Ontological User Profiles for Personalized Web Search

Ahu Sieg, Bamshad Mobasher, Robin Burke

Center for Web Intelligence
School of Computer Science, Telecommunication and Information Systems
DePaul University
243 South Wabash Avenue, Chicago, Illinois, 60604, USA
{asieg, mobasher, rburke}@cs.depaul.edu

Abstract

The goal of Web search personalization is to tailor search results to a particular user based on that user's interests and preferences, thus allowing for more efficient information access. One of the key factors for effective personalization of information access is the user context. We present an approach to personalized search that involves building models of users context as ontological profiles by assigning implicitly derived interest scores to existing concepts in a domain ontology. A spreading activation algorithm is used to maintain the interest scores based on the user's ongoing behavior. Our experiments show that re-ranking the search results based on the interest scores and the semantic evidence in an ontological user profile is effective in presenting the most relevant results to the user.

Introduction

Web personalization alleviates the burden of information overload by tailoring the information presented based on an individual user's needs. One of the key factors for accurate personalized information access is user context. A system that does not know who is asking for information and for what purpose will never be able to provide more than very general answers.

In recent years, personalized search has attracted interest in the research community as a means to decrease search ambiguity and return results that are more likely to be interesting to a particular user and thus providing more effective and efficient information access (Singh & Nakata 2005; Shen, Tan, & Zhai 2005; Aktas, Nacar, & Menczer 2004; Liu, Yu, & Meng 2004).

Despite their popularity, users' interactions with Web search engines can be characterized as one size fits all (Allan *et al.* 2003). The representation of user preferences, search context, or the task context is generally non-existent in most search engines (Lawrence 2000). Indeed, contextual retrieval has been identified as a long-term challenge in information retrieval. Allan et al. (Allan *et al.* 2003) define the problem of *contextual retrieval* as follows: "Combine search technologies and knowledge about query and user context into a single framework in order to provide the most appropriate answer for a user's information needs."

Researchers have long been interested in the role of context in a variety of fields including artificial intelligence, context-aware applications, and information retrieval. The notion of *context* may refer to a diverse range of ideas depending on the nature of the work being performed (Finkelstein *et al.* 2002).

Here we consider three essential elements that collectively play a critical role in personalized Web information access. These three independent but related elements are the user's short-term information need, semantic knowledge about the domain being investigated, and the user's profile that capture long-term interests. Each of these elements are considered to be critical sources of contextual evidence, a piece of knowledge that supports the disambiguation of the user's context for information access.

In this paper, we present a novel approach for building ontological user profiles by assigning interest scores to existing concepts in a domain ontology. These profiles are maintained and updated as annotated specializations of a pre-existing reference domain ontology. We propose a spreading activation algorithm for maintaining the interest scores in the user profile based on the user's ongoing behavior. Our experimental results show that re-ranking the search results based on the interest scores and the semantic evidence in an ontological user profile successfully provides the user with a personalized view of the search results by bringing results closer to the top when they are most relevant to the user.

Ontologies for Web Personalization

Our goal is to utilize the user context to personalize search results by re-ranking the results returned from a search engine for a given query. Our unified context model for a user is represented as an instance of a reference domain ontology in which concepts are annotated by *interest scores* derived and updated implicitly based on the user's information access behavior. We call this representation an *ontological user profile*.

An ontology is a specification of a conceptualization - description of the concepts and relationships that can exist for an agent/user or a community of agents/users (Gruber 1993). One increasingly popular method to mediate information access is through the use of ontologies (Haav & Lubi 2001; Ravindran & Gauch 2004). Researchers have attempted to utilize ontologies for improving navigation effectiveness as

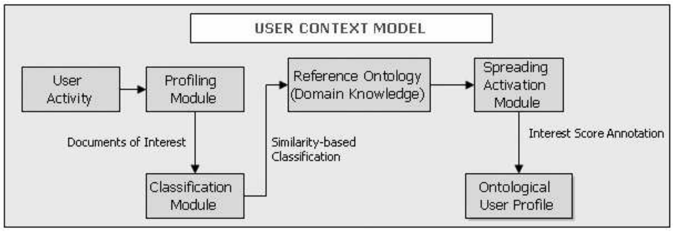
USER CONTEXT MODEL
User Activity
Profiling Module
Reference Ontology (Domain Knowledge)
Spreading Activation Module
Documents of Interest
Similarity-based Classification
Interest Score Annotation
Classification Module
Ontological User Profile

Figure 1: Ontological User Profile as the Context Model

well as personalized Web search and browsing, specifically when combined with the notion of automatically generating semantically enriched ontology-based user profiles (Gauch, Chaffee, & Pretschner 2003; Ravindran & Gauch 2004).

Since semantic knowledge is an essential part of the user context, we use a domain ontology as the fundamental source of semantic knowledge in our framework. An ontological approach to user profiling has proven to be successful in addressing the *cold-start problem* in recommender systems where no initial information is available early on upon which to base recommendations (Middleton, Shadbolt, & Roure 2003). When initially learning user interests, systems perform poorly until enough information has been collected for user profiling. Using ontologies as the basis of the profile allows the initial user behavior to be matched with existing concepts in the domain ontology and relationships between these concepts.

In (Trajkova & Gauch 2004), the similarity is calculated between the Web pages visited by a user and the concepts in a domain ontology. After annotating each concept with a weight based on an accumulated similarity score, a user profile is created consisting of all concepts with non-zero weights.

In our approach, the purpose of using an ontology is to identify topics that might be of interest to a specific Web user. Therefore, we define our ontology as a hierarchy of topics, where the topics are utilized for the classification and categorization of Web pages. The hierarchical relationship among the concepts is taken into consideration for building the ontological user profile as we update the annotations for existing concepts using spreading activation.

Ontological User Profiles

The Web search personalization aspect of our research is built on the previous work in ARCH (Sieg *et al.* 2004). In ARCH, the initial query is modified based on the user's interaction with a concept hierarchy which captures the domain knowledge. This domain knowledge is utilized to disambiguate the user context.

In the present framework, the *user context* is represented using an *ontological user profile*, which is an annotated instance of a reference ontology. Figure 1 depicts a high-level picture of our proposed context model based on an *ontological user profile*. When disambiguating the context, the domain knowledge inherent in an existing reference ontology is called upon as a source of key domain concepts.

Each ontological user profile is initially an instance of the reference ontology. Each concept in the user profile is annotated with an *interest score* which has an initial value of one. As the user interacts with the system by selecting or viewing new documents, the ontological user profile is updated and the annotations for existing concepts are modified by spreading activation. Thus, the *user context* is maintained and updated incrementally based on user's ongoing behavior.

Accurate information about the user's interests must be collected and represented with minimal user intervention. This can be done by passively observing the user's browsing behavior over time and collecting Web pages in which the user has shown interest. Several factors, including the frequency of visits to a page, the amount of time spent on the page, and other user actions such as bookmarking a page can be used as bases for heuristics to automatically collect these documents (Dumais *et al.* 2003).

Representation of Reference Ontology

Our current implementation uses the *Open Directory Project*[1], which is organized into a hierarchy of topics and Web pages that belong to these topics. We utilize the Web pages as training data for the representation of the concepts in the reference ontology. The textual information that can get extracted from Web pages explain the semantics of the concepts and is learned as we build a term vector representation for the concepts.

We create an aggregate representation of the reference ontology by computing a term vector $\vec{n}$ for each concept n in the concept hierarchy. Each concept vector represents, in aggregate form, all individual training documents indexed under that concept, as well as all of its subconcepts.

We begin by constructing a global dictionary of terms extracted from the training documents indexed under each concept. A stop list is used to remove high frequency, but semantically non-relevant terms from the content. Porter stemming (Porter 1980) is utilized to reduce words to their stems. Each document d in the training data is represented as a term vector $\vec{d} = \langle w_1, w_2, ..., w_k \rangle$, where each term weight, w_i, is computed using term frequency and inverse document frequency (Salton & McGill 1983). Specifically, $w_i = tf_i * \log(N/n_i)$, where tf_i is the frequency of term i in document d, N is the total number of documents in the training set, and n_i is the number of documents that contain

[1] http://www.dmoz.org

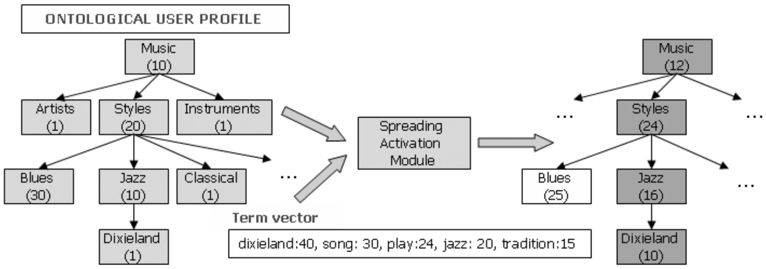

Figure 2: Portion of an Ontological User Profile where Interest Scores are updated based on Spreading Activation

term i. We further normalize each document vector, so that $\vec{d}$ represents a term vector with unit length.

The aggregate representation of the concept hierarchy can be described more formally as follows. Let $S(n)$ be the set of subconcepts under concept n as non-leaf nodes. Also, let $\{d_1^n, d_2^n, ..., d_{k_n}^n\}$ be the individual documents indexed under concept n as leaf nodes. $Docs(n)$, which includes of all of the documents indexed under concept n along with all of the documents indexed under all of the subconcepts of n is defined as:

$$Docs(n) = [\bigcup_{n' \in S(n)} Docs(n')] \cup \{d_1^n, d_2^n, ..., d_{k_n}^n\}$$

The concept term vector $\vec{n}$ is then computed as:

$$\vec{n} = \left[\sum_{d \in Docs(n)} \vec{d} \right] / \, |Docs(n)|$$

Thus, $\vec{n}$ represents the centroid of the documents indexed under concept n along with the subconcepts of n. The resulting term vector is normalized into a unit term vector.

Context Model

Figure 2 depicts a portion an ontological user profile corresponding to the node *Music*. The interest scores for the concepts are updated with spreading activation using an input term vector.

Each node in the ontological user profile is a pair, $\langle C_j, IS(C_j) \rangle$, where C_j is a concept in the reference ontology and $IS(C_j)$ is the interest score annotation for that concept. The input term vector represents the active interaction of the user, such as a query or localized context of current activity.

Based on the user's information access behavior, let's assume the user has shown interest in *Dixieland Jazz.* Since the input term vector contains terms that appear in the term vector for the *Dixieland* concept, as a result of spreading activation, the interest scores for the *Dixieland*, *Jazz*, *Styles*, and *Music* concepts get incremented whereas the interest score for *Blues* gets decreased. The *Spreading Activation* algorithm and the process of updating the interest scores are discussed in detail in the next section.

Learning Profiles by Spreading Activation

We use *Spreading Activation* to incrementally update the *interest score* of the concepts in the user profiles. Therefore, the ontological user profile is treated as the semantic network and the interest scores are updated based on activation values.

Traditionally, the spreading activation methods used in information retrieval are based on the existence of maps specifying the existence of particular relations between terms or concepts (Salton & Buckley 1988). (Alani, O'Hara, & Shadbolt 2002) use spreading activation to search ontologies in Ontocopi, which attempts to identify communities of practice in a particular domain. In (Rocha, Schwabe, & de Aragao 2004), the authors use spreading activation to find related concepts in an ontology given an initial set of concepts and corresponding initial activation values.

In our approach, we use a very specific configuration of spreading activation, depicted in Algorithm 1, for the sole purpose of maintaining *interest scores* within a user profile. We assume a model of user behavior can be learned through the passive observation of user's information access activity and Web pages in which the user has shown interest in can automatically be collected for user profiling.

The algorithm has an initial set of concepts from the ontological user profile. These concepts are assigned an initial activation value. The main idea is to activate other concepts following a set of weighted relations during propagation and at the end obtain a set of concepts and their respective activations.

As any given concept propagates its activation to its neighbors, the weight of the relation between the origin concept and the destination concept plays an important role in the amount of activation that is passed through the network. Thus, a one-time computation of the weights for the relations in the network is needed. Since the nodes are organized into a concept hierarchy derived from the domain ontology, we compute the weights for the relations between each concept and all of its subconcepts using a measure of containment. The containment weight produces a range of values between zero and one such that a value of zero indicates no overlap between the two nodes whereas a value of one indicates complete overlap.

The weight of the relation w_{is} for concept i and one of its subconcepts s is computed as $w_{is} = \frac{\vec{n}_i \cdot \vec{n}_s}{\vec{n}_i \cdot \vec{n}_i}$, where $\vec{n}_i$ is the term vector for concept i and $\vec{n}_s$ is the term vector for subconcept s. Once the weights are computed, we process the weights again to ensure the total sum of the weights of the relations between a concept and all of its subconcepts equals to 1.

```
Input: Ontological user profile with interest scores and a set of documents
Output: Ontological user profile concepts with updated activation values
CON = $\{C_1, ..., C_n\}$, concepts with interest scores
$IS(C_j)$, interest score
$IS(C_j) = 1$, no interest information available
$I = \{d_1, ..., d_n\}$, user is interested in these documents
foreach $d_i \in I$ do
    Initialize $priorityQueue$;
    foreach $C_j \in CON$ do
        $C_j.Activation = 0$; // Reset activation value
    end
    foreach $C_j \in CON$ do
        Calculate $sim(d_i, C_j)$;
        if $sim(d_i, C_j) > 0$ then
            $C_j.Activation = IS(C_j) * sim(d_i, C_j)$;
            $priorityQueue.Add(C_j)$;
        else
            $C_j.Activation = 0$;
        end
    end
    while $priorityQueue.Count > 0$ do
        Sort $priorityQueue$; // activation values(descending)
        $C_s = priorityQueue[0]$; // first item(spreading concept)
        $priorityQueue.Dequeue(C_s)$; // remove item
        if passRestrictions($C_s$) then
            $linkedConcepts = GetLinkedConcepts(C_s)$;
            foreach $C_l$ in linkedConcepts do
                $C_l.Activation+ = C_s.Activation * C_l.Weight$;
                $priorityQueue.Add(C_l)$;
            end
        end
    end
end
```

Algorithm 1: Spreading Activation Algorithm

The algorithm considers in turn each of the documents assumed to represent the current context. For each iteration of the algorithm, the initial activation value for each concept in the user profile is reset to zero. We compute a term vector for each document d_i and compare the term vector for d_i with the term vectors for each concept C_j in the user profile using a cosine similarity measure. Those concepts with a similarity score, $sim(d_i, C_j)$, greater than zero are added in a priority queue, which is in a non-increasing order with respect to the concepts' activation values.

The activation value for concept C_j is assigned to $IS(C_j) * sim(d_i, C_j)$, where $IS(C_j)$ is the existing interest score for the specific concept. The concept with the highest activation value is then removed from the queue and processed. If the current concept passes through restrictions, it propagates its activation to its neighbors. The amount of activation that is propagated to each neighbor is proportional to the weight of the relation. The neighboring concepts which are activated and are not currently in the priority queue are added to queue, which is then reordered. The process repeats itself until there are no further concepts to be processed in the priority queue.

The neighbors for the spreading concept are considered to be the linked concepts. For a given spreading concept, we can ensure the algorithm processes each edge only once by iterating over the linked concepts only one time. The order of the iteration over the linked concepts does not affect the results of activation. The linked concepts that are activated are added to the existing priority queue, which is then sorted with respect to activation values.

The interest score for each concept in the ontological user profile is then updated using Algorithm 2. First the resulting activation value is added to the existing interest score. The interest scores for all concepts are then treated as a vector, which is normalized to a unit length using a pre-defined constant, k, as the length of the vector. Rather than gradually increasing the interest scores, we utilize normalization so that the interest scores can get decremented as well as getting incremented. The concepts in the ontological user profile are updated with the normalized interest scores.

```
Input: Ontological user profile concepts with updated activation values
Output: Ontological user profile concepts with updated interest scores
CON = $\{C_1, ..., C_n\}$, concepts with interest scores
$IS(C_j)$, interest score
$C_j.Activation$, activation value resulting from Spreading Activation
k, constant
$n = 0$;
foreach $C_j \in CON$ do
    $IS(C_j) = IS(C_j) + C_j.Activation$;
    $n = n + (IS(C_j))^2$; // sum of squared interest scores
    $n = \sqrt{n}$; // square root of sum of squared interest scores
end
foreach $C_j \in CON$ do
    $IS(C_j) = (IS(C_j) * k)/n$; // normalize to constant length
end
```

Algorithm 2: Algorithm for the Normalization and Updating of Interest Scores in the Ontological User Profile

Search Personalization

Our goal is to utilize the user context to personalize search results by re-ranking the results returned from a search engine for a given query. Figure 3 displays our approach for search personalization based on ontological user profiles.

Assuming an ontological user profile with interest scores exists and we have a set of search results, Algorithm 3 is utilized to re-rank the search results based on the interest scores and the semantic evidence in the user profile.

A term vector $\vec{r}$ is computed for each document $r \in R$, where R is the set of search results for a given query. The term weights are obtained using the *tf.idf* formula described earlier. To calculate the rank score for each document, first

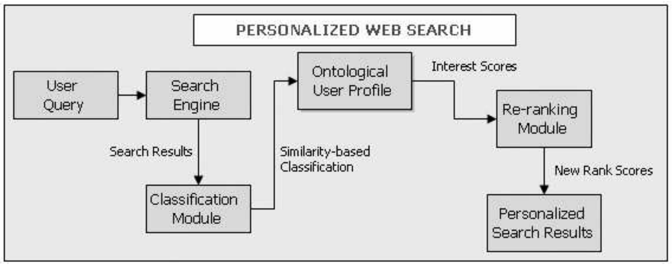

Figure 3: Personalized Web Search based on Ontological User Profiles

```
Input: Ontological user profile with interest scores and a set of search results
Output: Re-ranked search results
CON = $\{C_1, ..., C_n\}$, concepts with interest scores
$IS(C_j)$, interest score
$R = \{d_1, ..., d_n\}$, search results from query q
foreach $d_i \in R$ do
    Calculate $sim(d_i, q)$;
    maxSim = 0;
    foreach $C_j \in CON$ do
        Calculate $sim(d_i, C_j)$;
        if $sim(d_i, C_j) \geq maxSim$ then
            (Concept)c = $C_j$;
            maxSim = $sim(d_i, C_j)$;
        end
    end
    Calculate $sim(q, c)$;
    if $IS(c) > 1$ then
        rankScore($d_i$) = $IS(c) * \alpha * sim(d_i, q) * sim(q, c)$;
    else
        rankScore($d_i$) = $IS(c) * sim(d_i, q) * sim(q, c)$;
    end
end
Sort R based on rankScore;
```

Algorithm 3: Re-ranking Algorithm

the similarity of the document and the query is computed using a cosine similarity measure. Then, we compute the similarity of the document with each concept in the user profile to identify the best matching concept. Once the best matching concept is identified, a rank score is assigned to the document by multiplying the interest score for the concept, the similarity of the document to the query, and the similarity of the specific concept to the query. If the interest score for the best matching concept is greater than one, it is further boosted by a tuning parameter α. Once all documents have been processed, the search results are sorted in descending order with respect to this new rank score.

Experimental Evaluation

Since the queries of average Web users tend to be short and ambiguous (Spink *et al.* 2002), our goal is to demonstrate that re-ranking based on ontological user profiles can help in disambiguating the user's intent particularly when such queries are used. We measure the effectiveness of re-ranking in terms of *Top-n Recall* and *Top-n Precision*.

Evaluation Methodology and Experimental Data Sets

As of December 2006, the *Open Directory* contained more than 590,000 concepts. For experimental purposes, we decided to use a branching factor of three with a depth of ten levels in the hierarchy. Our experimental data set contained 506 concepts in the hierarchy and a total of 8857 documents that were indexed under various concepts.

We processed the indexed documents into three separate sets including a *training set*, a *test set*, and a *profile set*. For each concept, we used 60 percent of the associated documents for the training set, 20 percent for the test set, and the remaining 20 percent for the profile set. For all of the data sets, we kept track of which concepts these documents were originally indexed under in the hierarchy. The *training set* was utilized for the representation of the reference ontology, the *profile set* was used for spreading activation, and the *test set* was utilized as the document collection for searching.

The *training set* consisted of 5157 documents which were used for the one-time learning of the reference ontology. The concept terms and corresponding term weights were computed using the formula described in the Representation of Reference Ontology section.

Query	# of Terms	Criteria
Set 1	1	highest weighing term in concept term vector
Set 2	2	two highest weighing terms in concept term vector
Set 3	3	three highest weighing terms in concept term vector
Set 4	2 or more	overlapping terms within highest weighing 10 terms

Table 1: Set of Keyword Queries

A total of 1675 documents were included in the *test set*, which were used as the document collection for performing our search experiments. Depending on the search query, each document in our collection can be treated as a signal or a noise document. The signal documents are those documents relevant to a particular concept that should be ranked high in the search results for queries related to that concept. The noise documents are those documents that should be ranked low or excluded from the search results.

The *test set* documents that were originally indexed under a specific concept and all of its subconcepts were treated as signal documents for that concept whereas all other test set documents were treated as noise. In order to create an index for the signal and noise documents, a *tf.idf* weight was

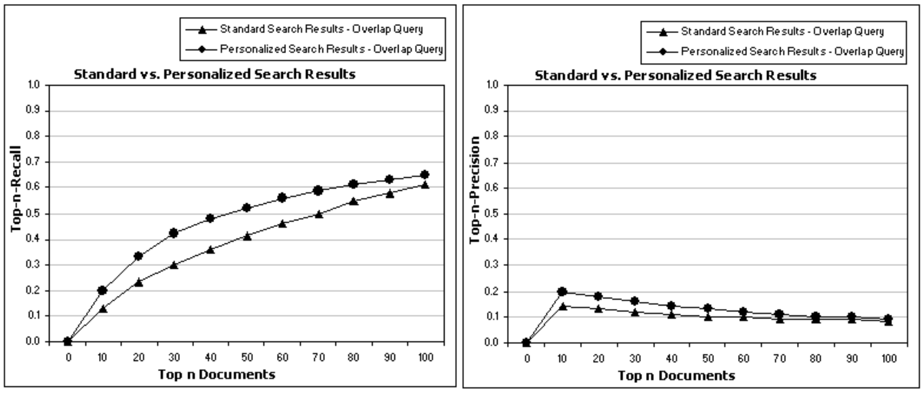

Figure 4: Average *Top-n Recall* and *Top-n Precision* comparisons between the personalized search and standard search using "overlap queries".

computed for each term in the document collection using the global dictionary of the reference ontology.

The *profile set* consisted of 2000 documents, which were treated as a representation of specific user interest for a given concept to simulate ontological user profiles. As we performed the automated experiments for each concept/query, only the profile documents that were originally indexed under that specific concept were utilized to build an ontological user profile by updating the interest scores with the spreading activation algorithm.

We constructed keyword queries to run our automated experiments. We decided to extract the query terms from the concept term vectors in the ontology. Each concept term vector was sorted in descending order with respect to term weights. Table 1 depicts the four query sets that were automatically generated for evaluation purposes.

Our keyword queries were used to run a number of automated search scenarios for each concept in our reference ontology. The first set of keyword queries contained only one term and included the highest weighing term for each concept. In order to evaluate the search results when a single keyword was typed by the user as the search query, the assumption was that the user was interested in the given concept.

The second set of queries contained two terms including the two highest weighing terms for each concept. The third set of queries were generated using the three highest weighing terms for each concept. As the number of keywords in a query increase, the search query becomes less ambiguous.

Even though one to two keyword queries tend to be vague, we intentionally came up with a fourth query set to focus specifically on ambiguous queries. We generated this query set by computing the overlapping terms using the highest weighing ten terms in each concept term vector. Only the overlapping concepts were included in the experimental set with each query consisting of two or more overlapping terms within these concepts.

Our evaluation methodology was as follows. We used the system to perform a standard search for each query. As mentioned above, each query was designed for running our experiments for a specific concept. In the case of standard search, a term vector was built using the original keyword(s) in the query text. Removal of stop words and stemming was utilized. Each term in the original query was assigned a weight of 1.0.

The search results were retrieved from the test set, the signal and noise document collection, by using a cosine similarity measure for matching. Using an interval of ten, we calculated the *Top-n Recall* and *Top-n Precision* for the search results.

Starting with the top one hundred results and going down to top ten search results, the values for *n* included $n = \{100, 90, 80, 70, ..., 10\}$. The *Top-n Recall* was computed by dividing the number of signal documents that appeared within the top *n* search results at each interval with the total number of signal documents for the given concept. We also computed the *Top-n Precision* at each interval by dividing the number of signal documents that appeared within the top *n* results with *n*.

For example, at $n = 100$, the top 100 search results were included in the computation of recall and precision, whereas at $n = 90$, only the top 90 results were taken into consideration.

Next, documents from the profile set were utilized to simulate user interest for the specific concept. For each query, we started with a new instance of the ontological user profile with all interest scores initialized to one. Such a user profile represents a situation where no initial user interest information is available. We performed our spreading activation algorithm to update interest scores in the ontological user profile.

After building the ontological user profile, we sorted the original search results based on our re-ranking algorithm and computed the *Top-n Recall* and *Top-n Precision* with the personalized results.

In order to compare the standard search results with the personalized search results, we computed the average *Top-n Recall* and *Top-n Precision*, depicted in Figure 4. We have also computed the percentage of improvement between standard and personalized search for *Top-n Recall* and *Top-n*

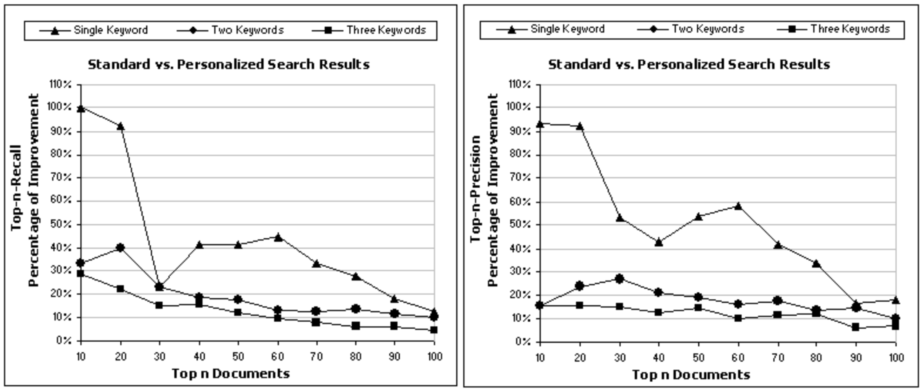

Figure 5: Percentage of improvement in *Top-n Recall* and *Top-n Precision* achieved by personalized search relative to standard search with various query sizes.

Precision, depicted in Figure 5.

Discussion of Experimental Results

Every user has a distinct background and a specific goal when searching for information through entering keyword queries into a search engine. The user queries are typically ambiguous and contain between one to three keywords. The search results that are returned from the search engine may satisfy the search criteria but often fail to meet the user's search intention. Personalized search provides the user with results that accurately satisfy their specific goal and intent for the search.

The queries used in our experiments were intentionally designed to be short to demonstrate the effectiveness of our Web search personalization approach, especially in the typical case of Web users who tend to use very short queries. Simulating user behavior allowed us to run automated experiments with a larger data set.

In the worst case scenario, the user would enter only a single keyword. The evaluation results show significant improvement in recall and precision for single keyword queries as well as gradual enhancement for two-term and three-term queries. As the number of keywords in a query increase, the search query becomes more clear.

In addition to the one, two, and three keyword queries, we ran experiments with the overlap query set to focus on ambiguous queries. Two users may use the exact same keyword to express their search interest even though each user has a completely distinct intent for the search. For example, the keyword *Python* may refer to *python as a snake* as well as the *Python programming language* sense. The purpose of the overlap queries is to simulate real user behavior where the user enters a vague keyword query as the search criteria. Our evaluation results verify that using the ontological user profiles for personalizing search results is an effective approach. Especially with the overlap queries, our evaluation results confirm that the ambiguous query terms are disambiguated by the semantic evidence in the ontological user profiles.

Conclusions and Outlook

We have presented a framework for contextual information access using ontologies and demonstrated that the semantic knowledge embedded in an ontology combined with long-term user profiles can be used to effectively tailor search results based on users' interests and preferences.

In our future work we plan to evaluate the stability and convergence properties of the ontological profiles as interest scores are updated over consecutive interactions with the system.

We plan to design experiments to determine when a user profile becomes stable and starts accurately representing user interests. Every time a new Web page, which the user has shown interest in, is processed via spreading activation, the interest scores for the concepts in the ontological user profile are updated. Initially, the interest scores for the concepts in the profile will continue to change. However, once enough information has been processed for profiling, the amount of change in interest scores should decrease. Our expectation is that eventually the concepts with the highest interest scores should become relatively stable. Therefore, these concepts will reflect the user's primary interests.

Since we focus on implicit methods for constructing the user profiles, the profiles need to adapt over time. Our future work will also involve designing experiments that will allow us to monitor user profiles over time to ensure the incremental updates to the interest scores accurately reflect changes in user interests.

References

Aktas, M.; Nacar, M.; and Menczer, F. 2004. Using hyperlink features to personalize web search. In *Proceedings of the 6th International Workshop on Knowledge Discovery from the Web, WebKDD 2004*.

Alani, H.; O'Hara, K.; and Shadbolt, N. 2002. Ontocopi: Methods and tools for identifying communities of practice. In *Proceedings of the IFIP 17th World Computer Congress - TC12 Stream on Intelligent Information Processing*, 225–236.

Allan, J.; et. al. 2003. Challenges in information retrieval and language modeling. *ACM SIGIR Forum* 37(1):31–47.

Dumais, S.; Joachims, T.; Bharat, K.; and Weigend, A. 2003. Sigir 2003 workshop report: Implicit measures of user interests and preferences. *ACM SIGIR Forum* 37(2):50–54.

Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; and Ruppin, E. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems* 20(1):116–131.

Gauch, S.; Chaffee, J.; and Pretschner, A. 2003. Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems* 1(3-4):219–234.

Gruber, T. R. 1993. Towards principles for the design of ontologies used for knowledge sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation*.

Haav, H., and Lubi, T. 2001. A survey of concept-based information retrieval tools on the web. In *5th East-European Conference, ADBIS 2001*, 29–41.

Lawrence, S. 2000. Context in web search. *IEEE Data Engineering Bulletin* 23(3):25–32.

Liu, F.; Yu, C.; and Meng, W. 2004. Personalized web search for improving retrieval effectiveness. *IEEE Transactions on Knowledge and Data Engineering* 16(1):28–40.

Middleton, S.; Shadbolt, N.; and Roure, D. D. 2003. Capturing interest through inference and visualization: Ontological user profiling in recommender systems. In *Proceedings of the International Conference on Knowledge Capture, K-CAP 2003*, 62–69.

Porter, M. 1980. An algorithm for suffix stripping. *Program* 14(3):130–137.

Ravindran, D., and Gauch, S. 2004. Exploting hierarchical relationships in conceptual search. In *Proceedings of the 13th International Conference on Information and Knowledge Management, ACM CIKM 2004*.

Rocha, C.; Schwabe, D.; and de Aragao, M. P. 2004. A hybrid approach for searching in the semantic web. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, 374–383.

Salton, G., and Buckley, C. 1988. On the use of spreading activation methods in automatic information. In *Proceedings of the 11th annual international ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 1988*, 147–160.

Salton, G., and McGill, M. 1983. *Introduction to Modern Information Retrieval*. New York, NY: McGraw-Hill.

Shen, X.; Tan, B.; and Zhai, C. 2005. Ucair: Capturing and exploiting context for personalized search. In *Proceedings of the Information Retrieval in Context Workshop, SIGIR IRiX 2005*.

Sieg, A.; Mobasher, B.; Lytinen, S.; and Burke, R. 2004. Using concept hierarchies to enhance user queries in web-based information retrieval. In *Proceedings of the International Conference on Artificial Intelligence and Applications, IASTED 2004*.

Singh, A., and Nakata, K. 2005. Hierarchical classification of web search results using personalized ontologies. In *Proceedings of the 3rd International Conference on Universal Access in Human-Computer Interaction, HCI International 2005*.

Spink, A.; Ozmutlu, H.; Ozmutlu, S.; and Jansen, B. 2002. U.s. versus european web searching trends. *ACM SIGIR Forum* 15(2).

Trajkova, J., and Gauch, S. 2004. Improving ontology-based user profiles. In *Proceedings of the Recherche d'Information Assiste par Ordinateur, RIAO 2004*, 380–389.