Personalized Cluster-based Semantically Enriched Web Search for E-learning

Leyla Zhuhadar  
University of Louisville  
Dept of Computer Engineering and Computer Science  
University of Louisville, Louisville, KY 40292, USA  
leyla.zhuhadar@wku.edu

Olfa Nasraoui  
University of Louisville  
Dept of Computer Engineering and Computer Science  
University of Louisville, Louisville, KY 40292, USA  
olfa.nasraoui@louisville.edu

ABSTRACT

We present an approach for personalized search in an e-learning platform, that takes advantage of semantic Web standards (RDF and OWL) to represent the content and the user profiles. Personalizing the finding of needed information in an e-learning environment based on context requires intelligent methods for representing and matching the learning needs and the variety of learning contexts. Our framework consists of the following phases: (1) building the semantic e-learning domain using the known college and course information as concepts and sub-concepts in a lecture ontology, (2) generating the semantic learner’s profile as an ontology from navigation logs that record which lectures have been accessed, (3) clustering the documents to discover more refined sub-concepts (top terms in each cluster) than provided by the available college and course taxonomy, (4) re-ranking the learner’s search results based on the matching concepts in the learning content and the user profile, and (5) providing the learner with semantic recommendations during the search process, in the form of terms from the closest matching clusters of their profile. One important aspect of our approach is the combination of an authoritatively supplied taxonomy by the colleges, with the data driven extraction (via clustering) of a taxonomy from the documents themselves, thus making it easier to adapt to different learning platforms, and making it easier to evolve with the document/lecture collection. Our experimental results show that the learner’s context can be effectively used for improving the precision and recall in e-learning search, particularly by re-ranking the search results based on the learner’s past activities.

Categories and Subject Descriptors

H.3.3 [Information Systems]: Information Storage and Retrieval-Information Search and Retrieval

General Terms

Design, Algorithms, Experimentation

Keywords

Search Personalization, Ontology Student Profile, Information Retrieval

1. INTRODUCTION AND RELATED WORK

Personalizing the search for needed information in an e-learning environment based on context requires intelligent methods for representing and matching both the learning resources and the variety of learning contexts. On the one hand, semantic web technologies can provide a representation of the learning content (lectures). On the other hand, the semantics of the user interests or profiles can form a good representation of the learning context, that promises to enhance the results of retrieval via personalization. The key knowledge nugget in any personalization strategy for e-learning is an accurate user model. User Modeling is an active research area in information retrieval and personalization, especially when abstracting the user away from the problem [1], an abstraction that has, over the years, contributed to the design of more effective retrieval systems. Despite this improvement, the main focus in most information retrieval systems, for the past decade, has been on models that are “good for all users” [1], and not for a specific user. The enormous increase of information on the web led the information retrieval community to strive toward changing the concept of “good for all” to “good for everyone”. This in turn popularized personalized semantic search engines and semantically enhanced recommendation systems, with some related work in [10, 4, 12, 7, 8, 9, 6, 3, 15]. The need of an infrastructure that can provide, manage, and collect data that permits high levels of adaptability and relevance to learners’ profiles with innovative algorithms is a key requirement for the new wave of educational intelligence. Within this context, the semantic web is already playing a crucial factor that can contribute to the design of this infrastructure. Several approaches related to semantic e-learning have been proposed. The following list is not exhaustive, but provides a good overview of the relevant studies:

1- Ontologizing knowledge flows: Adding an ontology layer to the learning content [2] or creating learning paths on top of domain ontologies [13].

2- Semantic Modeling for e-learners: The development
of programs based on learning experiences and directed to learners’ profiles[14].

3- Semantic Annotation in e-learning content: Semantic Web Technologies, such as OWL and RDF, can integrate learning object components in a hierarchical structure.

4- Semantic Social-Networks: Promoting a semantic vision of the e-learning that can change the view from individual learning to collaborative learning. For example, OpenLearn3 at the Open University, UK enables users to learn together.

In this paper, we propose a framework that consists of the following phases: (1) building the semantic e-learning domain using the known college and course information as concepts and sub-concepts in a lecture ontology, (2) generating the semantic learners’ profiles (thus their user models) as an ontology from their navigation logs that record which lectures have been accessed, (3) clustering the documents to discover more refined sub-concepts (top terms in each cluster) than provided by the available college and course taxonomy, (4) re-ranking the learner’s search results based on the matching concepts in the learning content and the user profile, and (5) providing the learner with semantic recommendations during the search process, in the form of terms from the closest matching clusters of their profile.

Of all related work, the idea in [15] seems to be the closest to ours. However, there are major differences between our approach and theirs, which are the following: (1) our study focuses on domain specific retrieval (the domain being e-learning), (2) our search engine provides not only re-ranking based on the user’s profile, but also on clustering-based similarity metrics, and (3) our search engine can recommend semantic terms that are extracted from the closest cluster’s description. These term recommendations help the user (learner) navigate into refined concepts that are related to his/her query. Being based on document clustering, the recommended terms and the refined sub-concepts in the content and user profile ontologies, transcend the rigid taxonomy provided by the college and course information by adapting to the specific lecture contents in the learning platform.

2. MOTIVATIONS

2.1 Why did we need a Semantic Search Engine?
Western Kentucky University2 hosts a “HyperManyMedia” open-source repository of lectures3. Hundreds of online lectures are available in different formats: text, power-point, audio, video, podcast, vodcast, and RSS. For the last two years, the number of lectures added to this platform has grown significantly, and this led to several problems: (1) searching for a specific college, course name, or lecture, is time consuming, and the search results are not always accurate (i.e. there is a need for personalized, focused search), (2) searching for a topic that is semantically related to the learners’ profiles is lacking (i.e. there is a need for semantic search). Therefore, we implemented a personalized semantic search engine to overcome these limitations.

2.2 Why did we need a Semantic Cluster-based Search Engine?

While the college name which indicates an area of study, seemed to offer a valid categorization basis, we thought that this might be too broad. For example, in the “math” college alone, the topics may vary from history of math to calculus, or to geometry. A course-based categorization alone would also be obtuse, because there may be courses with great overlap. As a result, we decided to use a clustering technique to divide the documents into an optimal categorization that is not influenced by the hand-made taxonomy of the colleges and course titles. This can be expected to provide a finer granularity compared to the coarse college-based categories, and most importantly, promises to offer a more adaptive approach in the face of a future addition of courses or colleges, or in the face of migrating the search approach to other e-learning platforms. In other words, clustering is used to both refine the college-based ontology constructed in Section 4.1, and also as a mechanism to "shake" the rigidity of an otherwise entirely manually constructed ontology, that may not be appropriate for all users and for all times. The most important advantage of clustering from the personalization perspective, is that the clusters are later used as automatically constructed labels for each user profile. Hence, depending on the document collection, and its evolution, both the user profiles, and their underlying ontology labels are allowed to change or evolve accordingly.

3. PROPOSED ARCHITECTURE

Our proposed architecture is divided into three layers as shown in Figure 1: (1) semantic representation (knowledge representation), (2) algorithms (core software), and (3) personalization interface.

4. METHODOLOGY

4.1 Semantic Domain Structure

Let R represents the root of the domain which is represented as a tree, and Ci represents a concept under R. In this case:

$$ R = \bigcup_{i=1}^{n} C_i $$

where n is Number of concepts in the domain. Each concept Ci consists either of sub-concepts SCji, which can be children of Ci, or leaves which are the actual lecture documents ($\bigcup_{k=1}^{m} d_{ki}$), that is

$$ C_i = \begin{cases} \bigcup_{j=1}^{m} SC_{ji} & \text{if } C_i \text{ has subconcepts} \\ \bigcup_{k=1}^{m} d_{ki} & \text{leaves} \end{cases} $$

We encoded the above semantic information into a tree-structured domain ontology in OWL, based on the hierarchy of the e-learning resources. The root concepts are the colleges, while the subconcepts are the courses, and the leaves are the resources of the domain (lectures). Each node (non-leaf) holds the following information: <parent node, concept node, visited node, child node>, while a leaf node holds <parent node, visited node, document, nil>. Figure 2 illustrates part of the tree structure generated from the OWL file.

---

1openlearn.open.ac.uk
2http://www.wku.edu
3http://blog.wku.edu/podcasts
4.2 Building A Learner’s Semantic Profile

We build the semantic learner profiles by extracting the learner interests (encoded as a pruned tree) from the semantic domain (which is the complete tree). Since our log of the user access activity shows the visited documents (which are the leaves), a bottom-up pruning algorithm is used to extract the semantic learner concepts that he/she is interested in.

Each learner \(U_i \subset R\) has a dynamic semantic representation. First, we collect the learner’s activities over a period of time to form an initial learner profile, as follows:

Let \(\text{docs}(U_i) = \bigcup_{k=1}^{l} d_{ki}\) be the visited documents by the \(i^{th}\) learner, \(U_i\). Starting from the leaves, the bottom-up pruning algorithm searches for each document visited by the learner in the “domain semantic structure” illustrated in Figure 2, and then increments the visit count (initialized with 0) of each visited node all the way up to the root. After back-propagating the counts of all the documents in this way in the domain structure, the pruning algorithm keeps only the concepts (colleges) and sub-concepts (courses) related to the learner interests with their weighted interests (which are the number of visits), as in Figure 3. Algorithm 1 shows the bottom-up pruning steps. The output of this algorithm is the learner’s semantic profile. The duration of the time window, during which the visit counts are accumulated for a user, is a parameter that controls the memory span of the profile, so it can range from short to long term.

Algorithm 1 Bottom-up Pruning Algorithm: Building the learner’s Semantic Profile

```
Input: docs(U_i) = \bigcup_{k=1}^{l} d_{ki}; // # of visited documents by user U_i
Output: RU_i = \bigcup_{m=1}^{n} C_i; // User Ontology Tree (learner’s semantic profile)
R = \bigcup_{n=1}^{m} C_i; // Domain Ontology Tree
DomainConcept = root; // DomainOntology Tree

While (CollegeConcept <> nil) do
    If (CollegeConcept.counter = 0)
        remove(CollegeConcept, DomainConcept);
    else
        CourseConcept = CollegeConcept.child;
        UpperConcept = CollegeConcept;
        While (CourseConcept <> nil) do
            If (CourseConcept.counter = 0)
                Remove(CourseConcept, UpperConcept);
            else
                SubConcept = CourseConcept.child;
                ParentConcept = CourseConcept;
                While (SubConcept <> nil) do
                    If (SubConcept.counter = 0)
                        Remove(SubConcept, ParentConcept);
                    end
                    ParentConcept = SubConcept;
                    SubConcept = SubConcept.next;
                end
                UpperConcept = CollegeConcept;
                CourseConcept = CourseConcept.next;
            end
        end
    end
CollegeConcept = CollegeConcept.next;
end

RU_i = DomainConcept;
```

4.3 Cluster-based Semantic Profile

One important aspect of our approach is the combination of an authoritatively supplied taxonomy by the colleges, with the data-driven extraction (via clustering) of a taxonomy from the documents themselves, thus making it easier to adapt to different learning platforms, and making it easier to evolve with the document/lecture collection. Thus, we need to cluster the documents into meaningful groups that form a finer granularity compared to the broader college and course categories provided by the available e-learning taxonomy.

We compared different hierarchical algorithms for a dataset consisting of 2812 documents using the clustering package Cluto\(^4\). We repeatedly applied each clustering algorithm with all possible combinations of clustering criterion functions for different numbers of clusters: 20, 25, 30, 35, 40. By considering each college as one broad class (thus 10 categories), we tried to ensure that the clusters are as pure as possible, i.e. each cluster contains documents mainly from the same category. However, since a class may be parti-

\(^4\)http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview
tioned into several clusters (as was the case here), the clusters are more refined versions of the college categories, which is our desired aim. We used the cluster entropy measure [17] to evaluate the quality of each clustering solution. This measure evaluates the overall quality of a cluster partition based on the distribution of the documents in the clusters, as follows

$$E(S_r) = -\frac{1}{\log q} \sum_{i=1}^{q} \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r},$$

where $q =$ number of classes in the dataset, and $n_r^i$ is the number of documents of the $i^{th}$ class that were assigned to the $r^{th}$ cluster. This measure is calculated for each cluster ($r$). Then the entropy of the entire partition, consisting of $p$ clusters is computed as follows [17]

$$E(T) = \sum_{r=1}^{p} \frac{1}{p} E(S_r).$$

We implemented three different clustering algorithms that are based on the agglomerative, partitional, and graph partitioning paradigms [16]. In agglomerative algorithms, starting from assigning each document to its own cluster, the goal is to find the pair of clusters to be merged at the next step, and this can be done using known approaches, such as single-link, weighted single-link, complete-link, weighted complete link, UPGMA or others, using different criterion functions [16]: $I_1$, $I_2$, $E_1$, $G_1$, $G_1^*$, $H_1$, $H_2$, with each criterion measuring different aspects of intra-cluster similarity and inter-cluster dissimilarity. From our experiments, we found, as shown in Table 1, the best performing criterion to be the $H_2$ (given below), with $u$ and $v$, being documents and $S_r$ being the $r^{th}$ cluster, containing $n_r$ documents, while $\text{sim}(u,v)$ denotes the similarity between $u$ and $v$ [17].

$$H_2 = \sum_{i=1}^{k} \sqrt{\frac{1}{n_r^i} \sum_{u,v \in S_r^i} \text{sim}(u,v)}$$

In partitional clustering algorithms, the goal is to find the clusters by partitioning the set of documents into a predetermined number of disjoint sets, each related to one specific cluster by optimizing various criterion functions [17]. We experimented with two partitional algorithms, direct K-way clustering (similar to K-means), and repeated bisection or Bisecting K-Means, which makes a sequence of bisections (running K-means with $K=2$ clusters) to find the best solution; and experimented with all criterion functions. For direct K-way, $I_2$ [17] performed best, whereas $H_1$ [17] performed best for repeated bisection, as shown in Table 1. $I_2$ and $H_1$ are given below.

$$I_2 = \sum_{i=1}^{k} \left( \sum_{u,v \in S_r^i} \text{sim}(u,v) \right)$$

$$H_1 = \sum_{i=1}^{k} \frac{1}{n_r^i} \sqrt{\sum_{u,v \in S_r^i} \text{sim}(u,v)}$$

We also experimented with graph-partitioning-based clustering algorithms which use a sparse graph to model the affinity relations between different documents, and then discover the desired clusters by partitioning this graph [11] [5]. Of all the algorithms mentioned so far, graph-partitioning produced the best clustering results as shown in Table 1, with 35 clusters and the lowest entropy.

Table 1: Clustering Entropy Measures for various algorithms (rows) and partitioning criteria (columns)

<table>
<thead>
<tr>
<th>Algorithmic Methods</th>
<th>$I_1$</th>
<th>$I_2$</th>
<th>$E_1$</th>
<th>$G_1$</th>
<th>$G_1^*$</th>
<th>$H_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct K-way Methods</td>
<td>0.040</td>
<td>0.025</td>
<td>0.032</td>
<td>0.058</td>
<td>0.036</td>
<td>0.022</td>
</tr>
<tr>
<td>$H_2$ Slink WSLink Clink WCLink UPGMA</td>
<td>0.023</td>
<td>0.493</td>
<td>0.493</td>
<td>0.060</td>
<td>0.060</td>
<td>0.067</td>
</tr>
<tr>
<td>Direct Bisection Methods</td>
<td>0.036</td>
<td>0.020</td>
<td>0.046</td>
<td>0.067</td>
<td>0.055</td>
<td>0.038</td>
</tr>
<tr>
<td>$H_2$ Slink WSLink Clink WCLink UPGMA</td>
<td>0.037</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Repeated Bisection Methods</td>
<td>0.004</td>
<td>0.037</td>
<td>0.038</td>
<td>0.058</td>
<td>0.036</td>
<td>0.027</td>
</tr>
<tr>
<td>Graph Partitional Methods</td>
<td>$pG_1$</td>
<td>$pH_1$</td>
<td>$pH_2$</td>
<td>$pI_1$</td>
<td>$pI_2$</td>
<td></td>
</tr>
<tr>
<td>$p$ Slink WSLink Clink WCLink UPGMA</td>
<td>0.033</td>
<td>0.044</td>
<td>0.042</td>
<td>0.102</td>
<td>0.043</td>
<td></td>
</tr>
<tr>
<td>$H_2$ Slink WSLink Clink WCLink UPGMA</td>
<td>0.032</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Graph partitioning of the entire collection into 35 clusters generated the confusion matrix shown in Table 2, with only 41 misclassified documents out of 2812 (~1%). We relabeled each cluster, based on the majority of assigned documents in each college and from each course, as follows: college-name\course-name, as shown in the last column in Table 2.

4.4 Cluster to Profile Ontology Mapping

Each learner’s profile $\mathcal{U}_l$ is considered as a set $D$ of documents $doc\mathcal{S}(\mathcal{U}_l) = \cup_{l=1}^{L} d_k$. The domain clusters $C_l = \cup_{l=1}^{L} Cl_k$ are obtained from the clustering in section 4.3. The mapping procedure, shown in Algorithm 2, measures the similarity $Sim(D,CL_k)$ between the learner profile documents and each cluster description (frequent terms). The most similar
cluster is considered as a recommended cluster. The recommended cluster has two effects on our searching mechanism: first, on the re-ranking algorithm, and second, on the learner’s semantic term recommendation.

### 4.5 Changing the Learner’s Semantic Profile

After extracting the most similar cluster \( C_i = \text{BestCluster} \) (recommended-cluster), which is summarized by the \( \text{Topn} \) keywords (significant or frequent terms), we modified the learner’s semantic ontology (in the OWL description) accordingly, by adding the cluster’s terms as semantic terms under the concepts (parent nodes) that these documents belong to.

### 4.6 Re-ranking the Learner’s Search Results

We start by representing each of the \( N \) documents as a term vector \( \vec{d} = < w_1, w_2, ..., w_n > \), where \( w_i = t_i \cdot \log \frac{N}{n_i} \) is the term weight for term \( i \), combining the term frequency, \( t_i \), and the term’s Inverse Document Frequency \( (\text{IDF}_i) = \log \frac{N}{n_i} \), given that this term occurs in \( n_i \) documents. When a learner searches for lectures using a specific query \( q \), the cosine similarity measure is used to retrieve the most similar documents that contain the terms in the query. In our approach, these results have been re-ranked based on two main factors: (1) the semantic relation between these documents and the learner’s semantic profile, and (2) the related similar documents, i.e., the most similar documents to the recommended cluster (Algorithm 1), where each document \( d_i \), belonging to a learner’s semantic profile, is assigned a priority ranking \( (\alpha = 5.0) \), and each document \( d_i \), belonging to the recommended cluster (Category 2) is assigned a priority ranking \( (\beta = 3.0) \), while the rest of the documents (Category 3) be the lowest priority \( (\gamma = 1.0) \). All the documents, in each category, are then re-ranked based on their cosine similarity to the query \( q \). Our search engine (based on nutch) uses optional boosting scores to determine the importance of each term in an indexed document, when adding up the document-to-query term matches in the cosine similarity. Thus, a higher boosting factor for a term will force a larger contribution from that term in the sum. More details about this boosting algorithm is in 5. We modified the boosting score as follows: \( \text{doc.setBoost}(d_i) = \alpha \), in case of Category 1, \( \text{doc.setBoost}(d_i) = \beta \), in case of Category 2, and \( \text{doc.setBoost}(d_i) = \gamma \), in case of Category 3.

---

### Algorithm 2 Best Cluster Mapping algorithm for a learner \( U \)

**Input:** \( D \) \( \forall k = 1, n \) if \( k \neq \text{visited docs} \)

**Output:** \( \text{BestCluster} \) \( \forall k = 1, n \) \( \text{most similar cluster} \)

\( \text{BestCluster} = \bigcup_{k=1}^{n} \text{CL}_k \); \( n = \# \text{of clusters} \)

\( \text{for each} \ C_{L_1} \in \text{CL} \) if \( \text{Sim}(D, C_{L_1}) > \text{Sim}(D, \text{BestCluster}) \) then \( \text{BestCluster} = C_{L_1} \)

end

### Algorithm 3 Re-ranking a learner’s search results

**Input:** \( q \) \( // \) keyword search

**Output:** \( \text{Rank} = \{d_1, d_2, ..., d_n\}; // \text{RANK} \)

\( \text{Rank} = \{d_1, d_2, ..., d_n\}; // \text{RANK} \)

\( \text{default search results for query} q \)

\( \text{UR}_k = \bigcup_{k=1}^{n} \text{CL}_k \)

\( \text{RC} = \bigcup_{k=1}^{n} \text{d}_k ; // \text{d}_k \text{documents in Recommended Cluster} \)

**for each** \( d_j \in \text{Rank} \)

if \( d_j \notin \text{UR}_k \) then

\( d_j \cdot \text{boost} = \alpha; // \text{document is in user profile} \)

end

else

if \( d_j \notin \text{RC} \) then

\( d_j \cdot \text{boost} = \beta; // \text{document is in recommended cluster} \)

end

else

\( d_j \cdot \text{boost} = \gamma; \)

end

Sort Rank based on boost field \( d_j \cdot \text{boost} \)

---

4.7 Semantic Term Recommendation

For each query \( q \) submitted by a learner, a semantic mapping between the query and the learner’s semantic profile brings all the concepts/subconcepts/cluster-based-recommended-terms (added in Section4.5.) This framework allows the learner to navigate through the semantic structure of his/her query, as shown in Figure 4. by possibly clicking on one of the recommended terms. The effect of this action is to add the selected term to the query and repeat the search. Therefore the search is finally personalized via a query expansion using the recommended term that is selected.

Figure 4: Semantic terms recommendation

5. IMPLEMENTATION

5.1 Representing the Semantic Domain

As of November 2006, Western Kentucky University\(^7\) hosted a “HyperManyMedia”\(^6\) open-source repository of lectures. These resources (lectures) are available in different formats: text, power-point, audio, video, podcast, vodcast, and rss. Designing the domain semantics from this platform is based on the hierarchical structure of these resources. The “HyperManyMedia” platform contains eleven different concepts (colleges): “English”, “Social Work”, “History”, “Chemistry”, “Accounting”, “Math”, “Consumer and Family Sciences”, “Architecture and Manufacturing Sciences”, “Engineering”, and “Communication Disorders”. Each concept (college) contains a different number of sub-concepts (courses), and under each sub-concept (course), the learning object represents a leaf of this tree.

5.2 Constructing the Dataset

Our methodology for this phase consists of (1) crawling and downloading all the webpages (lectures) from the e-learning platform, (2) extracting the text (documents) from the webpages, (3) pre-processing the documents (removing stop words and high frequency words, and applying Porter’s stemming algorithm), and (4) applying text mining algorithms (document clustering). Our corpus consists of a total of 2,812 documents indexed under different concepts/subconcepts. We used these documents to mine the clusters, as explained in Section 4.3. We constructed an RDF-based (OWL) ontology for our entire e-learning domain based on the hierarchical structure explained in Section 5.1. Ten learners were selected, with each learner representing a college. Each learner profile was constructed based on the learner’s logs (navigated lectures) during a time window spanning two semesters. These lectures represent the learner’s interests in the e-learning resources, as described in Section 4.2. We finally constructed keyword queries related to each learner’s profile using terms from the subconcepts (course name) and lecture names, and a combination of the most significant terms under each concept. Three datasets of queries have thus been used (single-keyword, two keywords, three keywords).

6. EXPERIMENTAL EVALUATION

We used Top-n-Recall and Top-n-Precision to measure the effectiveness of re-ranking based on the learner’s semantic profile, using as a training set, the whole e-learning domain, i.e. 10 concepts (colleges), 28 subconcepts (courses), and a total of 2,812 lectures (documents) that were indexed under various concepts/subconcepts. After constructing the domain ontology, we selected 10 learner profiles, as explained in Section 5.2. and built the semantic profile for each learner using Algorithm 1, from Section 4.2. A total of 1,406 lectures (documents) represented the profiles, with the size of each profile varying from one learner to another, as follows. Learner1 (English)= 86 lectures, Learner2 (Consumer and Family Sciences) = 74 lectures, Learner3 (Communication Disorders) = 160 lectures, Learner4 (Engineering) = 210 lectures, Learner5 (Architecture and Manufacturing Sciences) = 119 lectures, Learner6 (Math) = 374 lectures, Learner7 (Social Work) = 86 lectures, Learner8 (Chemistry) = 58 lectures, Learner9 (Accounting) = 107 lectures, and Learner10 (History) = 132 lectures. We finally used our semantic search engine \(^9\) to evaluate each query, and computed the Top-n-Precision and Top-n-Recall for normal search and for personalized semantic search for each learner. Our evaluation results, shown in Figure 5, show the Average Percentage of Improvement in Top-n Recall and Top-n Precision for the personalized Search over the normal search, with three sizes of queries (1, 2, and 3 keywords). The personalized semantic search shows an improvement in precision that varies between 5-25 %. This improvement is noticeable between the top-30 and top-50 results for single-keyword and two-keywords queries. The recall results show a noticeable improvement in recall between the top-20 and top-40 results. Overall, these results show the effectiveness of the re-ranking based on the learner’s semantic profile.

7. CONCLUSION AND FUTURE WORK

We have shown that extracting the semantic interests of learner profiles can form a reasonable and simple way to represent the learning context, and that this semantic learner profile, coupled with a semantic domain ontology that represents the learned content, can enhance the retrieval results on a real e-learning platform. In our future work, we will investigate different clustering methods, in addition to the effect of the semantic profile time window parameter more thoroughly. We will possibly use a mixed approach with both a short term and long term profile at the same time, to be used under different conditions/users. A continuous

\(^6\)http://www.wku.edu

\(^7\)HyperManyMedia: refers to any educational material on the web (hyper) in a format that could be a multimedia format (image, audio, video, podcast, vodcast) or a text format (webpage, powerpoint).

\(^8\)http://blog.wku.edu/podcasts

\(^9\)http://blog.wku.edu/podcasts
forgetting factor to gradually forget older user activity is another interesting and adaptive approach.

8. ACKNOWLEDGMENTS

This work is partially supported by National Science Foundation CAREER Award IIS-0133948 to Olfa Nasraoui. We are also grateful to our evaluation team (Kunal Gosar, Satish Bhavanasi, Sudhir Jammalamadaka).

9. REFERENCES


