7-2006

# OntoSearch: A full-text search engine for the semantic web

Xing JIANG

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



## Citation

JIANG, Xing and TAN, Ah-hwee. OntoSearch: A full-text search engine for the semantic web. (2006). *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI '06): Boston, July 16-20*. 2, 1325-1330.

**Available at:** https://ink.library.smu.edu.sg/sis_research/6509

# OntoSearch: A Full-Text Search Engine for the Semantic Web

**Xing Jiang** and **Ah-Hwee Tan**
School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
E-mail: {jian0008, asahtan}@ntu.edu.sg

**Abstract**

OntoSearch, a full-text search engine that exploits ontological knowledge for document retrieval, is presented in this paper. Different from other ontology based search engines, OntoSearch does not require a user to specify the associated concepts of his/her queries. Domain ontology in OntoSearch is in the form of a semantic network. Given a keyword based query, OntoSearch infers the related concepts through a spreading activation process in the domain ontology. To provide personalized information access, we further develop algorithms to learn and exploit user ontology model based on a customized view of the domain ontology. The proposed system has been applied to the domain of searching scientific publications in the ACM Digital Library. The experimental results support the efficacy of the OntoSearch system by using domain ontology and user ontology for enhanced search performance.

## I. Introduction

Semantic Web (Berners-Lee, Hendler, & Lassila 2001) is an initiative by the World Wide Web consortium to utilize ontological information for enhanced information access. Among the numerous possibilities enabled by the Semantic Web, we are particularly interested in exploiting semantic information in retrieving documents. The OntoSearch system presented in this paper is such an exploration.

Ontologies are the backbone of the Semantic Web by providing the vocabularies and formal conceptualization of a given domain (Gruber 1993) to facilitate information sharing and exchange. In view that domain ontology can capture useful prior knowledge of the concepts in a domain, there have been many systems which utilize domain ontology in document retrieval. For instance, OntoSeek (Guarino, Masolo, & Vetere 1999) made use of ontologies in formulating queries so as to improve the precision of the documents retrieved. Guha *et al.* (2003) employed ontologies to improve traditional web search by augmenting the search results with the related concepts in the ontology. Hyvnen et al. (Hyvnen *et al.* 2004) used domain ontology to build a semantic portal for the Finnish museums on the Semantic Web.

Although many ontology based applications have been developed, all of them require the users to include some forms of semantic annotations explicitly in their queries. For instance, a user of OntoSeek would need to identify the corresponding concepts of his/her query terms from a domain ontology. Khan *et al.* (2004) required the user to write SQL-like queries wherein the exact concepts are incorporated. These applications are thus not suitable for typical information users as it is usually not straightforward to identify the matching concepts of a query from a domain ontology. Contreras *et al.* (2004) enabled a user to submit queries in natural language by using Natural Language Processing (NLP) tool to extract concepts and instances from the queries. However, the performance of their application heavily depended on the quality of the NLP tool. Therefore, a simple and natural search system, whereby users do not need to worry about the matching semantics when forming queries, is preferred.

The OntoSearch system presented in this paper is a full-text search engine for retrieving documents in the Semantic Web. Although it takes keyword based queries as input, an ontology based inference mechanism is combined with the classical keyword based method to yield an enhanced search performance. Specifically, OntoSearch models a domain ontology using a semantic network, wherein a spreading activation procedure infers the relevance of the concepts in the domain ontology, with respect to a given query. The scores of the conceptual relevance are then used to re-rank the documents retrieved using a traditional keyword based retrieval method. To provide personalized information access, we further develop algorithms for learning and exploiting user-specific ontological models, known as user ontologies, each serves as a customized weighted excerpt of the domain ontology. OntoSearch has been applied to the domain of searching scientific publications in the ACM Digital Library. The experimental results have supported the efficacy of the OntoSearch system with the use of domain ontology and user ontology for enhanced search performance.

The rest of this paper is organized as follows. Section II presents how OntoSearch incorporates the semantic network based domain ontology model for document retrieval. The extension to user ontology with its learning and inferencing algorithms are presented in Section III. The experimental results are presented in Section IV. Concluding remarks and future work are in the final section.

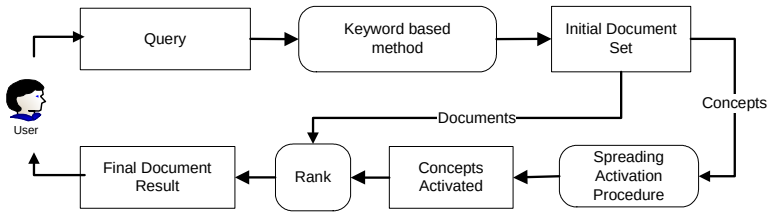


Figure 1: The system flow of OntoSearch for enhanced document retrieval.

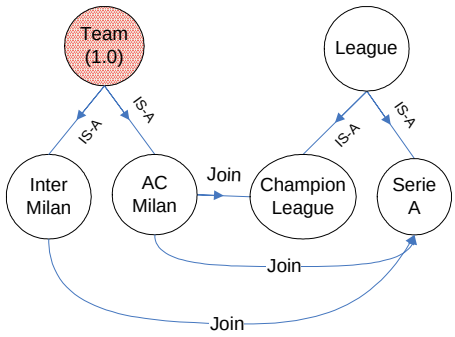


Figure 2: Initial stage of the spreading activation process.

## II. OntoSearch Search Engine

The procedure of the OntoSearch system in handling search queries is highlighted in Figure 1. Similar to that of a traditional search engine, a user submits queries consisting of keywords to the system[1], wherein the corresponding semantic annotation is not required. OntoSearch then returns an initial list of documents obtained with a keyword based search method. Since the documents are pre-annotated with the ontological information, we also obtain a set of the associated concepts based on the documents retrieved. Using these concepts as the seeds to our semantic network based domain ontology, the spreading activation theory (Anderson 1983) process infers the concepts that are semantically related to the initial concept set. Finally, the conceptual relevance scores, in terms of the concept activations in the domain ontology, are used to re-rank the documents before presentation to the user. The detailed algorithms are presented in the following sections.

### Ontological Indexing

We use the classical vector space model (Baeza-Yates & Ribeiro-Neto 1999) to index documents in the OntoSearch system. Given a document $d_j$, it is represented by a vector

$$\overrightarrow{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{m,j}, c_{1,j}, c_{2,j}, \ldots, c_{n,j}),$$

where $m$ is the total number of index keywords in the system, $n$ is the total number of index concepts in the system, $w_{i,j}$ represents the keyword $w_i$'s weight in document $d_j$, and $c_{i,j}$ represents the concept $c_i$'s weight in document $d_j$.

For each keyword $w_i$, its weight $w_{i,j}$ is calculated using the traditional $tf/idf$ measure (Rijsbergen 1979)

$$w_{i,j} = freq_{i,j} \times \log \frac{N}{n_i},$$

where $freq_{i,j}$ represents $w_i$'s frequency in $d_j$, $N$ is the total number of documents, and $n_i$ is the number of documents where the keyword $w_i$ appears.

For each concept $c_i$, we use a simple method to determine its weight $c_{i,j}$. If the concept $c_i$ is specified in the document $d_j$, its weight $c_{i,j}$ is 1, else its weight is 0. This approach is different from the way of pagerank-like algorithms to process conceptual information (Guo *et al.* 2003).

[1]OntoSearch uses the same query syntax as Google to form queries.

### Inferencing in Ontology

When designing the system, we aim to use the traditional way of forming queries. Considering the sample document in Figure 4, a user only needs to use the keywords in the content part to form queries. For instance, one only specifies "Bayesian Network" as a query term to retrieve this document, while the semantic concepts will be extracted automatically by the system. In OntoSearch, the relevant concepts are determined through a spreading activation inference process in the domain ontology.

In the field of cognitive science, one popular form of storing knowledge in long term memory is semantic network (Anderson 1976). Concepts are represented as nodes in the network and 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 an initial input activating specific nodes of the network, after the spreading activation process finishes, each and every concept in the network will be activated with certain values depending on its relations to neighbouring nodes. As spreading activation theory has been proven to be efficient for inferencing and a domain ontology is structurally similar to a semantic network, it is adopted as a natural choice of inferencing in the domain ontology.

An illustration of the spreading activation procedure in a domain ontology is given below. Referring to Figure 2, the node "Team" has been 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, the nodes in the network will be activated with certain activation values such as those shown in Figure 3. 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 more related to "Team" in this semantic network.

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:

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

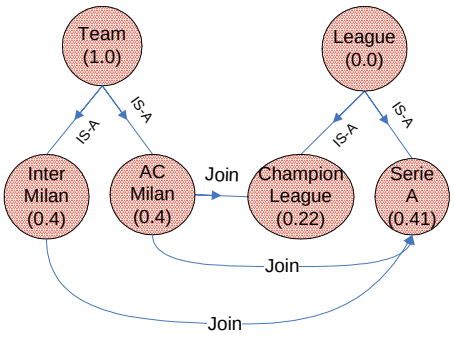


Figure 3: Final stage of the spreading activation process.

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 $t_i$, $w_{xy}$ is the link between nodes $y$ and $x$, and $\alpha$ is a decay factor to represent the energy loss in the spreading activation process. A simplified spreading activation theory is that 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 whole spreading activation process can be calculated using the following formula:

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

where $I = [I_1, \ldots, I_n]^T$ is the initial input to the network, $w$ is the matrix representation of the domain ontology whose element $w_{ij}$ represents the link between concepts $c_i$ and $c_j$, $\alpha$ is the decay factor, $\mathcal{E}$ is an $n \times n$ identity matrix of order $n$, and $O = [O_1, \ldots, O_n]^T$ is the final output vector of the spreading activation process in which $O_i$ is the value of concept $c_i$ obtained from the spreading activation process.

In our case, after a query is submitted to the system, a list of documents are retrieved using the keyword based search method. Since documents have been annotated with concepts, besides the documents retrieved, we obtain a set of associated concepts. The spreading activation theory is 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 $I_q = [I_{1,q}, I_{2,q}, \ldots, I_{n,q}]^T$ as the input to the spreading activation process, where $I_{i,q}$, the input to the concept $c_i$ for query $q$, is calculated by

$$I_{i,q} = \frac{freq(c_i)}{\sum_{c_i} freq(c_i)}, \tag{3}$$

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

Upon receiving the input vector $I_q$, the spreading activation procedure is performed on the domain ontology to infer the concepts of relevance to the user's query $q$. In our system, the configuration of the matrix representation $w$ for the spreading activation procedure is described as follows. We first extract all the semantic relations from the data set. The element $w_{ij}$'s value of the matrix $w$ is the frequency of the semantic relation $r_{ij}$ in the data set. Then, we normalize the matrix using the following formula:

$$w_{ij}^0 = \frac{freq(r_{ij})}{\sum_j freq(r_{ij})}, \tag{4}$$

where $freq(r_{ij})$ represents the frequency of the relation $r_{ij}$ in the data set and $w_{ij}^0$ is an initial estimation of $w_{ij}$.

Using the spreading activation formula (eq 2), we calculate the relevance factor $O_{i,q}$ of each concept $c_i$ with respect to the user's query $q$. Therefore, given a query $q$ submitted to the system, it can be represented by a vector

$$\overrightarrow{q} = (w_{1,q}, w_{2,q}, \ldots, w_{m,q}, c_{1,q}, c_{2,q}, \ldots, c_{n,q}), \tag{5}$$

where $c_{i,q}$, normalized value of $O_{i,q}$, represents the relevance of the concept $c_i$ to $q$, and $w_{i,q}$ represents the keyword $w_i$'s relevance to $q$ .

### Ranking Measure

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

$$sim(d_j, q) = \frac{\overrightarrow{|d_j|} \cdot |\overrightarrow{q}|}{\overrightarrow{|d_j|} \times |\overrightarrow{q}|}. \tag{6}$$

This formula is a classical measure used in the vector space model to calculate a document's similarity to a query. Although several variants of the equation are available, we adopt this version in the OntoSearch system because it outperforms the others in our earlier experiments.

## III. User Ontology

A natural extension of OntoSearch is to provide personalized service. Whereas most existing user modelling methods (Middleton, Shadbolt, & Roure 2004; Pretschner & Gauch 1999) consider only the importance of concepts in representing the users' interest, our user ontology model is based on an excerpt of the domain ontology model, which captures a rich semantics of user profiles. Specifically, each concept and relation in the domain ontology is assigned a specific value for indicating a user's interests. It is a personalized view of the domain conceptualization and yet is more comprehensive than the existing types of user models in representing user's interests in a specific domain.

A user ontology can be defined formally as a structure $\Theta = (C, R, \sigma, \theta, \overrightarrow{c_u}, w_u)$ consisting of

- two disjoint sets $C$ and $R$, whose elements $c_i$ and $r_{ij}$ are the *concepts* and *semantic relations* in the domain ontology, respectively,
- a function $\sigma : c_s \times c_o \rightarrow r_{so}$, in which $c_s$ and $c_o$ are the subject and object of the relation $r_{so}$ respectively, associating pairs of concepts with semantic relations,
- a function $\theta : \theta(C|R)$, which assigns weights to concepts and relations in the domain ontology, representing an individual's view of the particular domain,
- a vector $\overrightarrow{c_u} = [c_{1,u}, \ldots, c_{n,u}]$, in which element $c_{i,u}$ represents a user $u$'s long term interest to concept $c_i$, and

- a matrix $w_u = [w_{ij,u}]$, in which element $w_{ij,u}$ represents the user $u$'s interest to relation $r_{ij}$ and $\sum_j w_{ij,u} = 1$.

The procedure of utilizing the user ontology for providing personalized service is described as follows. Given a query $q$ submitted by a user $u$, the matrix $w_u$ is first used to infer $O_{i,q}$, the estimation of the concept $c_i$'s relevance to the user's current query $q$ (eq 2). Then, the current relevance factor $O_{i,q}$ is combined with the user's long term interest fact $c_{i,u}$ to derive a final score $S_{i,u}$ for the concept $c_i$. The score strikes a balance between user's long time interest and current relevance. In our application, the final score $S_{i,u}$ is computed by

$$S_{i,u} = O_{i,q} + c_{i,u} \times \delta^{-b}, \quad b \in (0,1) \tag{7}$$

where $\delta$ represents the time interval since the last query and $b$ is a real-valued constant to simulate the decay function occurred in the long time memory (Anderson 1993). Finally, the score $S_{i,u}$ will be used to form the query vector $\overrightarrow{q}$ and retrieve documents.

After documents are retrieved, the user may specify which documents are relevant. These relevant documents can be used to update the user ontology $\Theta$. Specifically, we use the following functions in OntoSearch for the update process.

1. For each element $c_{i,u}$ of $\overrightarrow{c_u}$, its value is calculated by

$$c(t+1)_{i,u} = c(t)_{i,u} \times \delta^{-b} + O_{i,q} \tag{8}$$

where $c(t)_{i,u}$ is the user $u$'s long time interest to concept $c_i$ at time $t$, $O_{i,q}$ is the user's current interest for concept $c_i$, and $\delta^{-b}$ is a decay function to prevent saturation of the interest factor $c_i$ (Anderson 1993).

2. For each element $w_{ij,u}$, a typical Bayesian solution (Anderson 1993) computes a weighted average of the initial value and the empirical value as follows:

$$w_{ij,u} = \frac{a \times w^0_{ij,u} + freq(r_{ij})}{a + \sum_j freq(r_{ij})}, \quad a \in (0,1) \tag{9}$$

where $w^0_{ij}$ is the initial estimation of $w_{ij}$, $a$ is a constant to normalize the empirical value and the initial estimation, and $freq(r_{ij})$ is the frequency of the relation $r_{ij}$ in the relevant documents.

## IV. Experiments

### The Data Set and Domain Ontology

As we are not aware of a publicly available document set pre-annotated with an ontology, we use the academic publications in the ACM Digital Library for our experiments. One key advantage of the ACM publications is that they have been annotated with terms according to the ACM Computing Classification System (CCS)[2]. The CCS thus can be treated 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 the documents in the data set is given in Figure 4. We can see that this document contains the keywords such as "Bayesian network", "Markov blanket", and "classifier"". Also, it is annotated with the concept "G.3 Probability and Statistics".

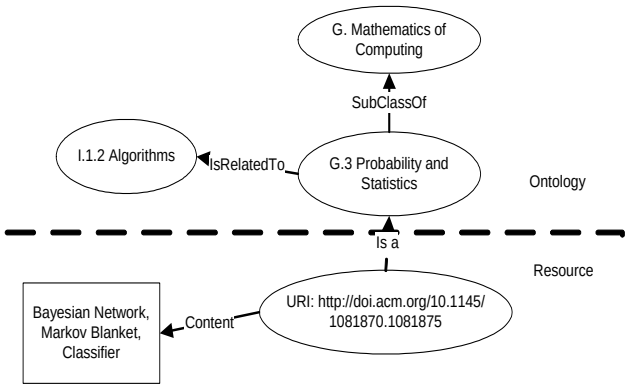


Figure 4: An illustration of a document in the data set.

Because the CCS ontology only contains hierarchical relations. To enrich the CCS ontology with more semantic relation information, we hypothesized that two concepts are related semantically if they are used to index the same paper. For instance, 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 Brinkman's paper (Brinkman & Charikar 2005) on dimensional reduction in $l_1$. This relation can therefore be added into the CCS ontology and 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 (He, Hui, & Fong 2003).

Note that we have adopted an explicit, non-embedded annotation to link a given domain ontology to the documents. For each document, we create a separate file to store the concept and the relation information and bind it together 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 (Mayfield & Finin 2003).

### Using Domain Ontology

Researchers with diverse research profiles in our school were invited to test the OntoSearch's performance in searching publications. Each user was asked to submit a set of queries to the system and judged the relevance of the documents retrieved. Due to the limited scope of our data set, we restrict the test queries to those which have at least five relevant documents in the data set. Based on a set of 30 test queries, we compared OntoSearch with a traditional keyword based search engine known as Lucene (http://lucene.apache.org). As shown in Table 1, we can see that OntoSearch significantly outperforms Lucene in terms of the average precision based on the top 1, top 3, top 5, and top 10 documents retrieved. These results strongly validate

[2]http://www.acm.org/class

| | 1 | 3 | 5 | 10 |
|---|---|---|---|---|
| Lucene Search | 0.70 | 0.49 | 0.48 | 0.45 |
| OntoSearch | 0.83 | 0.64 | 0.65 | 0.52 |

Table 1: The average precision of OntoSearch compared with Lucene.

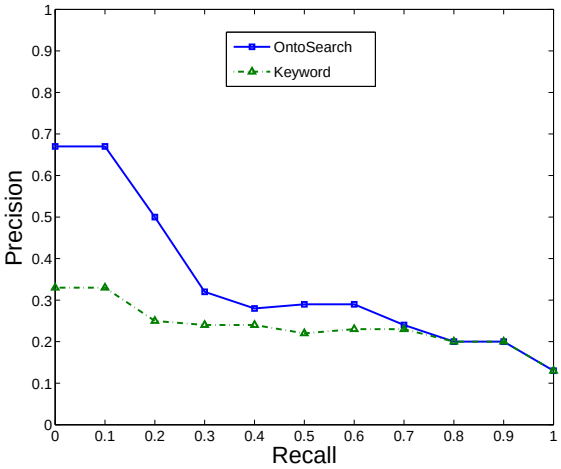


Figure 5: The performance of Lucene and OntoSearch in terms of 11-point average precision measure for a query on "quality of service".

our approach of using the semantic information to enhance search performance.

In Figure 5, we present the 11-point average precision scores of the OntoSearch system and the Lucene search engine in retrieving publications given a query on "*quality of service*". We observe that OntoSearch outperforms the keyword based method significantly when the recall value is low. As the performance of the OntoSearch system also relies on the keyword's weights, the precision of OntoSearch thus decreases for high recall values like the keyword based method. Nevertheless, OntoSearch can still produce performance comparable with those of the keyword based method for these high recall values.

An interesting issue arising from the experimentation of the OntoSearch system is how to determine the initial activation value $I_{i,q}$ for each concept $c_i$ that appears in the initial document set. At present, we compute the $I_{i,q}$ value based on the concept $c_i$'s frequency information (eq 3). An alternative approach is to utilize the concepts' ordering information (Rocha, Schwabe, & de Aragão 2004). Specifically, if a concept $c_i$ appears at the top of the initial document list, the activation value of $c_i$ will be higher than that of another concept $c_j$, which appears at the bottom of the list. Using the same 30 queries, we conducted experiments that incorporated the ordering information in retrieving documents. We used a decreasing function $e^{-i}$ to convert a concept $c_i$'s order information into the corresponding $I_{i,q}$ value. For instance, if the concept $c_i$ occurs in the first document of the

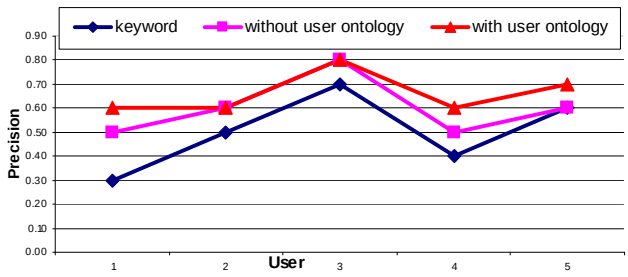


Figure 6: The average precision of OntoSearch with and without user ontology compared with keyword search in document retrieval.

initial document list, its $I_{i,q}$ value is $e^{-1}$. We compared the precision scores based on the top 20 documents retrieved and find no significant difference between the two methods. The results indicate that our approach is rather immune to the variation in determining the initial activation values.

### Using User Ontology

A group of five users was involved in evaluating the user ontology's ability for providing personalized service. Each user provided two sets of queries to the system, one for training the user ontology and the other for testing. When training the user ontology model, one had to browse the top 10 documents returned to his queries and provided feedback to the search engine on the documents that were relevant. These selected documents were then used to update the user ontology model (eq 8 & 9).

After several rounds of training, the performance of the system was evaluated by using the learnt user ontology to provide recommendation for the test queries. For evaluation, we simply measured the average precision of the top 10 documents retrieved. The performance of the OntoSearch system with and without user ontology, compared with the keyword based method, is summarized in Figure 6. We see that OntoSearch with user ontology consistently outperforms or at least produces equivalent performance compared with the other two methods. The results thus validate our approach of using user ontology to enhance search performance in the Semantic Web.

## VI. Conclusion

This paper has presented the OntoSearch system that exploits ontological knowledge in the Semantic Web for document retrieval. Compared with alternative systems, the proposed system has a few unique and important features.

First, semantic information is not required explicitly in the queries for retrieving documents. A user only needs to consider suitable keywords based on the desired content but does not have to specify which concepts the keywords correspond to. For instance, if a user wants to search documents on "Michael Jordan", he would not have to specify that "Michael Jordan" is an instance of the concept "Professor" instead of the concept "NBA player". Existing ontology based search applications (Guarino, Masolo, & Vetere 1999;

Mayfield & Finin 2003; Shah, Finin, & Joshi 2002) certainly outperform the traditional keyword based methods as the semantic information is used explicitly in the queries to retrieve documents. The OntoSearch System, on the other hand, explores possible ways of using these semantic information implicitly to retrieve documents. Therefore, it is more friendly for typical users to retrieve documents in the Semantic Web. Although there are some systems using the keyword based queries for information retrieval in the Semantic Web (Stojanovic 2003; Guha, McCool, & Miller 2003), these applications are meant for searching concepts and instances in a knowledge base, instead of full-text documents. Our approach is actually more similar to a method presented in (Cohen *et al.* 2003) for searching XML files.

Second, we adopt the spreading activation theory for performing inference in the domain ontology. The spreading activation theory has been used to infer concepts of relevance to the user's queries (Rocha, Schwabe, & de Aragão 2004). Our application can be regarded as a further extension of their work, since we combine conceptual information with keywords to retrieve documents. Although it is possible to use some pagerank-like algorithms (Guo *et al.* 2003) to calculate the concepts' relevance, these algorithms assign a static value to each concept no matter which queries are submitted, while the spreading activation theory calculates the concepts of relevance according to the queries submitted. Furthermore, the OntoSearch system with the spreading activation procedure can easily be extended to learn and exploit a personalized view of the domain ontology as user ontology. Compared with rule based inference systems (Mayfield & Finin 2003), our system provides a natural and integrated mechanism for enhanced personalized search.

As OntoSearch adopts a heuristic approach, especially in estimating the initial strengths of the semantic relations in the domain ontology (eq 4), in some rare occasions, OntoSearch could actually perform worse than the traditional keyword based search engine. This is an issue that we are investigating in our ongoing work. So far, we have applied OntoSearch to the domain of searching scientific publications in the ACM Digital Library. Although the initial results are encouraging, the present data set is relatively small. We aim to expand the data set significantly and conduct more extensive experiments. We also look forward to the availability of large scale date sets pre-annotated with associated ontologies, which will enable a rigorous quantitative comparison of our approach with state-of-the-art methods.

## References


Anderson, R. J. 1976. *Language, Memory and Thought*. Hillsdale, N. J.: Erlbaum.

Anderson, R. J. 1983. A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior* 22:261–295.

Anderson, R. J. 1993. *Rules of the Mind*. Hillsdale, N. J.: Erlbaum.

Baeza-Yates, R. A., and Ribeiro-Neto, B. A. 1999. *Modern Information Retrieval*. ACM Press / Addison-Wesley.

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific America*.

Brinkman, B., and Charikar, M. 2005. On the impossibility of dimension reduction in $\ell_1$. *J. ACM* 52(5):766–788.

Cohen, S.; Mamou, J.; Kanza, Y.; and Sagiv, Y. 2003. Xsearch: A semantic search engine for xml. In *VLDB*, 45–56.

Contreras, J.; Benjamins, V. R.; Blázquez, M.; Losada, S.; Salla, R.; Sevilla, J.; Navarro, D.; Casillas, J.; Mompó, A.; Patón, D.; Corcho, Ó.; Tena, P.; and Martos, I. 2004. A semantic portal for the international affairs sector. In *EKAW*, 203–215.

Gruber, T. R. 1993. A translation approach to portable ontology specification. *Knowledge Acquisition* 5:199 – 220.

Guarino, N.; Masolo, C.; and Vetere, G. 1999. Ontoseek: Content-based access to the web. *IEEE Intelligent Systems* 14(3):70–80.

Guha, R.; McCool, R.; and Miller, E. 2003. Semantic search. In *WWW '03*, 700–709.

Guo, L.; Shao, F.; Botev, C.; and Shanmugasundaram, J. 2003. Xrank: Ranked keyword search over xml documents. In *SIGMOD Conference*, 16–27.

He, Y.; Hui, S. C.; and Fong, A. C. M. 2003. Citation-based retrieval for scholarly publications. *IEEE Intelligent Systems* 18(2):58–65.

Hyvnen, E.; Junnila, M.; Kettula, S.; Mkel, E.; Saarela, S.; Salminen, M.; Syreeni, A.; Valo, A.; and Viljanen, K. 2004. Publishing museum collections on the semantic web: the museumfinland portal. In *WWW Alt. '04*, 418–419.

Khan, L.; McLeod, D.; and Hovy, E. 2004. Retrieval effectiveness of an ontology-based model for information selection. *The VLDB Journal* 13(1):71–85.

Mayfield, J., and Finin, T. 2003. Information retrieval on the Semantic Web: Integrating inference and retrieval. In *Proceedings of the SIGIR Workshop on the Semantic Web*.

Middleton, S. E.; Shadbolt, N. R.; and Roure, D. C. D. 2004. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.* 22(1):54–88.

Pretschner, A., and Gauch, S. 1999. Ontology based personalized search. In *ICTAI '99*, 391.

Rijsbergen, C. J. V. 1979. *Information Retrieval*. London: Butterworths, 2nd edition.

Rocha, C.; Schwabe, D.; and de Aragão, M. P. 2004. A hybrid approach for searching in the semantic web. In *WWW '04*, 374–383.

Shah, U.; Finin, T.; and Joshi, A. 2002. Information retrieval on the semantic web. In *CIKM '02*, 461–468.

Stojanovic, N. 2003. On analysing query ambiguity for query refinement: The librarian agent approach. In *ER*, 490–505.