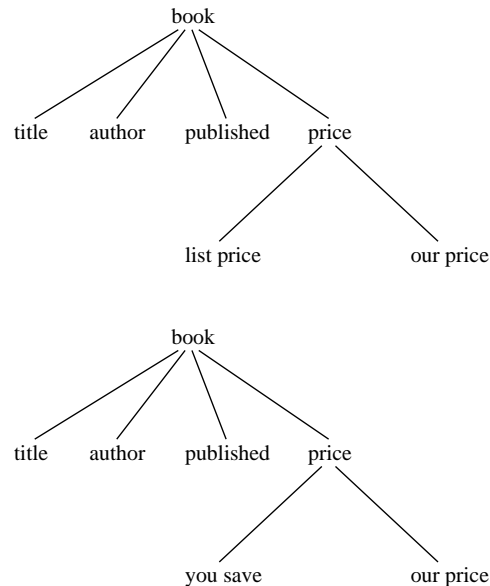


Figure 6: Two samples of potential ontology candidates for the Web site shown in Figure 1.

the header label, this pair becomes the positive training example, otherwise it becomes the negative training example. Next, we can train a binary classifier using this set of training examples. This classifier can predict whether the surrounding text is the header label of a concept, given a text fragment regarding the content of a concept and its surrounding text. Several features are considered in the learning of the classifier. For example, the relative distance between the text fragments and their surrounding text, the characteristics of the surrounding text such as the color, boldness, capitalization are also considered. We employ a naive Bayesian approach and maximum likelihood estimation, in a fashion similar the approach described in Section 4, for this classification task.

Similarly, we can pair the unclassified text fragments generated from Section 5.1 with each of their surrounding texts. Each pair is then classified by the learned classifier to predict the probability that the surrounding text is the header label. The new concepts are then labeled with the discovered header labels. The newly discovered concepts will then be added to the ontology according to the method described in Section 5.3.

**5.3 Refining Ontology** The newly discovered concepts will be added to the ontology. The location at which the new concept should be placed in the ontology is determined by considering the semantic meaning and the orthographic characteristics of the header labels. We construct a set of potential ontology candidates which form the search space of the parameter  $O$  in our framework described in Section 4. Figure 6 shows

two samples of potential ontology candidate for the Web site shown in Figure 1. The ontology of the Web site is then inferred iteratively in the EM algorithm.

We construct the potential ontology candidate using two different methods. One method is to consider the orthographic characteristics of the header labels. As mentioned before, similar concepts are likely to share some similarity in their header labels. For example, “list price” and “our price” are the header labels of two related concepts. We compare the header labels of the concepts. If they have common suffix or common prefix, they will be considered as the subconcepts of a concept labeled with their common suffix or common prefix. Another evidence is to consider semantic meaning of the header labels. For example, the phrase “list price” and the word “save” are semantically related to the concept “price”. They are likely to be the subconcepts of price. To find the semantic relationship between the header labels, we make use of Lexical FreeNet which is a query system for finding the semantic relationship between two phrases<sup>4</sup> [2]. For example, If querying Lexical FreeNet with the phrases “list price” and “save”, it returns the fact that “price” is the generalization of the phrases “list price” and “save”. Consequently this information is utilized to increase the likelihood that “list price” and “save” are the subconcepts of the concept “price”.

A number of potential ontology candidates are constructed using the two methods described. The idea of ontology refinement is to conduct the EM algorithm described in Section 4 with the following changes in the M-Step:

$$\alpha_{t+1} = \arg \max_{\alpha} L'(\alpha, O; \Lambda, \alpha_t, O_t)$$

$$O_{t+1} = \arg \max_O L'(\alpha, O; \Lambda, \alpha_{t+1}, O_t)$$

After convergence, the parameter  $O$  is the potential ontology candidate that maximizes the expected log likelihood function in Equation 4.6. Therefore, this ontology is the most probable ontology that generates the text fragments regarding the concepts in the new unseen site.

## 6 Identifying Informative Text Fragments

A Web page is made of text fragments. Some text fragments are concerned with the layout format such as HTML tags. Some text fragments are related to the contents of the concepts in an ontology. We call these informative text fragments. We develop a method which can precisely identify the informative text fragments in the new unseen site. The text fragments identified are

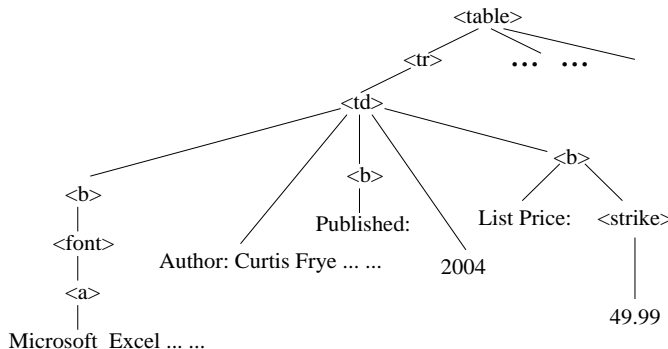


Figure 7: Part of the DOM structure representation for the Web page shown in Figure 1.

then utilized to adapt the ontology to the new site as described in Sections 4 and 5.

The idea of our method is to analyze the Document Object Model (DOM) structure of a Web page. A Web page can be represented by a DOM structure which is essentially an ordered tree consisting of two types of nodes. The first type of node is called element node which is used to represent HTML tag information. These nodes are labeled with the element name such as “<table>”, “<a>”, etc. The other type of node is called text node which includes the text displayed in the browser and labeled simply with the corresponding text. Figure 7 shows part of the DOM structure representation for the Web page shown in Figure 1.

We develop an algorithm that can effectively locate the informative text nodes in the DOM structure. The text fragments within the informative text nodes correspond to the informative text fragments. For each of the text nodes in the DOM structure, we define the path as the string created by concatenating the node labels from the first ancestor to the  $n$ -th ancestor where  $n$  is a pre-defined value. For example, as shown in Figure 7, the path for the text nodes labeled with “Published:” and “List Price:” are both equal to “<table> <tr> <td> <b>” and the path for the text nodes labeled with “Microsoft Excel 2003 Programming inside Out” is “<td> <b> <font> <a>” when  $n$  is set to 4. Note that each path may locate more than one text node in the DOM structure. We define the probability that the term  $w_i$  occurs in the text nodes located by the path  $p$  as:

$$P(w_i, p) = \frac{\Gamma(w_i, p)}{\sum_j \Gamma(w_j, p)}$$

where  $\Gamma(w_i, p)$  is the number of occurrence of  $w_i$  in all the text nodes located by  $p$ . Next, we define the *path entropy*,  $E(p)$ , as follows:

$$E(p) = - \sum_i P(w_i, p) \log P(w_i, p) \quad (6.18)$$

Note that  $E(p)$  can be calculated from more than one DOM structure by treating all the DOM structures as a forest and each  $P(w_i, p)$  is calculated by considering all the text nodes located by  $p$  in the forest. The idea of our algorithm is that the text fragments regarding the con-

<sup>4</sup>The URL of Lexical FreeNet is <http://www.lexfn.com/>.

	Web site (URL)
S1	Half Price Computer Books ( <a href="http://www.halfpricecomputerbooks.com">http://www.halfpricecomputerbooks.com</a> )
S2	Discount-PCBooks.com ( <a href="http://www.discount-pcbooks.com">http://www.discount-pcbooks.com</a> )
S3	mmistore.com ( <a href="http://www.mmistore.com">http://www.mmistore.com</a> )
S4	Amazon.com ( <a href="http://www.amazon.com">http://www.amazon.com</a> )
S5	1Bookstreet.com ( <a href="http://www.1bookstreet.com">http://www.1bookstreet.com</a> )
S6	Barnes & Noble.com ( <a href="http://www.barnesandnoble.com">http://www.barnesandnoble.com</a> )
S7	bookpool.com ( <a href="http://www.bookpool.com">http://www.bookpool.com</a> )
S8	half.com ( <a href="http://half.ebay.com">http://half.ebay.com</a> )
S9	DigitalGuru Technical Bookshops ( <a href="http://www.digitalguru.com">http://www.digitalguru.com</a> )
S10	Canon USA Consumer Product ( <a href="http://consumer.usa.canon.com">http://consumer.usa.canon.com</a> )
S11	Kodak ( <a href="http://www.kodak.com">http://www.kodak.com</a> )
S12	Panasonic USA ( <a href="http://www.panasonic.com">http://www.panasonic.com</a> )
S13	Olympus America Inc. ( <a href="http://www.olympusamerica.com/">http://www.olympusamerica.com/</a> )
S14	Konica Minolta Photo Imaging USA, Inc. ( <a href="http://www.konica.com">http://www.konica.com</a> )

Table 1: Web sites collected for experiments. S1 - S9 are collected from the book catalog domain. S10 - S14 are collected from the digital camera specification domain.

tent of the concepts are generally different in different Web pages. Therefore, we compute the entropy, which represents the randomness of the term distribution, of a node. If the term distribution in the text nodes is more random, it is likely that this text node contains text fragments regarding the content of the concepts. The details of our algorithm can be found in [18].

## 7 Case Study

Consider the Web site shown in Figure 1 which is associated with the ontology depicted in Figure 2. In this Web site, we collected a set of informative text fragments regarding the content of the concepts using an automatic information extraction method. For example, we collected the text fragments “Game Programming Gems 2”, “Microsoft Excel 2003 Programming inside Out”, and “Practical C++ Programming, 2nd Edition” for the concept “title”, and the text fragments “Mark Deloura”, “Curtis Frye”, and “Steve Oualline” for the concept “author”.

Although the Web sites in Figures 1 and 3 belong to the same domain, the existing ontology in Figure 2 is not suitable for the Web site in Figure 3. We apply our ontology refinement framework to adapt

the existing ontology from the Web site in Figure 1 to the new Web site in Figure 3. Based on the text fragments regarding the content of the concepts from the source Web site, our framework discovers that the text fragments “Oracle Advance: PL/SQL Programming with CD-ROM”, “Palm OS Programming from the Ground Up”, etc. are the book title names, the text fragments “Scott Urman”, “Robert Mykland”, etc. are the author names in new site. Similarly, it can be discovered that the concepts “you save”, and “our price” are also contained in the new site. There are in total four existing concepts involved in both Web sites and our framework can precisely discover them in the new unseen site.

The discovered existing concepts are retained in the ontology while those concepts such as “published” that cannot be found in the new site are removed. Some of the existing concepts discovered have different header labels in the new unseen site. For example, the concept “author” is associated with the header label “Author”. The concept “you save” is associated with “You Save” in Figure 3. Based on the characteristics of the header labels and the text fragments regarding the content of the concepts, our framework discovers that “ISBN” and “MSRP” are the new concepts in the new site. There are in total two new concepts in Figure 3 and our framework can correctly identify them.

The newly discovered concepts are added to the ontology by the method described in Section 5.3 to form a set of potential ontology candidates. Our framework then iteratively refines the ontology and obtain the one shown in Figure 4. This refined ontology is the same as the manually constructed ontology for the Web site in Figure 3. Therefore starting from the existing ontology in the source Web site, our framework is able to automatically refine the ontology and adapt to a new unseen Web site.

## 8 Experimental Results

We conducted extensive experiments on several real-world Web sites in two domains, namely book catalog domain and digital camera specification domain, to demonstrate the performance of our ontology refinement 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. S1 - S9 are Web sites collected from the book catalog domain. S10 - S14 are Web sites collected from the digital camera specification domain.

To evaluate the performance of our ontology refinement framework, we first manually construct the ontology for each Web site. This manually constructed

	S1		S2		S3		S4		S5		S6		S7		S8		S9		
	P	r	P	r	P	r	P	r	P	r	P	r	P	r	P	r	P	r	
S1	-	-	1.00	1.00	0.83	1.00	1.00	1.00	1.00	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	1.00	1.00
S2	1.00	1.00	-	-	0.80	1.00	0.67	1.00	0.67	0.67	0.75	0.75	1.00	1.00	0.80	1.00	1.00	0.75	0.75
S3	0.67	0.80	1.00	1.00	-	-	0.57	1.00	0.83	0.83	1.00	0.67	1.00	0.86	0.75	0.60	1.00	0.80	0.80
S4	1.00	1.00	1.00	1.00	0.75	0.75	-	-	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S5	0.83	1.00	0.83	0.83	0.67	0.67	0.60	0.75	-	-	1.00	0.80	1.00	0.83	1.00	0.60	1.00	0.75	0.75
S6	0.67	1.00	0.71	1.00	1.00	0.71	0.67	1.00	0.83	0.83	-	-	0.83	0.71	0.83	0.83	0.83	0.83	0.83
S7	1.00	1.00	1.00	1.00	0.57	0.57	0.60	0.75	0.50	0.50	0.57	0.57	-	-	0.67	0.57	1.00	0.71	0.71
S8	1.00	1.00	1.00	1.00	0.80	0.67	0.80	1.00	0.67	0.80	0.75	0.50	0.75	0.50	-	-	1.00	0.60	0.60
S9	0.83	1.00	1.00	1.00	0.60	0.50	0.75	0.75	0.50	0.50	0.50	0.33	0.40	0.33	0.80	0.67	-	-	-

Table 2: Performance of our ontology refinement framework on discovering existing concepts in the new unseen site in the book catalog domain. P and r refer to precision and recall for discovering the existing concepts in the new unseen site respectively.

ontology is considered as the gold standard for evaluation. Next our ontology refinement method was conducted to adapt the ontology from a source Web site to each of the other sites. For example, the ontology from S1 is adapted to S2 to S9 using our ontology refinement method. Each of the resulting ontologies is then compared with the corresponding manually constructed ontology for evaluation. Three aspects are examined in evaluating the performance. The first aspect is to evaluate the capability of discovering the existing concepts in the new unseen site. This is achieved by calculating the recall and precision of the discovery of existing concepts. Recall is calculated by dividing the number of existing concepts for which the system correctly discovered by the total number of actual existing concepts. Precision is calculated by dividing the number of existing concepts for which the system correctly discovered by the total number of existing concepts the system discovered. The second aspect is to evaluate the capability of discovering the new concepts in the new unseen site. This is achieved by calculating the recall and precision of the discovery of new concepts. The third aspect is to evaluate the capability of refining the ontology structure. This is achieved by calculating the tree edit distance between the adapted 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  $n$ . The second operation is to delete a node  $n$ , and make its children become the children of the original parent of  $n$ . The third operation is to insert a node  $n$  as the child of another node  $m$ , and make any child become the child of  $n$ . We fix the costs of all these edit operations to 1. The smaller the tree edit distance between the two ontologies, the higher their similarity. Readers can refer to [12] for the details of the tree edit distance.

### 8.1 Evaluation on the Book Catalog Domain

Table 2 shows the results of discovering existing concepts in the new unseen site. The first column shows the Web sites (source sites) from which the ontologies are

given. The first row shows the Web sites (new unseen sites) to which the ontologies are adapted. For example, the row labeled with S1 refers to the set of eight runs where the ontology from S1 is refined to adapt to S2 - S9. Each cell in Table 2 is divided into two sub-columns representing the precision and recall of discovering existing concepts respectively. Our method achieves a very satisfactory performance. In most of the runs, the precision and recall are above 80%. This shows that our framework can effectively discover the existing concepts in the new unseen site.

Table 3 shows the results of discovering new concepts in the new unseen site by adapting the ontology from the source Web site using our framework. The format of the table is similar to that of Table 2. Each cell in Table 3 is divided into two sub-columns representing the precision and recall of discovering new concepts respectively. Some of the cells have a value of “N/A” because there is no new concept in the unseen site. The results show that our framework can discover new concepts in most of the new sites. However, some runs such as the one discovering new concepts in S8 using the ontology from S2 have less satisfactory results. The reason is that some of the concepts are not associated with header labels and hence the new concept cannot be labeled. Nevertheless, our framework can still identify the text fragments regarding the content of new concepts and users can manually interpret the meaning of the new concepts. However, we consider that it fails to discover these new concepts in our experiments.

Table 4 shows the results of comparing the refined ontology with the manually constructed ontology in the new unseen site. The format of the table is similar to that of Table 2. Each cell in Table 4 is divided into two sub-columns representing the edit distance between the adapted ontology and the manually constructed ontologies ( $\epsilon$ ) and the edit distance between the adapted ontology and the manually constructed ontology, normalized by the total number of concepts ( $\epsilon'$ ). Note that the smaller the distance, the better is the performance. The results indicate that our framework achieves a very satisfactory result in refining the structure of the ontology. In most cases, the errors are mainly due to the

	S1		S2		S3		S4		S5		S6		S7		S8		S9	
	p	r	p	r	p	r	p	r	p	r	p	r	p	r	p	r	p	r
S1	-	-	1.00	1.00	0.67	0.67	N/A	N/A	1.00	0.50	1.00	1.00	1.00	0.67	0.80	0.80	1.00	1.00
S2	1.00	1.00	-	-	0.67	0.67	N/A	N/A	0.00	0.00	1.00	0.67	1.00	0.50	0.00	0.00	1.00	1.00
S3	1.00	1.00	1.00	1.00	-	-	N/A	N/A	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S4	1.00	1.00	1.00	1.00	0.80	0.75	-	-	1.00	0.50	1.00	0.67	1.00	0.50	0.00	0.00	1.00	0.67
S5	1.00	1.00	N/A	N/A	1.00	0.50	N/A	N/A	-	-	1.00	0.50	1.00	0.50	0.00	0.00	1.00	0.67
S6	1.00	1.00	0.50	1.00	1.00	1.00	N/A	N/A	1.00	1.00	-	-	1.00	1.00	N/A	N/A	1.00	1.00
S7	1.00	1.00	1.00	1.00	N/A	N/A	N/A	N/A	1.00	1.00	N/A	N/A	-	-	N/A	N/A	1.00	1.00
S8	1.00	1.00	0.50	1.00	1.00	0.50	N/A	N/A	1.00	0.33	0.00	0.00	1.00	0.50	-	-	1.00	1.00
S9	1.00	1.00	1.00	1.00	0.00	0.00	N/A	N/A	1.00	0.33	0.00	0.00	0.00	0.00	0.80	0.67	-	-

Table 3: Performance of our ontology refinement framework on discovering new concepts in the new unseen site in the book catalog domain. P and r refer to precision and recall for discovering the new concepts in the new unseen site respectively.

	S1		S2		S3		S4		S5		S6		S7		S8		S9	
	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$
S1	-	-	0.00	0.00	2.00	0.25	0.00	0.00	3.00	0.43	2.00	0.25	3.00	0.38	2.00	0.33	0.00	0.00
S2	0.00	0.00	-	-	2.00	0.25	0.00	0.00	4.00	0.57	3.00	0.38	2.00	0.25	2.00	0.33	1.00	0.14
S3	0.00	0.00	0.00	0.00	-	-	0.00	0.00	2.00	0.29	3.00	0.38	2.00	0.25	2.00	0.33	1.00	0.14
S4	0.00	0.00	0.00	0.00	2.00	0.25	-	-	3.00	0.43	3.00	0.38	3.00	0.38	2.00	0.33	1.00	0.14
S5	0.00	0.00	0.00	0.00	3.00	0.38	1.00	0.20	-	-	3.00	0.38	3.00	0.38	3.00	0.50	2.00	0.29
S6	0.00	0.00	0.00	0.00	4.00	0.50	0.00	0.00	2.00	0.29	-	-	3.00	0.38	1.00	0.17	1.00	0.14
S7	0.00	0.00	0.00	0.00	3.00	0.38	1.00	0.20	4.00	0.57	4.00	0.50	-	-	2.00	0.33	2.00	0.29
S8	0.00	0.00	0.00	0.00	4.00	0.50	0.00	0.00	4.00	0.57	5.00	0.63	5.00	0.63	-	-	2.00	0.29
S9	0.00	0.00	0.00	0.00	3.00	0.38	1.00	0.20	5.00	0.71	6.00	0.75	5.00	0.63	2.00	0.33	-	-

Table 4: Performance of our ontology refinement framework on refining the structure of the ontology in the book catalog domain.  $\epsilon$  refers to the tree edit distance between the refined ontology and the manually constructed ontology in the new unseen site.  $\epsilon'$  refers to the tree edit distance between the refined ontology and the manually constructed ontology normalized by the total number of concept in the new unseen site. (Note that the smaller the distance, the better is the performance.)

	S10		S11		S12		S13		S14	
	p	r	p	r	p	r	p	r	p	r
S10	-	-	1.00	0.85	1.00	1.00	1.00	1.00	1.00	1.00
S11	1.00	1.00	-	-	1.00	1.00	1.00	1.00	1.00	1.00
S12	1.00	1.00	1.00	0.93	-	-	1.00	1.00	1.00	0.83
S13	1.00	1.00	1.00	0.93	1.00	-	-	1.00	1.00	0.83
S14	1.00	1.00	1.00	0.93	1.00	1.00	1.00	1.00	-	-

	S10		S11		S12		S13		S14	
	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$	$\epsilon$	$\epsilon'$
S10	-	-	4.00	0.08	2.00	0.03	6.00	0.10	8.00	0.11
S11	0.00	0.00	-	-	4.00	0.06	8.00	0.13	15.00	0.20
S12	0.00	0.00	9.00	0.20	-	-	6.00	0.10	10.00	0.14
S13	0.00	0.00	7.00	0.16	6.00	0.09	-	-	13.00	0.18
S14	0.00	0.00	9.00	0.20	6.00	0.09	6.00	0.10	-	-

Table 5: Performance of our ontology refinement framework on discovering existing concepts in the new unseen site in the digital camera specification domain. P and r refer to precision and recall for discovering the existing concepts in the new unseen site respectively.

undiscovered concepts in the previous stage.

	S10		S11		S12		S13		S14	
	p	r	p	r	p	r	p	r	p	r
S10	-	-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.74
S11	1.00	1.00	-	-	1.00	1.00	1.00	1.00	1.00	0.64
S12	1.00	1.00	1.00	0.75	-	-	1.00	1.00	1.00	0.78
S13	1.00	1.00	1.00	0.81	1.00	-	-	1.00	1.00	0.71
S14	1.00	1.00	1.00	0.79	1.00	1.00	1.00	1.00	-	-

Table 6: Performance of our ontology refinement framework on discovering new concepts in the new unseen site in the digital camera specification domain. P and r refer to precision and recall for discovering the new concepts in the new unseen site respectively.

**8.2 Evaluation on the Digital Camera Specification Domain** We conducted experiments in the digital camera specification domain similar to the experiments described in Section 8.1. Unlike the book catalog domain, in which most of the Web pages contain more than one book records, each Web page in this domain contains only one record and each record consists of more than thirty concepts. Tables 5 and 6 show the results of discovering existing concepts and new concepts in the new unseen site respectively. The format

Table 7: Performance of our ontology refinement framework on refining the structure of the ontology in the digital camera specification domain.  $\epsilon$  refers to the tree edit distance between the refined ontology and the manually constructed ontology in the new unseen site.  $\epsilon'$  refers to the tree edit distance between the refined ontology and the manually constructed ontology normalized by the total number of concept in the new unseen site. (Note that the smaller the distance, the better is the performance.)

of the tables are the same as that of Tables 2 and 3 respectively. Our framework achieves a very good performance in discovering concepts in this domain. The main reason is that the data are organized in a more rigid table format in the Web pages and all the concepts have their header labels. Our framework can exploit such kind of regularity to discover the concepts. Table 7 shows the results of comparing the refined ontology with the manually constructed ontology in the new unseen site. The format of this table is same as that Table 4. The results indicate that our framework achieves a very promising result in ontology refinement. The header labels and the visual layout of the concepts provide very useful information to refine the ontology. For example, the header labels “LCD Monitor”, “LCD Pixels”, and “LCD Coverage” in S10 indicate the relationship of the three concepts. These three concepts are also located in three consecutive rows. Our framework can effectively refine the ontology from these clues.

## 9 Conclusions and Future Work

We have developed a probabilistic framework for refining an existing ontology from a source Web site to new unseen sites. Several clues are considered in our framework. The first clue is the text fragments regarding the content of the concepts extracted in the source Web site. The second clue is the text fragments regarding the header labels of the concepts. The third clue is the visual layout of the text fragments regarding the content and the header labels of the concepts. To cope with the uncertainty involved in these clues, we design a generative model representing the generation of text fragments regarding the concepts and the ontology corresponding to the Web page. We employ Bayesian learning techniques and expectation-maximization algorithm to achieve the goal. We have conducted extensive experiments to demonstrate the performance of our approach.

We intend to extend our framework in several directions. One possible direction is to incorporate the domain knowledge of the users. Sometimes, users have some knowledge about the ontology such as some constraints between concepts. Such domain knowledge is useful in the refinement task. Another possible direction is to integrate our refined ontology into other application such as information retrieval systems. The refined ontology may contain errors and therefore a more sophisticated approach is needed.

## References

- [1] R. Altman, M. Bada, X. Chai, M. Carillo, R. Chen, and N. Abernethy. Riboweb: An ontology-based system for collaborative molecular biology. *IEEE Transactions on Intelligent Systems*, 14(5):68–76, 1999.
- [2] D. Beeferman. Lexical discovery with an enriched semantic network. In *Proceedings of the Workshop on Applications of WordNet in Natural Language Processing Systems, ACL/COLING 1998*, 1998.
- [3] W. Cohen, M. Hurst, and L. Jensen. A flexible learning system for wrapping tables and lists in HTML documents. In *Proceedings of the Eleventh International World Wide Web Conference*, pages 232–241, 2002.
- [4] A. Dempster, A. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- [5] A. Doan, J. Madhavan, R. Dhamanker, P. Domingos, and A. Halevy. Learning to match ontologies on the semantic web. *The VLDB Journal*, 12(4):303–319, 2003.
- [6] M. Eirinaki, M. Vazigiannis, and I. Varlamis. SEWeP: Using site semantics and a taxonomy to enhance the web personalization process. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 99–107, 2003.
- [7] N. Kushmerick and B. Thomas. Adaptive information extraction: Core technologies for information agents. In *Intelligent Information Agents R&D In Europe: An AgentLink Perspective*, pages 79–103, 2002.
- [8] A. Maedche, B. Motik, and L. Stojanovic. Managing multiple and distributed ontologies on the semantic web. *The VLDB Journal*, 12(4):286–302, 2003.
- [9] A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.
- [10] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, Inc., 1997.
- [11] R. Navigli, P. Velardi, and A. Gangemi. Ontology learning and its application to automated terminology translation. *IEEE Intelligent Systems*, 18(1):22–31, 2003.
- [12] D. Shasha and K. Zhang. *Pattern Matching Algorithms*. Oxford University Press, 1997.
- [13] R. Stevens, C. Goble, I. Horrocks, and S. Bechhofer. Oiling the way to machine understandable bioinformatics resources. *IEEE Transactions on Information Technology in Biomedicine*, 6(2):129–134, 2002.
- [14] The Gene Ontology Consortium. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.
- [15] Y. Tijerino, D. Embley, D. Lonsdale, and G. Nagy. Ontology generation from tables. In *Proceedings of the Forth International Conference on Web Information Systems Engineering*, pages 242–249, 2003.
- [16] P. van der Vet and N. Mars. Bottom-up construction of ontologies. *IEEE Transaction on Knowledge and Data Engineering*, 10(4):513–525, 1998.
- [17] T. L. Wong and W. Lam. A probabilistic approach for adapting information extraction wrappers and discovering new attributes. In *Proceedings of the 2004 IEEE International Conference on Data Mining*, pages 257–264, 2004.
- [18] T. L. Wong and W. Lam. A probabilistic approach for adapting wrappers and discovering new attributes. In *The Chinese University of Hong Kong, Department of Systems Engineering and Engineering Management Technical Report*, 2004.
- [19] T. L. Wong and W. Lam. Text mining from site invariant and dependent features for information extraction knowledge adaptation. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-2004)*, pages 45–56, 2004.
- [20] World Wide Web Consortium (W3C). Semantic web. In <http://www.w3.org/2001/sw/>, 2001.