7-2009

Learning and inferencing in user ontology for personalized Semantic Web search

Xing JIANG

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Information Sciences 179 (2009) 2794–2808

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Learning and inferencing in user ontology for personalized Semantic Web search

Xing Jiang *, Ah-Hwee Tan

School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 17 July 2008
Received in revised form 17 February 2009
Accepted 5 April 2009

Keywords:
Semantic Web
User ontology
Domain ontology
Personalization
Spreading activation theory

ABSTRACT

User modeling is aimed at capturing the users' interests in a working domain, which forms the basis of providing personalized information services. In this paper, we present an ontology based user model, called user ontology, for providing personalized information service in the Semantic Web. Different from the existing approaches that only use concepts and taxonomic relations for user modeling, the proposed user ontology model utilizes concepts, taxonomic relations, and non-taxonomic relations in a given domain ontology to capture the users' interests. As a customized view of the domain ontology, a user ontology provides a richer and more precise representation of the user's interests in the target domain. Specifically, we present a set of statistical methods to learn a user ontology from a given domain ontology and a spreading activation procedure for inferencing in the user ontology. The proposed user ontology model with the spreading activation based inferencing procedure has been incorporated into a semantic search engine, called OntoSearch, to provide personalized document retrieval services. The experimental results, based on the ACM digital library and the Google Directory, support the efficacy of the user ontology approach to providing personalized information services.

1. Introduction

Domain ontologies serve as a backbone of the Semantic Web [6] by providing vocabularies and formal conceptualization of a given domain [15] to facilitate information sharing and exchange. In view that domain ontology can capture useful knowledge of a domain and the information resources, there have been many systems which utilize domain ontology in information retrieval. For example, OntoSeek [16] makes use of the concepts in the ontologies in formulating queries so as to improve the precision of the documents retrieved. Guha et al. [17] improve the quality of information retrieved by augmenting the search results with related concepts in the ontology. Hyvnen et al. [21] build a semantic portal for the Finnish museums, where a user can browse the collections with the help of the relations in the domain ontology. In addition, domain ontologies have been used in many other related applications, including video retrieval [12,45], image retrieval [13,44], information extraction [47], and knowledge management [27].

Despite the many applications of domain ontology in information retrieval, relatively few of them [8,32,34,43] are concerned with providing personalized services. Following the use of domain ontology, an obvious advantage of using ontology based models for user modeling is the support of a richer structure as well as more precise definitions of semantics. Among the prior work, Vallet et al. [43] utilize an ontology based user model with contextual information to provide personalized multimedia content access. Middleton et al. [32] exploit an ontological approach to modeling users for recommending online academic research papers. Using a similar approach, Pretschner and Gauch [34] build ontological user profiles for

* Corresponding author. Tel.: +65 9079 1750; fax: +65 6792 6559.
E-mail addresses: jiangxing@pmail.ntu.edu.sg (X. Jiang), asahtan@ntu.edu.sg (A.-H. Tan).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2009.04.005

information re-ranking and filtering. It is important to note that all of the prior work adopt a shallow approach to exploiting semantic information in the domain ontology. Specifically, they only consider the importance of concepts but not that of relations in capturing the user's interests.

A semantically rich user model and an efficient way of processing semantics are the keys to providing personalized services [11,14,31]. In view of the existing limitations, we develop an ontology based user model, called *user ontology*, which has the same level of semantics as a domain ontology. Specifically, the notion of *user ontology* signifies that the semantic relations as used in the domain ontology can also be used for user modelling. A user ontology is a specialization of domain ontology by assigning each concept and relation of the domain ontology with a specific value for indicating a user's interests. It is a personalized view of the domain conceptualization and is more comprehensive than the existing types of user models in representing a user's interests in a particular domain.

We develop a set of statistical methods for learning individual user ontologies from an existing domain ontology and adopt a spreading activation theory (SAT) [3] based procedure for inferencing in the user ontology. The proposed user ontology model and the spreading activation based inferencing procedure have been incorporated into a semantic search engine called OntoSearch [23] which originally utilizes domain ontology with keywords for document retrieval. It is a seamless extension of the OntoSearch system with the added advantages of using user ontology. We evaluate the performance of the user ontology model using two real-world document sets, namely the ACM digital library[1] and the Google Directory.[2] The experimental results support the efficacy of the proposed user ontology model and the validity of learning and exploiting user ontology.

The rest of the paper is organized as follows: Section 2 discusses the related works. Section 3 provides a formulation of the user ontology model. Section 4 describes how the spreading activation procedure can be used for inferencing in user ontology. The methods for user ontology learning are discussed in Section 5. An application of user ontology with the spreading activation procedure is provided in Section 6. The experimental results are presented in Section 7. The final section concludes and highlights future work.

2. Related works

A major challenge of providing personalized information services is to capture a user's interests in the working domain. Traditional methods, e.g., [42], rely on keywords to model user's interests and information requirements. However, due to the vagueness of keywords, the resultant user profiles cannot represent a user's interests of the target domain accurately, resulting in poor performance. In view that an ontology can support a richer semantics and offer a clear conceptual definition of the resources, people have began to develop ontology based user models to tackle this problem.

Pretschner and Gauch [34] are the pioneers to make use of ontology for modeling users and for providing personalized document access. Initially, a domain ontology is employed to organize documents, such that each document is classified as a particular concept in the domain ontology. Then, by analyzing a user's surfing history, they obtain the user's interest in each concept and record them as the user model. Finally, personalized document accesses are performed by referring to the stored user's interest factor for each concept in the user model. As the user model is structured hierarchically, it is deemed as an ontological user profile. Unfortunately, it is not known whether the other components of the ontology, such as the semantics of the relations and the structure of the ontology, could be used. Also, this user modeling method does not consider the change in the user's interests [28]. The user profile would thus be inaccurate and result in significant degradation in the system's performance over time. A similar ontological model is proposed by Vallet et al. [43] for personalized multimedia content access.

Middleton et al. [32] also present an ontology based user model for information recommendation. The user profile is built based on the user's browsing history, similar to [34]. Furthermore, a distance based inference method is performed when capturing the user's interests. When a user selects a low level concept in the concept hierarchy, it hypothesizes that the user may also be interested in a high level concept. A value is subsequently assigned to the high level concept to indicate the user's degree of interest. Finally, relevant academic papers are recommended based on the user's interest factor for each concept recorded in the user model. A major problem of this work is that the inference method is rather simple as it only considers the distance between two concepts during inferencing. Such an approach may be suitable for processing ontologies with taxonomic relations only [32]. But it has shown to be inefficient in handling complex ontologies in our prior experiments [23]. More importantly, it cannot utilize the semantics of the relations to capture a user's interests. A simpler version of this user model is adopted by Chirita et al. [8] with large taxonomies, such as the ODP taxonomy, to perform personalized document retrieval.

In summary, while many have developed ontology based user models for providing personalized services, most existing models only consider the importance of the concepts in capturing a user's interests. Although some models [8,32,43] have utilized semantic relations for user modeling, these relations are merely used to indicate that certain concepts are connected, and the semantics of the relations are not considered. To build more precise user profiles, it is critical to explore effective ways of combining semantic relations with concepts for representing a user's interests.

[1] http://portal.acm.org/dl.cfm.
[2] http://www.google.com/dirhp.

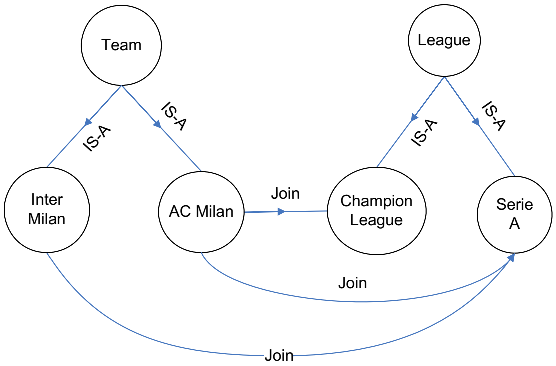

Fig. 1. A partial domain ontology for the Italian soccer teams.

3. User ontology model

A user model represents a user's interests in a particular subject domain, which forms the basis of providing personalized services. Building upon domain ontology as used in the Semantic Web, we propose that a user model for the Semantic Web could also be a type of ontology based model that captures all the semantics of individual users' interests in the domain ontology. It should be part of and can be extracted from the domain ontology. We call this model *user ontology*.

Formally, a user ontology model can be defined as a structure

$$\Theta = (\mathbf{C}, \mathbf{R}, \sigma, \theta_c, \theta_r)$$

consisting of

- two disjoint sets $\mathbf{C}$ and $\mathbf{R}$, whose elements c_x and r_{xy} are the *concepts* and *semantic relations* in the domain ontology, respectively,
- a function $\sigma : \mathbf{C} \times \mathbf{C} \to \mathbf{R}$, which associates a pair of concepts with a particular semantic relation,
- a function $\theta_c : \mathbf{C} \to [0, +\infty)$ and a function $\theta_r : \mathbf{R} \to [0, 1]$ which assign weights to concepts and relations in the domain ontology respectively.

An example to illustrate the relationship between the user ontology and the domain ontology is given as follows. Consider the sample domain ontology given in Fig. 1 that represents a partial conceptualization of the Italian soccer teams. We see that "AC Milan" and "Inter Milan" are Italian soccer teams in different leagues. But this domain ontology may be too general for individual's interests. For example, I am a big fan of the AC Milan team. Therefore, the concept "AC Milan" is more important to me than the concept "Inter Milan". Meanwhile, joining Champion League is more significant, in my opinion, than joining the Serie A League. The existing user modeling methods [8,32,34,43], however, only consider the importance of the concepts in capturing a user's interests. A user ontology, on the other hand, can capture all the necessary semantics in the domain ontology for user modeling. Specifically, each concept and relation in the domain ontology will be given a certain value for indicating user's interests. It is a personalized view of the conceptualization and is more comprehensive than the existing types of user models in representing a user's interests. The illustration of the user ontology is given in Fig. 2.

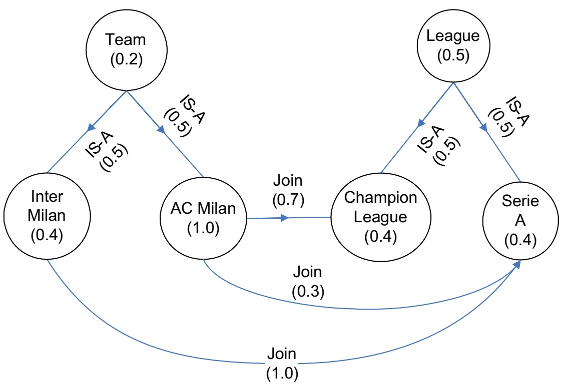

Fig. 2. An illustration of the user ontology.

For the purpose of implementation, we use a vector $\mathbf{v} = [v_1, \ldots, v_n]$, in which each element v_x stores a user's long term interest in the concept c_x, and a matrix $\mathbf{M} = [m_{xy}]$, in which each element m_{xy} records the user's long term interest in the relation r_{xy} and $\sum_y m_{xy} = 1$, to represent the functions θ_c and θ_r respectively.

4. Spreading activation theory

In the field of cognitive science, a popular representation for storing knowledge in long term memory is semantic networks [1]. In a semantic network, concepts are represented as nodes, which are linked through relations. Information processing in the semantic network typically follows the spreading activation theory, in which the activation value of each and every node spreads to its neighbouring nodes. Given a set of initial inputs to specific nodes of the network, after the spreading activation process, each and every concept in the network will be activated with certain values depending on its relations to the neighbouring nodes. As spreading activation theory has been proven to be efficient for inferencing in semantic networks, which are structurally similar to the user ontology, it is adopted as a natural choice of inferencing in the user ontology.

An illustration of the spreading activation procedure in a user ontology is given below. Referring to Fig. 3, the node "Team" is initially activated with an activation value of 1.0. Its activation then propagates across the entire semantic network following the spreading activation procedure. When the network stabilizes, all the nodes will be activated with certain activation values such as those shown in Fig. 4. Note that the activation value of each node does not depend solely on its distance from the initial node. For instance, the concept "Serie A" obtains a higher activation value than that of "Inter Milan" following the network configuration, which means "Serie A" is considered as more related to "Team".

The mechanism of the spreading activation theory is hereby defined formally below. Given a source node x and a destination node y, the activation propagation process follows the formula:

$$I_y(t_{i+1}) = O_x(t_i) \times m_{xy} \times (1 - \alpha), \quad \alpha \in [0, 1], \tag{1}$$

where $I_y(t_{i+1})$ is the input of node y at time $t_{i+1}, O_x(t_i)$ is the output of node x at time t_i, m_{xy} is the weight of the link between nodes x and y, and α is a decay factor to represent the energy loss in the spreading activation process. In a simplified

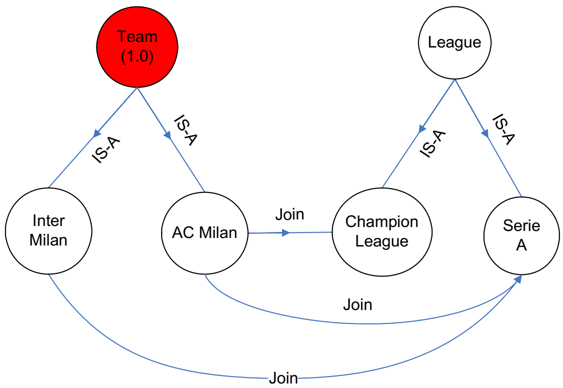

Fig. 3. Initial stage of the spreading activation process.

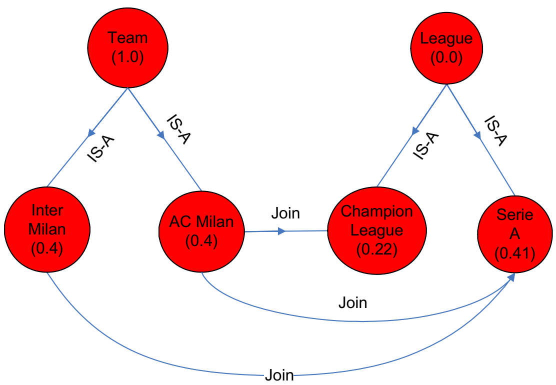

Fig. 4. Final stage of the spreading activation process.

spreading activation theory, the output of the node y at time t_i is the input of the node y at time t_i, $O_y(t_i) = I_y(t_i)$. Thus, the entire spreading activation process on the user ontology can be summarized into the following formula:

$$\mathbf{O} = [\mathscr{E} - (1-\alpha)\mathbf{M}^T]^{-1}\mathbf{I}, \tag{2}$$

where $\mathbf{I} = [I_1, \ldots, I_n]^T$ is the input to the network, $\mathbf{M}$ is the matrix representation of the user ontology whose element m_{xy} is the weight of the relation between concepts c_x and c_y, α is the decay factor, $\mathscr{E}$ is an $n \times n$ identity matrix of order n, and $\mathbf{O} = [O_1, \ldots, O_n]^T$ is the final output vector of the spreading activation process, in which O_x is the activation value of concept c_x obtained from the spreading activation process.

In this paper, the spreading activation procedure is used to obtain the user's current interests in the concepts. The results have to be combined with the user's long term interests for providing personalized services.

5. User ontology learning

After introducing the user ontology model and the associated inference algorithm, we now present a set of learning methods for assigning weights to concepts and relations in a user ontology according to a user's interests and requirements.

5.1. Learning concepts of interest

Estimating the interest factor of a user to a concept c_x is relatively straightforward. For each concept c_x in the user ontology, the degree of interest v_x is computed using an iterative formula as follows:

$$v_x(t_{i+1}) = v_x(t_i) \times \delta^{-b} + O_x, \tag{3}$$

where $v_x(t_i)$ represents the user's long term interest in concept c_x at time t_i, and O_x represents the user's current interest in concept c_x. The decay function δ^{-b} is used to prevent saturation of the interest factor c_x, where δ represents the time interval between time t_{i+1} and t_i, and $b \in [0, 1]$ is a real-value constant. This decay function is chosen over those used in the existing user models [32] for its correctness demonstrated in numerous experiments of human memory [2,4], which is structurally similar to our user ontology model.

5.2. Learning relations of interest

Learning relations of interest to a user is similar to learning concepts of interest by assigning interest factors to the relations in a user ontology. The assignment applies to both taxonomic as well as non-taxonomic relations. Initially, an estimated prior value m^*_{xy} is assigned to each element of the matrix $\mathbf{M}$. Then, a typical Bayesian solution [22] is used to compute a weighted average of the prior value and the empirical value iteratively for m_{xy} by:

$$m_{xy}(t_{i+1}) = \frac{a \times m_{xy}(t_i) + freq(r_{xy})}{a + \sum_y freq(r_{xy})}, \tag{4}$$

where $m_{xy}(t_i)$ is m_{xy}'s value at time t_i, $freq(r_{xy})$ is the frequency of the relation r_{xy} appearing in the information resources which the user is interested in, and $a \in [0, +\infty)$ is a constant to normalize the empirical value and the prior value. A small a implies that the user's long term interest in the relations is dominated by the recent observations. This solution is also consistent with the *fan effect*, an important property of the spreading activation procedure, that the activation value propagating from node x to node y decreases as the number of relations associated to node x increases [4].

5.3. An illustration

A sample scenario for user ontology learning is given as follows. Suppose Jason is a fan of AC Milan and he wishes to find some web pages that introduce AC Milan's new season in the European league and the Serie A league. Given two web pages selected by Jason, say one is annotated with three concepts "AC Milan", "European League", and "Serie A" and two relations "AC Milan join European League" and "AC Milan join Serie A", and the other is annotated with two concepts "AC Milan" and "European League" and one relation "AC Milan join European League", we may parse the two selected web pages to get the related concept and relation information. Suppose the current interest in concept "AC Milan", O_1, is 0.6. The frequency of the relation[3] "AC Milan join European League" is 2 and that of the relation "AC Milan join Serie A" is 1. We may use such information to update the user ontology. Assuming that Jason's degree of interest on the concept "AC Milan", the relation "AC Milan join European League", and the relation "AC Milan join Serie A" at time t_i are as follows: $v_1(t_i) = 0.5, m_{12}(t_i) = 0.5$, and $m_{13}(t_i) = 0.5$.

According to Eq. (3), v_1 will be updated by

$$v_1(t_{i+1}) = v_1(t_i) \times \delta^{-b} + O_1 = 0.5 \times 0.5 + 0.6 = 0.85,$$

where $\delta^{-b} = 0.5$ is the decay value to reflect the change of Jason's long term interests during t_i and t_{i+1}.

[3] In our experiments, the frequency of a relation r_{xy} for relation learning, $freq(r_{xy})$, is computed by $freq(r_{xy}) = \sum_{d_j} freq(r_{xy}, d_j)$, where d_j is a web page selected by the user and $freq(r_{xy}, d_j)$ is the frequency of the relation r_{xy} in d_j.

According to Eq. (4), m_{12} and m_{13} are computed by

$$m_{12}(t_{i+1}) = \frac{a \times m_{12}(t_i) + freq(r_{12})}{a + \sum_y freq(r_{1y})} = \frac{0.1 \times 0.5 + 2}{0.1 + 3} = 0.66,$$

$$m_{13}(t_{i+1}) = \frac{a \times m_{13}(t_i) + freq(r_{13})}{a + \sum_y freq(r_{1y})} = \frac{0.1 \times 0.5 + 1}{0.1 + 3} = 0.34,$$

where $a = 0.1$ is the normalization parameter value.

After the update, we obtain the user's degree of interest in the concepts and the relations at time t_{i+1}, given by $v_1(t_{i+1}) = 0.85, m_{12}(t_{i+1}) = 0.66$ and $m_{13}(t_{i+1}) = 0.34$. Compared with the user profile at time t_i, the new profile at time t_{i+1} indicates that Jason prefers the relation "AC Milan join European League" to the relation "AC Milan join Serie A". Documents annotated with the former relation can thus be recommended with a higher probability.

6. OntoSearch: a full-text search engine

User ontology can be used in many ways to support Semantic Web applications, including document re-ranking, information filtering, and query expansion. Document re-ranking involves the re-ordering of items returned by a search engine by moving items deemed as more relevant to a user towards the top of the list. Information filtering removes irrelevant items by referring to the user profile and delivers the relevant ones to the user. Query expansion adds related concepts to expand the user's original query for the purpose of improving the precision of the information retrieved. Here, we introduce a search engine called OntoSearch [23] that exploits user ontology model for document re-ranking.

6.1. Motivation

A key problem of providing information services in the Semantic Web is how to efficiently use these available ontologies to find the information required by individual users. To tackle this problem, many prior systems have adopted a simple solution which requires the users to include some forms of semantics explicitly in their queries [16,30,39]. For example, a user of OntoSeek [16] would need to identify the corresponding concepts of his/her query terms from the domain ontology. If he/she wants to search for information related to jaguar cars, he/she has to specify the concept "car" in the query so that the system can refer to this concept for filtering irrelevant documents.

Compared with the traditional keyword based methods, the explicit semantics in the query certainly improves the system's performance. However, this approach is rather unfriendly for typical users, since it is usually not straightforward to identify the matching concepts and relations of a query from the domain ontology. Although sophisticated interfaces [29,26] have been developed to help users select concepts and relations from the domain ontology, they ultimately do not relieve users from the burden.

In view of this problem, solutions [9,25,10,17,41] have been developed to extract related concepts directly from a submitted query. For example, the concepts *Team* and *Washington Wizards* are extracted from the query "tell me about team Wizards" for retrieving audio data about NBA team Washington Wizards [25]. However, this approach requires a robust natural language processing (NLP) tool to process the queries and a large thesaurus to map the extracted terms with the corresponding semantics in the ontology. In the above example, if the thesaurus or the domain ontology does not specify that "Washington Wizards" is the same as "Washington Bullets", the corresponding concept *Washington Wizards* could not be extracted from the query "tell me about team Bullets". In other words, the user's queries still affect the system's performance greatly. In this paper, the OntoSearch system is developed for relieving the users from specifying the semantics explicitly in the queries with a different approach.

6.2. Approach and assumption

The general principle of the OntoSearch system is to utilize the unique property of the documents on the Semantic Web, i.e., their annotations with ontological information, for inferring the semantics related to the queries submitted. Similar to traditional keywords based search engines, OntoSearch first makes use of keyword based queries to retrieve an initial set of documents. As each document retrieved has been annotated with specific semantics, by processing the associated semantics of the documents, OntoSearch derives the relevant semantics of the query submitted and uses it for document re-ranking. Specifically, the relevances of the concepts are determined implicitly in OntoSearch through a spreading activation inference in the domain ontology and further used to yield an enhanced search performance. Compared with applications that directly extract concepts from the queries, such an approach is more robust as it does not depend on thesauri or NLP tools for concept extraction, enabling users to form queries freely. A similar approach has been used to infer the most relevant concept upon a user's query in [38].

6.3. System flow

The procedure of the OntoSearch system for handling search queries is highlighted in Fig. 5. Similar to using a traditional search engine, a user submits queries consisting of keywords to the system, wherein the corresponding semantic annotation

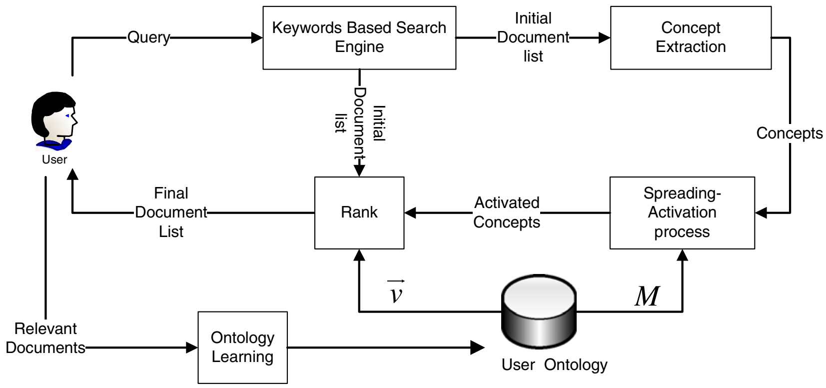

Fig. 5. The procedure of the OntoSearch system for enhanced document retrieval.

is not required. OntoSearch then returns an initial list of documents L obtained with a keyword based search method. Note that the choice of the specific keyword based search method does not greatly affect the final performance of the OntoSearch system. We have experimented with the vector space model [5] and the pagerank algorithm [33] for document retrieval. They both obtained satisfactory performance.

As the documents are pre-annotated with the ontological information, we can obtain a set of the associated concepts based on the documents in L. Using these concepts as the seeds to our user ontology, the spreading activation theory [3] process infers the concepts that are semantically related to the initial concept set. Then, the conceptual relevance scores, in terms of the concept activations in the user ontology, are combined with the user's long term interests to re-rank the documents before presentation to the user. The relevant documents are subsequently used to update the user ontology. The detailed algorithms are presented in the following sections.

6.4. Ontological indexing

We use the classical vector space model to index documents stored in the system. Specifically, a document j is represented by a vector

$$\overrightarrow{d_j} = (w^k_{1,j}, w^k_{2,j}, \ldots, w^k_{m,j}, w^c_{1,j}, w^c_{2,j}, \ldots, w^c_{n,j}), \tag{5}$$

where m is the number of index keywords in the system, n is the number of index concepts, $w^k_{i,j}$ represents the weight of the keyword k_i in document j, and $w^c_{i,j}$ represents the concept c_i's weight in document j.

For each keyword k_i, its weight $w^k_{i,j}$ is calculated using the traditional tf/idf measure [36]

$$w^k_{i,j} = freq(k_{i,j}) \times \log\frac{N}{n_i},$$

where $freq(k_{i,j})$ represents the frequency of the keyword k_i in document j, N is the number of documents in the system, and n_i is the number of documents in which keyword k_i appears.

For each concept c_i, we use a simple method to determine its weight $w^c_{i,j}$. If the concept c_i is specified in the document j, its weight $w^c_{i,j}$ is 1, else its weight is 0. This approach is different from that of pagerank-like algorithms in handling conceptual information [18].

6.5. Inferencing in user ontology

In our system, after a query is submitted, a list of documents is retrieved from the database using the keyword based search method. As documents are pre-annotated with the ontological information, we can obtain a set of the associated concepts besides the documents retrieved. The spreading activation theory is then used to infer the concepts of relevance to the user's query from the associated concept set.

Given the associated concepts together with their frequencies obtained, we form a vector

$$\mathbf{I}_q = [I_{1,q}, I_{2,q}, \ldots, I_{n,q}]^T$$

as the input to the spreading activation process. Specifically, the input to the concept c_i for a query q is calculated by

$$I_{i,q} = \frac{freq(c_i)}{\sum_{c_i} freq(c_i)}, \tag{6}$$

where $freq(c_i)$ represents the frequency of the concept c_i in the initial list.

Upon receiving the input $\mathbf{I}_q$, the spreading activation procedure is first performed on the matrix $\mathbf{M}$ to infer the concepts' current relevance to the user's query q by

$$\mathbf{O}_q = [\mathscr{E} - (1-\alpha)\mathbf{M}^T]^{-1}\mathbf{I}_q.$$

Then, for each concept c_i, the current relevance score $O_{i,q}$ is combined with the user's long term interest v_i to derive a final score $S_{i,u}$. This final score, representing a balance between the user's long time interest and the current relevance to the concepts, is used to re-rank the documents. In our application, the final score $S_{i,u}$ for concept c_i is computed by

$$S_{i,u} = O_{i,q} + v_i \times \delta^{-b}, \tag{7}$$

where δ represents the time interval since the last query, $b \in [0,1]$ is a real-valued constant, and δ^{-b} simulates the decay function occurred in the long term memory [4].

6.6. Similarity measures

Two similarity measures, namely the cosine measure and the position based measure, are used to re-rank documents in the initial list. Those documents deemed as more relevant are moved towards the top of the final list for presentation to the user.

6.6.1. Cosine measure

Similar to the way documents are indexed in OntoSearch, we can represent a query q by a vector

$$\overrightarrow{q} = (w^k_{1,q}, w^k_{2,q}, \ldots, w^k_{m,q}, s_{1,q}, s_{2,q}, \ldots, s_{n,q}), \tag{8}$$

where $s_{i,q}$ (the normalized value of $S_{i,u}$) represents the relevance of the concept c_i, and $w^k_{i,q}$ represents the keyword k_i's relevance to the query q. In OntoSearch, the value of $w^k_{i,q}$ is calculated by

$$w^k_{i,q} = \frac{freq(k_{i,q})}{\sum_{k_j \in q} freq(k_{j,q})}, \tag{9}$$

where $freq(k_{i,q})$ represents the frequency of keyword k_i in the query q.

Now, given the document vector $\overrightarrow{d_j}$ and the query vector $\overrightarrow{q}$, the similarity measure of a document j to the query q is computed as:

$$sim(j,q) = \frac{\overrightarrow{d_j} \cdot \overrightarrow{q}}{|\overrightarrow{d_j}| \times |\overrightarrow{q}|}. \tag{10}$$

This formula is a classical measure used in the vector space model. Although several variants are available, we adopt this version in OntoSearch as it outperforms the others in our prior experiments. Documents holding a higher cosine value are then moved towards the top of the final list and returned to the user.

6.6.2. Position based measure

For documents annotated with just one concept, we can use a simple but fast measure for re-ranking. Specifically, we compute a balance value B for the document j by

$$B_j = \beta \times F_j + (1-\beta) \times T_j, \tag{11}$$

where F_j is the rank of the document j in the initial list L, T_j represents the rank of the document j's associated concept among the elements of $\overrightarrow{S_u}$, and $\beta \in [0,1]$ is a constant. This similarity measure is based on a simple hypothesis that documents holding a higher ranking in L and annotated with the more activated concepts would be more relevant to the submitted query q. Documents with smaller B_j values are then moved towards the top of the final list for presentation to the user.

After browsing the final list, the user may specify those relevant documents, which can be used to update the concepts and relations of interest in the user ontology.

7. Experiments

Two real-world document collections are used in our experiments, namely the ACM digital library and the Google Directory. The details are given in the following sections.

7.1. Searching ACM digital library

7.1.1. The data set and domain ontology

The ACM digital library (http://portal.acm.org/dl.cfm) is an online database containing more than 54,000 computer science related articles from 30 journals and 900 proceedings of the Association for Computing Machinery. Compared with

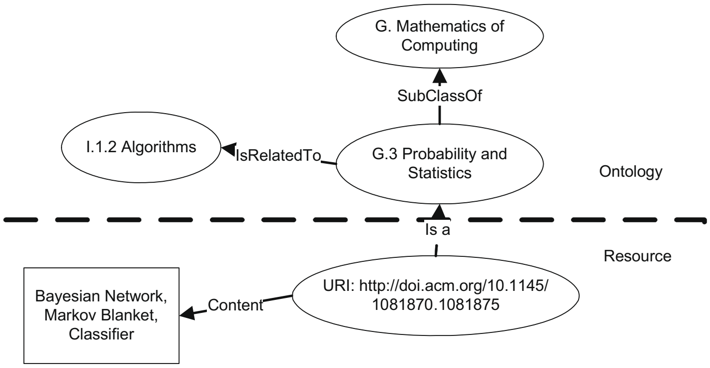

Fig. 6. An illustration of a document and its annotations in the ACM digital library data set.

ordinary document collections, one key advantage of the ACM digital library is that the publications have been annotated with terms according to the ACM Computing Classification System (CCS).[4] The CCS can be used as a simple domain ontology which provides a hierarchical structure to describe the various research fields in computer science. Documents indexed using the CCS terms are similar to web pages annotated with ontologies. An illustration of a document in the data set is given in Fig. 6. We see that this document contains the keywords, including "Bayesian network", "Markov blanket", and "classifier". Also, it is annotated with the concept "G.3 Probability and Statistics".

As the CCS ontology only contains hierarchical relations, we hypothesize that two concepts are related semantically if they are used to index the same paper so as to enrich the CCS ontology with more semantic relations. For example, the concept *F.2.2 Non-numerical Algorithms and Problems* is related to the concept *G.2.2 Graph Theory* as they are both used to index the Brinkman's paper [7] on dimensional reduction. This relation can therefore be added into the CCS ontology and further used to annotate this paper. Our approach of finding semantic relations between concepts is similar to the one for finding co-citation information in an author analysis application [19].

To facilitate experimentation, we build a local database to store the documents of the ACM digital library[5], wherein each document is indexed following the approach described in Section 6.4. Note that we have adopted an explicit, non-embedded annotation scheme to link a given domain ontology to the documents in the system. For each document, we create a separate file to store the concepts and the relations, and bind it with the original file. Although this method may fail to associate the semantic markup with the specific components in the document, it is relatively simple and easy to implement [30,39].

7.1.2. Results

A group of ten users is involved in evaluating the user ontology's ability for providing personalized services.

The profile of each user is initialized as follows. For each concept c_x of the user ontology, its value v_x equals zero. For each relation r_{xy}, its initial weight m^*_{xy} is given by

$$m^*_{xy} = \frac{freq(r_{xy})}{\sum_y freq(r_{xy})}, \tag{12}$$

where $freq(r_{xy})$ represents the frequency of the relation r_{xy} in the data set.

Each user submits two sets of queries to the system, one for training the user ontology and the other for testing. When training the model, one has to browse the top 30 documents returned and provide feedback on the documents that are relevant. This approach is simple but effective in capturing a user's interests. In general, we could also use other approaches to obtain the documents of relevance, for example, by using the surfing behaviors [20,37] or referring to the context environment [35].

After training, the learnt user ontology is incorporated into OntoSearch, accordingly called OntoSearch_U for distinguishing from the standard OntoSearch system, to provide recommendation for the test queries. For each test query, the corresponding user has to verify the documents returned and identify their relevance. To conduct a formal evaluation, we further compare OntoSearch_U with three other systems, measuring the performance gain one can obtain with the use of domain and user ontological information.

The baseline system is the Lucene search engine[6], which employs the keyword based vector space model. The second system is the standard OntoSearch system that utilizes the domain ontology and keywords for document retrieval. Both systems do not have any personalization capability.

[4] http://www.acm.org/about/class/1998
[5] We mainly download documents published after 1998, since most of the older papers are not labeled with the CCS ontology.
[6] http://lucene.apache.org/.

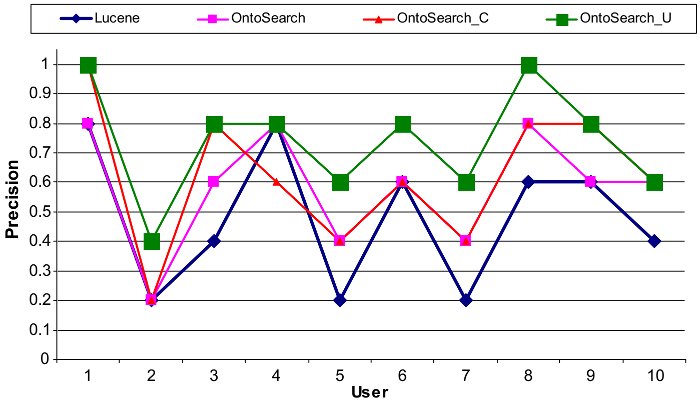

Fig. 7. The average precision of OntoSearch_U compared with Lucene, OntoSearch, and OntoSearch_C based on the top five documents retrieved.

Note that OntoSearch_U employs both concepts and relations in the user ontology model. To evaluate the benefit of concepts and relations individually, we conduct experiments on a scaled-down version of OntoSearch_U, called OntoSearch_C, that uses only concept learning in updating the user ontology models. Comparing the performance of OntoSearch_C and OntoSearch evaluates the benefit of user modeling with concepts. Benchmarking OntoSearch_U and OntoSearch_C investigates the advantage of using relations to capture the users' interests.

The performance of the four systems, namely Lucene, OntoSearch, OntoSearch_C and OntoSearch_U, in terms of the average precision for the top five (*Precision*@5) and top ten (*Precision*@10) documents retrieved, is depicted in Figs. 7 and 8 respectively. We can see OntoSearch_U consistently outperforms, or at least produces equivalent performance, compared with the other three systems in the experiments. Based on the top five documents retrieved, OntoSearch_U outperforms the other three systems in five out of the ten users. For the case of top ten documents retrieved, it outperforms in six out of the ten users. The average precision scores of the four systems and their standard deviations are summarized in Table 1. We can observe that OntoSearch_U produces the best results among the four systems, validating our approach of using user ontology to enhance search performance in the Semantic Web.

Furthermore, to verify whether the performance gain by OntoSearch_U is statistically significant, we perform paired *t*-tests on the precision scores over the ten users. The *null hypothesis* is that the performance of OntoSearch_U is equal to those

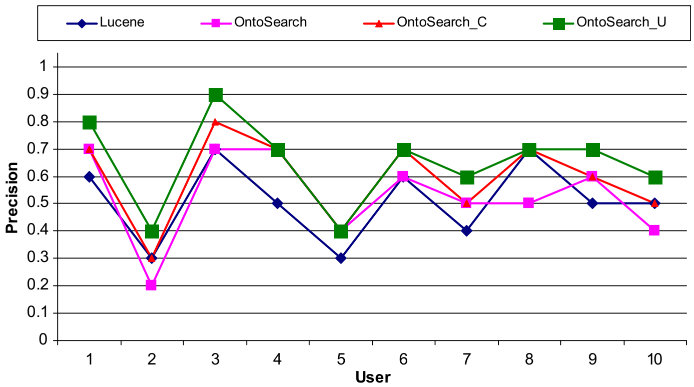

Fig. 8. The average precision of OntoSearch_U compared with Lucene, OntoSearch, and OntoSearch_C based on the top ten documents retrieved.

Table 1
The average precision and standard deviation of Lucene, OntoSearch, OntoSearch_C and OntoSearch_U on the top five and ten documents retrieved.

Search engine	Top five		Top ten	
	Avg Prec.	Std	Avg Prec.	Std
Lucene	0.48	0.23	0.51	0.14
OntoSearch	0.58	0.19	0.53	0.16
OntoSearch_C	0.62	0.23	0.59	0.16
OntoSearch_U	0.74	0.19	0.65	0.16

Table 2
The *p*-values for the paired *t*-tests on the ACM digital library data set.

Top *n*	OntoSearch_U over Lucene	OntoSearch_U over OntoSearch	OntoSearch_U over OntoSearch_C
Top five	0.0002	0.0002	**0.005**
Top ten	0.0001	0.0009	**0.005**

Level 1: Computers
Level 2: Computers/Software
Level 3: Computers/Software/Internet
…
Level 7: Computers/Software/Internet/Client/WWW/Browsers/Firefox

Fig. 9. An illustration of the ODP taxonomy's structure, where each level represents a particular category.

of Lucene, OntoSearch, and OntoSearch_C in retrieving documents. The *alternative hypothesis* is that the performance of OntoSearch_U is better than those of the other three systems. As indicated by the *p*-values for the paired *t*-tests reported in Table 2, we can see the performance of OntoSearch_U is significantly better than those of the other three systems statistically. In particular, given a Type I Error (α) of 0.05, we can reject the null hypothesis and conclude that OntoSearch_U certainly outperforms OntoSearch_C in terms of precision scores. It follows that semantic relations are useful in capturing the users' interests.

7.2. Searching Google Directory

7.2.1. The data set and domain ontology

The Google Directory (http://www.google.com/dirhp) is a service provided by Google that integrates Google's search technology with Open Directory pages (http://www.dmoz.org/) for finding information of high quality on the Web. Similar to the ACM digital library, web pages indexed in the Google Directory are labelled using the concepts of the Open Directory Project (ODP) taxonomy.[7]

The ODP taxonomy is a human constructed and maintained taxonomy that organizes the Open Directory pages into 16 main categories and numerous sub categories. An illustration of the ODP taxonomy's structure is given in Fig. 9. This diagram presents a partial structure of the computer category, where the concept *Firefox* is the lowest category in this structure.

Whereas the CCS ontology only contains taxonomic relations, the ODP taxonomy already encodes many non-taxonomic relations, for example, the *related* relation and the *symbolic* relation. We therefore do not need to use additional methods to enrich this taxonomy. But a major challenge of using the ODP taxonomy is that it is very large, containing too many concepts and relations. If we select only the top level concepts, they will be too general to capture a user's interests [8]. On the other hand, if we use the whole taxonomy, the inferencing process will be very time consuming. After initial experimentation, we select concepts from level 1 to level 3 to form the domain ontology. Altogether, there are 5965 concepts, 24,506 taxonomic relations, and 52,662 non-taxonomic relations used in our experiments.[8]

As the Google Directory contains many more documents[9], we do not store all the web pages into a local database. Instead, for each query, we extract the relevant documents with their associated concepts from the Google Directory's result page during run time. They are then re-ranked by the various search systems and returned to the users.

7.2.2. Results

A group of ten users is involved in evaluating the user ontology. The setting of this experiment is similar to the one used for the ACM digital library data set. Each user prepares two groups of queries, one for training and the other for testing. When training a user ontology, the corresponding user would browse the top 20 documents returned by his queries and provide relevance feedback to the search engine. The selected documents are then used to update the user ontology.

A problem of training user ontology in the Google Directory data set is that each document is only annotated with one concept. As a result, no explicit links between concepts can be used for relation learning. To solve this problem, we hypothesize that, for those selected relevant documents, if they are labelled with different concepts, relations will be built between these concepts. For example, when searching information about ‘‘grape wine”, documents annotated with the concepts ‘‘Recreation/Food/Drink” and ‘‘Shopping/Food/Beverage” can both be selected. A semantic relation is thus built for the two concepts and used subsequently for relation learning.

After training, the learnt user ontology is used to provide recommendation for the test queries. As in the previous experiments, we use four systems, namely Google Directory, OntoSearch, OntoSearch_C, and OntoSearch_U, to make comparisons.

[7] This taxonomy can be downloaded from http://rdf.dmoz.org/rdf/structure.rdf.u8.gz.
[8] For those documents labeled with the low level concepts, we will use the corresponding high level concepts instead.
[9] Google Directory has collected over 1.5 million URLs on November 30th, 2008. This latest data is available in http://www.google.com/dirhelp.html.

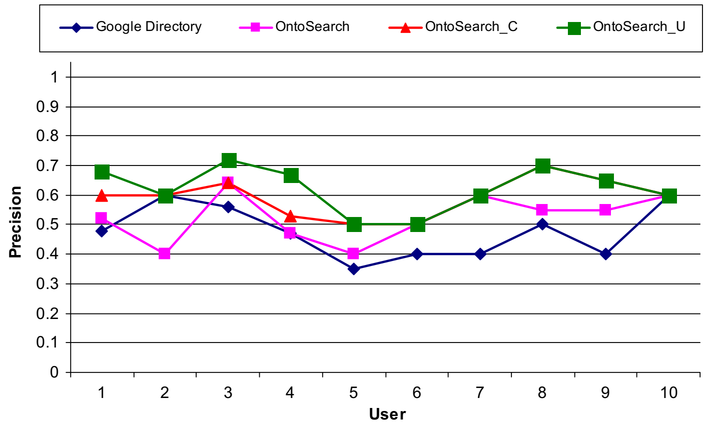

Fig. 10. The average precision of OntoSearch_U compared with Google Directory, OntoSearch, and OntoSearch_C based on the top five documents retrieved.

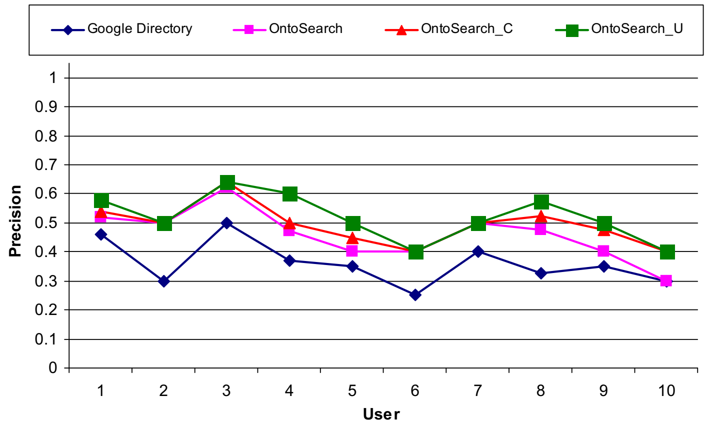

Fig. 11. The average precision of OntoSearch_U compared with Google Directory, OntoSearch, and OntoSearch_C based on the top ten documents retrieved.

The performance of the four systems for the ten users, in terms of the average precision for the top five (*Precision*@5) and top ten (*Precision*@10) documents retrieved, is depicted in Figs. 10 and 11 respectively. We observe that OntoSearch_U is the best system of all based on both evaluation criteria. Notably, based on the top ten documents retrieved, it outperforms the other three systems in five out of the ten users and produces equivalent performance to OntoSearch_C for the remaining users. The average precision scores and the standard deviations of the four systems are given in Table 3. Although the performance difference is not very significant on the Google Directory data set, OntoSearch_U still produces the best performance, demonstrating the user ontology model's efficacy in supporting personalized services.

As in the previous experiments, we conduct paired *t*-tests on the precision scores over the ten users to verify whether the performance gain by OntoSearch_U is statistically significant. The *null hypothesis* is that the performance of OntoSearch_U is not different from those of the other three systems in document retrieval. The *alternative hypothesis* is that OntoSearch_U outperforms the other three systems. The *p*-values for the paired *t*-tests are reported in Table 4. Given a Type I Error (α) of 0.05, we can reject the null hypothesis again and conclude that semantic relations are certainly useful for user modeling.

Table 3
The average precision and standard deviation of Google Directory, OntoSearch, OntoSearch_C and OntoSearch_U on the top five and ten documents retrieved.

Search engine	Top five		Top ten	
	Avg Prec.	Std	Avg Prec.	Std
Google Directory	0.48	0.09	0.36	0.08
OntoSearch	0.52	0.08	0.46	0.09
OntoSearch_C	0.59	0.07	0.49	0.07
OntoSearch_U	0.62	0.08	0.52	0.08

Table 4
The *p*-values for the paired *t*-tests on the Google Directory data set.

Top *n*	OntoSearch_U over Google Directory	OntoSearch_U over OntoSearch	OntoSearch_U over OntoSearch_C
Top five	0.0002	0.0016	**0.0478**
Top ten	0.000002	0.0022	**0.0172**

Table 5
Analysis of the performance gain brought by relation learning on the ACM digital library data set and the Google Directory data set.

Data set	*Precision*@5			*Precision*@10		
	OntoSearch_C	OntoSearch_U	Ratio	OntoSearch_C	OntoSearch_U	Ratio
ACM digital library	0.62	0.74	19%	0.59	0.65	10%
Google Directory	0.59	0.62	5%	0.49	0.52	6%

Table 6
The average precision and standard deviation of Onto_U on the top ten documents retrieved from the ACM digital library data set with different decay values for inferencing in the user ontology.

α	0.3	0.5	0.8
Avg Prec.	0.65	0.64	0.63
Std	0.16	0.17	0.21

7.3. Discussion

7.3.1. Influence of the ontology's size on the performance

In the previous sections, we have reported the performance of the OntoSearch system with user ontology on two real-world document sets. Two key observations can be drawn from the experimental results: (1) OntoSearch systems with the personalization capability (i.e., OntoSearch_U and OntoSearch_C) outperform their counterpart (i.e., Lucene, Google Directory, and standard OntoSearch) without such a capability; (2) OntoSearch with the full set of concept learning and relation learning (i.e., OntoSearch_U) outperforms its counterpart with only concept learning (i.e., OntoSearch_C). Such results support our approach of using user ontology to enhance search performance in the Semantic Web. By recording a user's degree of interest in both concepts and relations, a more precise user profile is built, which can be used subsequently to improve the system's performance for document retrieval.

However, we note that the performance gain brought by relation learning differed rather significantly on the two document collections. As reported in Table 5, we can see the precision improvements for the ACM digital library are more considerable than those for the Google Directory. This difference would be caused by the limitation of the user ontology model on large ontologies. Note that a user ontology model is supposed to record the user's degree of interest in the relations besides the concepts. For small ontologies, such as the CCS ontology used in the ACM digital library, this operation is relatively easy to perform, as there are fewer relations encoded. But large ontologies normally contain many relations. For example, the trimmed ODP taxonomy used for the Google Directory data set has a total of 77,168 relations. Given the small number of training queries, it is difficult to construct a user ontology to record a user's interests fully on all the relations. Therefore, the improvement on the Google Directory data set is not as significant in our experiments. To guarantee the effectiveness of the user ontology model for capturing users' interests, a longer training time is required for working with large ontologies.

7.3.2. Using customized parameter values for individual users

Additional experiments are conducted to investigate the possibility of improving the user ontology model's performance by using customized values for key parameters, such as the decay rate α during inferencing, for individual users. As shown in Table 6, the average precision of OntoSearch_U on the top ten documents retrieved from the ACM digital library data set does not differ significantly with the different decay values. However, for individual users (see Fig. 12), some (e.g., user 1 and 7) obtain better results with a large decay value while others (e.g., user 9 and 10) prefer a small decay value. Such results show that setting customized parameter values for individual users is a possible approach to improving the user ontology model's performance. However, adopting such an approach would also suffer from other problems. For example, the complete ODP taxonomy contains more than 590,000 concepts and 1,000,000 relations.[10] Finding an appropriate decay value for individual users on this ontology will be extremely time consuming. Consequently, the performance gain obtained may not outweigh the computational cost of finding appropriate parameter values for every user. This is an issue worthy of future investigation.

[10] The statistics are based on the snapshot of the ODP taxonomy on 2007-01-25.

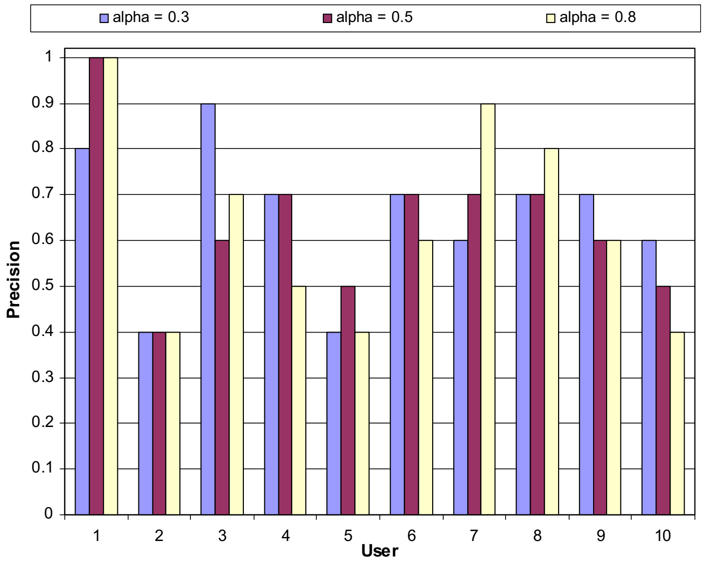

Fig. 12. The average precision of OntoSearch_U on the top ten documents retrieved from the ACM digital library data set with different decay values for inferencing in the user ontology.

8. Conclusion

This paper has presented a semantic based user model, called user ontology, for supporting personalized applications in the Semantic Web. Compared with prior work, the user ontology model has a richer structure and more precise definitions of semantics by utilizing both concepts and semantic relations in the domain ontology for representing a user's interests. We present a set of statistical methods for learning user ontology and a spreading activation procedure for inferencing in the user ontology. It is our goal in the near future to develop an integrated approach that leverages on spreading activation theory for both learning and exploiting user ontology.

The proposed user ontology model has been integrated into a semantic search engine and applied to document retrieval in the ACM digital library and the Google Directory. Although the initial results are encouraging, we look forward to the availability of more large scale date sets pre-annotated with associated ontologies, which will enable a more rigorous quantitative comparison of our approach with the state-of-the-art methods. Meanwhile, in view that relation learning requires a longer learning cycle for large ontologies, we shall explore *Collaborative filtering* [40] method for building user ontology. By referring to the profiles of the users with similar interests, a relatively precise user ontology can be constructed within a shorter time. Also, we shall investigate the possibility of incorporating the user information collected during run time, such as the time spent on a web page [46] and the user's eye movement during browsing [24], into the user ontology model, since such information has shown to be useful for capturing a user's interests.

Acknowledgements

This work was supported by the Singapore National Research Foundation Interactive Digital Media R\&D Program, under research Grant NRF2008IDM-IDM004-037. The authors would like to thank the anonymous reviewers for providing many invaluable comments to the previous versions of the paper.

References

[1] J.R. Anderson, Language, Memory and Thought, Erlbaum, Hillsdale, NJ, 1976.
[2] J.R. Anderson, The Architecture of Cognition, Harvard University Press, Cambridge, MA, 1983.
[3] J.R. Anderson, A spreading activation theory of memory, Journal of Verbal Learning and Verbal Behavior 22 (1983) 261–295.
[4] J.R. Anderson, Rules of the Mind, Erlbaum, Hillsdale, NJ, 1993.
[5] R.A. Baeza-Yates, B.A. Ribeiro-Neto, Modern Information Retrieval, ACM Press/Addison-Wesley, 1999.
[6] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities, Scientific American 285 (5) (2001) 34–43.
[7] B. Brinkman, M. Charikar, On the impossibility of dimension reduction in ℓ_1, Journal of ACM 52 (5) (2005) 766–788.
[8] P.-A. Chirita, W. Nejdl, R. Paiu, C. Kohlschütter, Using odp metadata to personalize search, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 178–185.
[9] J. Contreras, V.R. Benjamins, M. Blázquez, S. Losada, R. Salla, J. Sevilla, D. Navarro, J. Casillas, A. Mompó, D. Patón, Ó. Corcho, P. Tena, I. Martos, A semantic portal for the international affairs sector, in: Engineering Knowledge in the Age of the Semantic Web, 14th International Conference, 2004, pp. 203–215.

[10] J. Davies, R. Weeks, Quizrdf: search technology for the semantic web, in: 37th Hawaii International Conference on System Sciences, 2004, pp. 40112–40119.
[11] P. Dolog, W. Nejdl, Semantic web technologies for the adaptive web, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web, Methods and Strategies of Web Personalization, 2007, pp. 697–719.
[12] G. Erozel, N.K. Cicekli, I. Cicekli, Natural language querying for video databases, Information Science 178 (12) (2008) 2534–2552.
[13] J. Fan, Y. Gao, H. Luo, Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation, IEEE Transactions on Image Processing 17 (3) (2008) 407–426.
[14] S. Gauch, M. Speretta, A. Chandramouli, A. Micarelli, User profiles for personalized information access, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web, Methods and Strategies of Web Personalization, 2007, pp. 54–89.
[15] T.R. Gruber, A translation approach to portable ontology specification, Knowledge Acquisition 5 (1993) 199–220.
[16] N. Guarino, C. Masolo, G. Vetere, Ontoseek: content-based access to the web, IEEE Intelligent Systems 14 (3) (1999) 70–80.
[17] R.V. Guha, R. McCool, E. Miller, Semantic search, in: Proceedings of the 12th International World Wide Web Conference, 2003, pp. 700–709.
[18] L. Guo, F. Shao, C. Botev, J. Shanmugasundaram, Xrank: ranked keyword search over xml documents, in: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, 2003, pp. 16–27.
[19] Y. He, S.C. Hui, A.C.M. Fong, Citation-based retrieval for scholarly publications, IEEE Intelligent Systems 18 (2) (2003) 58–65.
[20] E.H. hsin Chi, P. Pirolli, K. Chen, J.E. Pitkow, Using information scent to model user information needs and actions and the web, in: Proceedings of the SIG-CHI on Human factors in Computing Systems, 2001, pp. 490–497.
[21] E. Hyvnen, M. Junnila, S. Kettula, E. Mkel, S. Saarela, M. Salminen, A. Syreeni, A. Valo, K. Viljanen, Publishing museum collections on the semantic web: the museumfinland portal, in: Proceedings of the 13th International Conference on World Wide Web – Alternate Track Papers & Posters, 2004, pp. 418–419.
[22] G.R. Iversen, Bayesian Statistical Inference, Sage Publications, Inc., 1984.
[23] X. Jiang, A.-H. Tan, Ontosearch: A full-text search engine for the semantic web, in: Proceedings, The 21 National Conference on Artificial Intelligence, 2006, pp. 1325–1330.
[24] T. Joachims, L.A. Granka, B. Pan, H. Hembrooke, F. Radlinski, G. Gay, Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search, ACM Trans. Inf. Syst. 25 (2) (2007) 7.
[25] L. Khan, D. McLeod, E. Hovy, Retrieval effectiveness of an ontology-based model for information selection, The VLDB Journal 13 (1) (2004) 71–85.
[26] A. Kiryakov, B. Popov, D. Ognyanoff, D. Manov, A. Kirilov, M. Goranov, Semantic annotation, indexing, and retrieval, in: International Semantic Web Conference, 2003, pp. 484–499.
[27] L.F. Lai, A knowledge engineering approach to knowledge management, Information Science 177 (19) (2007) 4072–4094.
[28] W. Lam, S. Mukhopadhyay, J. Mostafa, M.J. Palakal, Detection of shifts in user interests for personalized information filtering, in: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1996, pp. 317–325.
[29] A. Maedche, S. Staab, N. Stojanovic, R. Studer, Y. Sure, Semantic portal: the seal approach, in: Spinning the Semantic Web, 2003, pp. 317–359.
[30] J. Mayfield, T. Finin, Information retrieval on the semantic web: integrating inference and retrieval, in: Proceedings of the SIGIR Workshop on the Semantic Web, 2003.
[31] A. Micarelli, F. Gasparetti, F. Sciarrone, S. Gauch, Personalized search on the world wide web, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web, Methods and Strategies of Web Personalization, 2007, pp. 195–230.
[32] S.E. Middleton, N.R. Shadbolt, D.C.D. Roure, Ontological user profiling in recommender systems, ACM Transactions on Information Systems 22 (1) (2004) 54–88.
[33] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: bringing order to the web, Tech. rep., Stanford Digital Library Technologies Project, 1998, URL <http://citeseer.ist.psu.edu/page98pagerank.html>.
[34] A. Pretschner, S. Gauch, Ontology based personalized search, in: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, 1999, pp. 391–398.
[35] B.J. Rhodes, P. Maes, Just-in-time information retrieval agents, IBM Systems Journal 39 (3–4) (2000) 685–704.
[36] C.J.V. Rijsbergen, Information Retrieval, second ed., Butterworth, London, 1979.
[37] R.L. Roberto, S.R.P. da Silva, An approach for identification of user's intentions during the navigation in semantic websites, in: The Semantic Web: Research and Applications, 4th European Semantic Web Conference, 2007, pp. 371–383.
[38] C. Rocha, D. Schwabe, M.P. de Arag ao, A hybrid approach for searching in the semantic web, in: Proceedings of the 13th International Conference on World Wide Web, 2004, pp. 374–383.
[39] U. Shah, T. Finin, A. Joshi, Information retrieval on the semantic web, in: Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, 2002, pp. 461–468.
[40] M. Slaney, J. Subrahmonia, P.P. Maglio, Modeling multitasking users, in: User Modeling 2003, 9th International Conference, 2003, pp. 188–197.
[41] N. Stojanovic, On analysing query ambiguity for query refinement: the librarian agent approach., in: Conceptual Modeling – ER 2003, 22nd International Conference on Conceptual Modeling, 2003, pp. 490–505.
[42] A.-H. Tan, C. Teo, Learning user profiles for personalized information dissemination, in: Proceedings, International Joint Conference on Neural Networks, 1998, pp. 183–188.
[43] D. Vallet, P. Castells, M. Fernández, P. Mylonas, Y.S. Avrithis, Personalized content retrieval in context using ontological knowledge, IEEE Transactions on Circuits and Systems for Video Technology 17 (3) (2007) 336–346.
[44] H. Wang, X. Jiang, L.-T. Chia, A.-H. Tan, Ontology enhanced web image retrieval: aided by wikipedia & spreading activation theory, in: Multimedia Information Retrieval, 2008, pp. 195–201.
[45] C. Xu, Y.-F. Zhang, G. Zhu, Y. Rui, H. Lu, Q. Huang, Using webcast text for semantic event detection in broadcast sports video, IEEE Transactions on Multimedia 10 (7) (2008) 1342–1355.
[46] S. Xu, Y. Zhu, H. Jiang, F.C.M. Lau, A user-oriented webpage ranking algorithm based on user attention time, in: AAAI, 2008, pp. 1255–1260.
[47] H.-T. Zheng, B.-Y. Kang, H.-G. Kim, An ontology-based approach to learnable focused crawling, Information Science 178 (23) (2008) 4512–4522.