Dual Representation of the Semantic User Profile for Personalized Web Search in an Evolving Domain

Leyla Zhuhadar and **Olfa Nasraoui** *
Knowledge Discovery and Web Mining Lab
Department of Computer Engineering
and Computer Science
University of Louisville, USA

Robert Wyatt
Department of Distance Learning
Western Kentucky University, USA

Abstract

Several semantic based user profile approaches have been introduced in the literature to learn the users' interests for personalized search. However, many of them are ill-suited to cope with a domain of information that evolves and user interests that may change over time. In this paper, we propose a novel dual representation of a user's semantic profile to deal with this problem: (1) a lower-level semantic representation, consisting of an accumulated gathering of user activities over a long period of time, that uses a standard machine learning algorithm to detect user convergence, (2) a higher-level semantic representation that detects shifts in the user activities- once this shift is detected, the higher-level semantic representation automatically updates the user profiles and reinitialize the system. Our experimental results demonstrate the feasibility of this approach.

Introduction

Information filtering (IF) is concerned with the problem of delivering information that is relevant to a user's interests. Typically, the relevance of information is related to the user's preferences, which is commonly referred to as the user profile (Belkin and Croft 1992). Our research deals with the user profile-based category of information filtering systems. User profiling also known as user modeling is a very active field of research in information retrieval. It focuses on abstracting the user away from the problem and creating user-generic approaches that are "great" for anyone and not only "good enough" for everyone (Allan et al. 2003).

Our methodology of constructing user profiles is based on observing the user's activities over the long-term. These activities represent the individual interests. The objective is to estimate the user profiles from an ongoing tracking of activities so that the filtering system can effectively present the information that is as relevant to the user's interests as possible. Hypothetically, the performance of such a system should improve with time since the more interaction happening with the user, the more the system learns about this specific user. This hypothesis holds as long as the user's interests do not change with time.

*This work is supported by National Science Foundation CAREER Award IIS-0133948 to Olfa Nasraoui.
Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

However, since we are dealing with an evolving domain, where both content and users evolve, this hypothesis may fail. Therefore, a new methodology must be considered to handle this challenge. We discovered that the changes of interests may occur for two reasons in our domain: (i) if the user changed his/her domain of interest, e.g. if changes happened in the user's personal or professional status, (ii) if the content of the domain has changed, e.g. adding or removing colleges, courses, lectures, etc. Undetected user shifts of interest may cause a significant decrease in the performance of an information filtering system. In this paper, we propose a dual representation approach that can deal with such a situation.

Related Work

Recently, several new efforts in the area of the semantic information retrieval have also been well presented, such as (Pretschner and others 1999; Gauch 2003; Sheth et al. 2002; Sieg, Mobasher, and Burke 2007). Of all related work, the latter seems to be the closest to ours. However, there are major differences between our approach and theirs: **(1)** our study focuses on an *evolving user* in an *evolving domain*, and **(2)** our search engine provides re-ranking based not only on the user's profile, but also on cluster-based similarity metrics that capture the distribution of the documents in the domain. Despite many improvement in IF systems, the majority of existing systems do not adequately address the problem of evolving domains and evolving user profiles. Of the few that have addressed the evolution of user interests, we cite (Mitchell et al. 1994)'s Personal Assistant that trained decision trees to learn how to schedule an individual's meetings in a personalized calendar. A time window (consisting of the last 180 examples) was used to confine the training samples for learning, and to adapt to the changing user's scheduling preferences. The newly generated rules are merged with old ones, with poorly performing rules ranking lower on the list. Another system is NewsDude (D. and J. 1999 pp 99 108), an intelligent agent built to adapt to changing users' interests by learning two separate user models that represent short term and long term interests. The short term model is learned from the most recent observations only, while the long term model represents the user's general preferences. If the short term model cannot classify the story, it is passed on to the long term model. In (Barry Crabtree and Soltysiak

1998), a user profiling system was developed based on monitoring the user's web browsing and e-mail habits. This system used a clustering algorithm to group user interests into several interest themes, and the user profiles had to adapt to changing interests of the users over time. (Koychev and Schwab 2000) used gradual forgetting to deal with drifting interests. A time decreasing weight was assigned to examples that were used in training a decision tree, thus giving more importance to more recent observations while learning. More recently, (Nasraoui et al. 2008) presented a semantic Web usage mining methodology for mining evolving user profiles on dynamic websites by clustering the user sessions in each period and relating the user profiles of one period with those discovered in previous periods. All the above efforts addressed the evolution of user interests, however they did not implement their methods within '

Updated Architecture

Our previous architecture (Zhuhadar and Nasraoui) was composed of three layers which are shown as the upper three layers in **Figure 1**: (1) semantic representation (knowledge representation), (2) algorithms (core software) , and (3) personalization interface. In this paper, we add layer (4) "updating the user's semantic profile". The main purpose of this layer is to detect shifts in the user's interests, which is the main subject of this paper.

Semantic Domain Structure

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

$$\mathrm{R} = \cup_{i=1}^{n} C_i, \tag{1}$$

where n is the number of concepts in the domain. Each concept C_i consists of either sub-concepts which can be children, (SC_{ji}), or leaves which are the actual lecture documents ($\cup_{k=1}^{l} d_{ki}$), i.e.,

$$C_i = \begin{cases} C_i = \cup_{j=1}^{m} SC_{ji} & if\ C_i\ has\ subconcepts \\ \\ \cup_{k=1}^{l} d_{ki} & leaves \end{cases} \tag{2}$$

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>***.

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 $docs(U_i) = \cup_{k=1}^{l} d_{ki}$ 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's semantic structure", 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). **Algorithm 1** shows the bottom-up pruning steps.

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

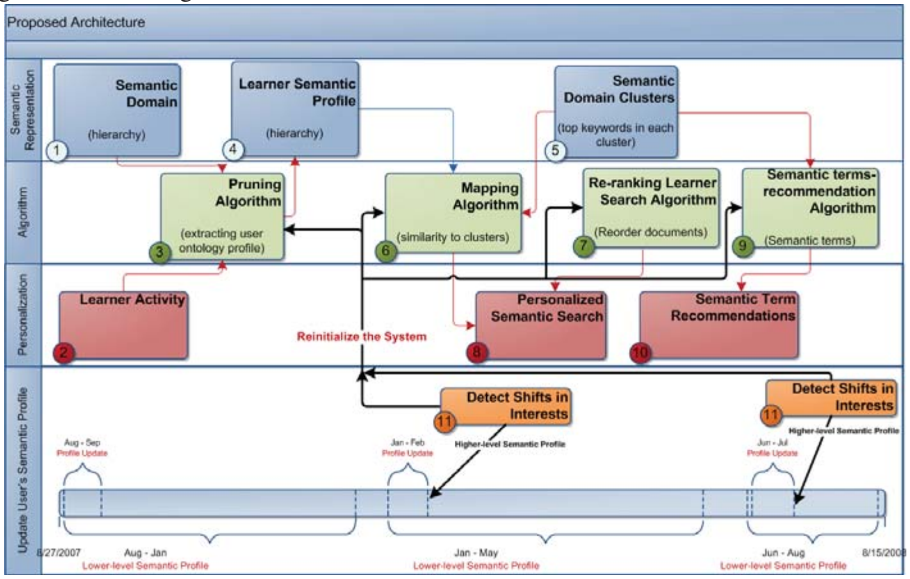
```
Input: docs(U_i) = ∪_{k=1}^{l} d_{ki}; //l = #of visited documents by user U_i
Output: RU_i = ∪_{i=1}^{m} C_i; //User Ontology Tree(learner's semantic profile)
R = ∪_{i=1}^{n} C_i; //Domain Ontology Tree
DomainConcept = root;
CollegeConcept = root.child;
While (CollegeConcept <> nil) do
    If (CollegeConcept.counter = 0)
        remove(CollegeConcept, DomainConcept);
    end
    else
        CourseConcept = CollegeConcept.child;
        UpperConcept = CollegeConcept;
        While(CourseConcept <> nil) do
            If (CourseConcept.counter = 0)
                Remove(CourseConcept, UpperConcept);
            End
            Else
                SubConcept = CourseConcept.child;
                ParentConcept = CourseConcept;
                While(SubConcept <> nil) do
                    If (SubConcept.counter = 0)
                        Remove(SubConcept, ParentConcept);
                    End
                    ParentConcept = SubConcept;
                    SubConcept = SubConcept.next;
                End
            End
            UpperConcept = CollegeConcept;
            CourseConcept = CourseConcept.next;
        End
    End
    DomainConcept = CollegeConcept;
    CollegeConcept = CollegeConcept.next;
End
RU_i = DomainConcept;
```

Cluster-based Semantic Profiles

We compared several hierarchical clustering algorithms for a dataset consisting of 2812 documents using the clustering package Cluto [1]. We ran each clustering algorithm with all possible combinations of clustering criterion functions and 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. We used the *entropy measure* (Zhao and Karypis 2002) 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 (Zhao and Karypis 2002). We implemented three different clustering algorithms that are based on the agglomerative, partitional, and graph partitioning paradigms (Zhao and Karypis 2002). In agglomerative algorithms, starting

[1] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

Figure 1: E-learning Personalization Framework with Detection of Shifts in User Interests



from assigning each document to its own cluster, the goal is to find the pairs 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, that use different criterion functions (Zhao and Karypis 2002): I1, I2, E1, G1, G1*, H1, H2, with each criterion typically 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 H2 criterion, with u and v, being documents and S_i being the i^{th} cluster, containing n_i documents, and $sim(u,v)$ denoting the similarity between u and v (Zhao and Karypis 2002). 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 (Zhao and Karypis 2002). We experimented with two methods of partitional algorithms, direct K-way clustering (similar to K-means), and repeated bisection or Bisecting K-Means (makes a sequence of bisection to find the best solution), and experimented with all criterion functions. For direct K-way, I2 (Zhao and Karypis 2002) performed best, whereas H1 (Zhao and Karypis 2002) performed the best for repeated bisection, as shown in Table 1. We also experimented with graph-partitioning-based clustering algorithms which use a sparse graph to model the affinity relations between different documents, then discover the desired clusters by partitioning this graph (Karypis et al. 1999). 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. Graph partitioning of the entire collection into 35 clusters generated a confusion matrix 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.

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

Agglomerative Methods					
$I1$	$I2$	$E1$	$G1$	$G1*$	$H1$
0.040	0.025	0.039	0.102	0.043	0.024
$H2$	$Slink$	$WSLink$	$Clink$	$WCLink$	$UPGMA$
0.023	0.493	0.493	0.060	0.060	0.067
Direct k-way Methods					
$I1$	$I2$	$E1$	$G1$	$G1*$	$H1$
0.036	0.020	0.040	0.067	0.055	0.038
$H2$	$Slink$	$WSLink$	$Clink$	$WCLink$	$UPGMA$
0.037	-	-	-	-	-
Repeated Bisection Methods					
$L1$	$L2$	$E1$	$G1$	$G1*$	$H1$
0.027	0.034	0.036	0.058	0.036	0.022
Graph Partitional Methods					
pe	pG_1	$pH1$	$pH2$	$pI1$	$pI2$
0.033	0.051	0.042	0.01	0.32	0.017
$H2$	$Slink$	$WSLink$	$Clink$	$WCLink$	$UPGMA$
0.032	-	-	-	-	-

Cluster to Profile Ontology Mapping

Each learner's profile U_i is considered as a set D of documents $docs(U_i) = \cup_{k=1}^{l} d_{ki}$. The domain clusters $CL = \cup_{k=1}^{n} CL_k$ are obtained from the clustering in *section: cluster based Semantic Profiles* . The mapping procedure, shown in Algorithm 2, measures the similarity $Sim(D,CL_i)$ between the learner profile documents and each cluster description (frequent terms). The most similar cluster is considered as a recommended cluster.

Algorithm 2 Best Cluster Mapping algorithm for a learner U_i

Input: $D = \cup_{k=1}^{l} d_{ki}; // l = \#of\ visited\ docs$
Output: $BestCluster\ ; //\quad most\ similar\ cluster$
$CL = \cup_{k=1}^{n} CL_k; //\quad n = \#of\ clusters$
$BestCluster = CL_1$
foreach $CL_i \in CL$
if $Sim(D, CL_i) > Sim(D, BestCluster)$ then
$BestCluster = CL_i$
End
End

Changing the Learner's Semantic Profile

After extracting the most similar cluster $C_i = BestCluster$ (recommended-cluster), which is summarized by the Top_n 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.

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 $w_i = tf_i * log\frac{N}{n_i}$ is the term weight for term (i), combining the term frequency, tf_i, and the term's Inverse Document Frequency ($IDF_i = log\frac{N}{n_i}$), 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 most similar cluster to the learner's semantic profile (recommended cluster). Algorithm 3 maps the ranked documents to the learner's semantic profile (Category 1), where each document d_i, belonging to a learner's semantic profile, is assigned a highest priority ranking ($\alpha = 5.0$), and each document d_i belonging to the recommended cluster (Category 2) is assigned an intermediate priority ranking ($\beta = 3.0$), while the rest of the documents (Category 3) receive 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 [2]. We modified the boosting score as follows: $doc.setBoost() = \alpha$, in case of Category 1, $doc.setBoost() = \beta$, in case of Category 2, and $doc.setBoost() = \gamma$, in case of Category 3.

[2] http://hudson.zones.apache.org/hudson/job/Lucene-trunk/javadoc//org/apache/lucene/search/Similarity.html

Algorithm 3 Re-ranking a learner's search results

Input: $q; // keyword\ search$
Output: $Rank = \{d_1, d_2, ..., d_n\}; // Re-rank$
$Rank = \{d_1, d_2, ..., d_n\}; // default\ search\ results\ for\ query\ q$
$UR_i = \cup_{j=1}^{n} SC_{ji} + \cup_{k=1}^{l} d_{ki}$
$RC = \cup_{c=1}^{z} d_c; // l = \#of\ documents\ in\ Recommended\ Cluster$
foreach $d_j \in Rank$
 if $d_j \in UR_i$ then
  $d_j.boost = \alpha; // document\ is\ in\ user\ profile$
 End
 else
 if $d_j \in RC$ then
  $d_j.boost = \beta; // document\ is\ in\ recommended\ cluster$
 End
 else
 $d_j.boost = \gamma;$
 End
End

Sort Rank based on boost field $d_j.boost$

Semantic Term Recommendation

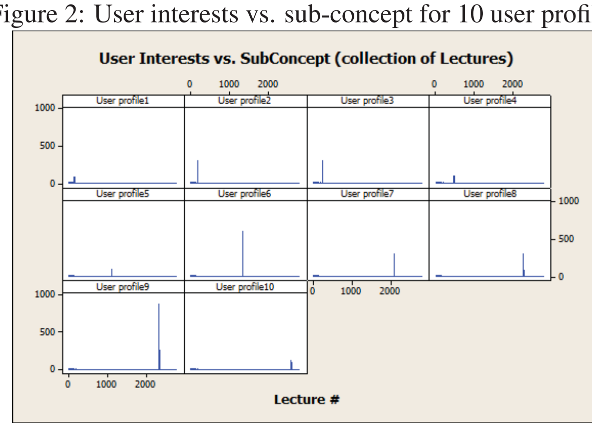
For each query q submitted by a learner, a semantic mapping between the query and the learner's semantic profile brings all the concepts/subconcepts/cluster-based-recommended-terms (added in *section: Changing the Learner's Semantic Profile*). This framework allows the learner to navigate through the semantic structure of his/her query. 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.

Dual Semantic Profile representation

This section presents layer (4) in our updated architecture, depicted in **Figure 1**.

Building the Lower-level of a Learner's Semantic Profile

Figure 2: User interests vs. sub-concept for 10 user profiles



Generating user profiles can be done by tracking the users' activities over a period of time. These activities describe the concepts/subconcepts that the users have visited/re-visited. The main goal of this section is to find the timeframe where the user's interests converge. Our assumption that a user's interests must converge after a period of time is based on related studies, such as (Gauch 2003; Gauch, Chaffee, and Pretschner 2003; Pretschner and others 1999;

Trajkova and Gauch 2004). These studies assumed that each person has a relatively stable collection of interests which converges over time. From an intuitive point view, in an e-learning environment, this assumption holds since the main concern of a user (student) is to retrieve the most relevant information to his/her domain of interest after adapting to the system.

Conceptually, the convergence represents the time at which the rate of increase in the interests for concepts/subconcepts stabilizes. To discover this timeframe, we selected 10 profiles from different colleges. Each profile is generated using Algorithm 1. Figure 2 shows the users' activities over a period of one year. Ordering these lectures was based on concepts, as follows: 1-173 (*English*), 174-222 (*Consumer and Family Sciences*), 223-442 (*Communication Disorders*), 443-1062 (*Engineering*), 1063-1300 (*Architecture and Manufacturing Sciences*), 1301-2048 (*Math*), 2049-2221 (*Social Work*), 2222-2336 (*Chemistry*), 2337-2550 (*Accounting*), 2551-2812 (*History*). As we can notice, the user's activities are concentrated in a window frame related to the concept he/she is interested in. Moreover, the randomness of user activities appears to stabilize after a period of time. We modified Algorithm 1, in this stage, as shown in Algorithm 4 to record a vector $\mathrm{VU_i} = \cup_{i=1}^{m} SC_i$ for each learner of a dimension equals to the number of subconcepts under the *CollgeConcept* he/she is most interested in, in addition to the weight of each subconcept (visit score). This vector represents the concept (college) that the learner is interested in and their subconcepts (courses/lectures). Based on our experimental results, the users' interests converge$\mathrm{VU_i} = \cup_{i=1}^{m} SC_i$ after a period of one month, as shown in Figure 3.

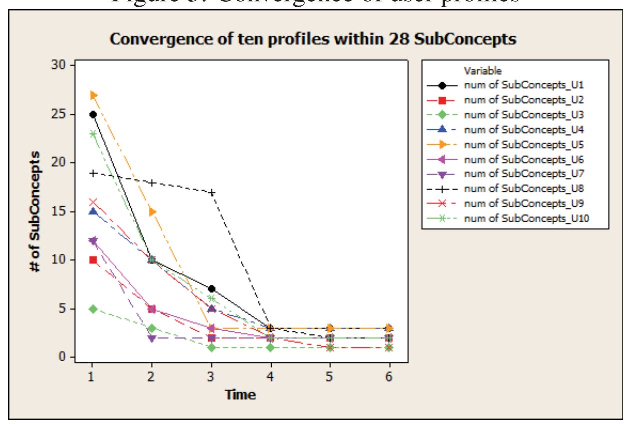
Algorithm 4 Tracking the user's history of interests

Input: $docs(U_i) = \cup_{k=1}^{l} d_{ki}; //l = \#of\,visited\,documents\,by\,user\,U_i$
Output: $\mathrm{VU_i} = \cup_{i=1}^{m} SC_i; //User\,Concept\,interests$
*Call Algorithm*1;
DomainConcept = root;
CollegeConcept = root.child;
While (*CollegeConcept* <> *nil*) do
  If (*CollegeConcept.counter* <> 0) then
    FindTopInterest(*CollegeConcept*); //Find user's CollegeConcept
  End
End
While (*CollegeConcept.child* <> *nil*) do
  If (*CourseConcept.count* <> *o*) then
    AddSubConceptToVectorWithWeights;
  End
End
$\mathrm{VU_i} = \cup_{i=1}^{m} SC_i; //User\,interests$
End

Semantic User Profile Evaluation: Convergence

The evaluation of the semantic user profiles demonstrates the notion of convergence. This part of our study shows that all the lower-level semantic profiles converge after a short period of time. The *time* in Figure 3 refers to the first six weeks of user activities in a semester, we notice that after week 4, the user's interests in subconcepts (courses) started to converge. However, choosing a different hierarchy level for examining the convergence may produce different results. If we analyze the surfing behavior of the users based on a higher level, such as the college level, we would have a convergence from the first week. On the one hand, we decided to have a more accurate user profile that allows us to build a user model with a deeper granularity. On the other hand, as we are going to explain in *section: Building the Higher-level of a Learner's Semantic Profile* , that user may change his/her domain of interest and a window frame of 1 week may not be enough to detect this shift. However, in our future work we will compare different granularity levels of examining the convergence. Based on the convergence results which were completely related to our e-learning domain, we decided to update our learner's profile with a window frame of 1 month, as shown in Figure 3.

Figure 3: Convergence of user profiles



Building the Higher-level of a Learner's Semantic Profile

The main purpose of this experiment is to extract the user's weighted interests in lectures in order to detect the user's shift in interests. We may assume that each learner has a relatively stable collection of interests that might change over time (Lam et al. 1996). This change might occur in the same domain of interest (SubConcept= courses) which will not affect the recommendations provided by the system heavily. Recall that the recommendations come from two sources: clusters recommendations, *section: Cluster to Profile Ontology Mapping* and term recommendations, *section: Semantic Term Recommendation* . But, if the user (learner) completely shifted his/her interests from concept to concept, a system with a long history of accumulated interests (*section: Semantic User Profile Evaluation*) will not notice these changes and the recommender system will keep providing the user (learner) with recommendations related to his/her past activities. To detect the changes in the learner's interests, we keep track of the user's main domain of interest (after his/her activities stabilizes), which is the *concept (college)* and all *subConcepts (courses)* that the user has visited from the previous semester, as described in Algorithm 4. Algorithm 5 detects shifts in the user interests. If this shift happens, it provide the system with immediate feedback to reinitialize the whole system for this specific user. The shift of interests affects two parts of the system: (1) the cluster to profile ontology mapping, and (2) changing the learner's semantic profile. Figure 1 shows the changes in our platform's architecture after detecting the shift in interests. The system will consider the user (learner) as a new learner and his/her new history will be based only on the last 4 weeks of activities. His/her all past activities will be ignored. This decision was made based on the nature of the e-learning domain that considers dealing with changes in the user interests that may have occurred due to changes in the learner's personal or professional situation.

Algorithm 5 Detecting shifts in the user's interests

Input: $\mathrm{VOU_i} = \cup_{i=1}^{m} SC_i$; $//User\ Concept\ interest$(learner's concept interest last semester)
Input: $\mathrm{VNU_j} = \cup_{j=1}^{n} SC_j$; $//User\ Concept\ interest$(learner's concept interest in a new semester)
output: $\mathrm{VNU_j} = \cup_{j=1}^{n} SC_j$; $//User\ Concept\ interest$(learner's concept interest updates)
If $\mathrm{VOU_i}.CollegeConcept <> \mathrm{VNU_i}.CollegeConcept$ then
    Call Agorithm1;//Building the leaner's Semantic Profile
    Call Agorithm2; //Best cluster Mapping for the user
    Call Agorithm3;//Re-ranking a learner's search results
End
End

Conclusion and Future Work

Most available semantic user profiles in the context of personalized web search are designed for stationary users. In this paper, we proposed a novel dual representation of a user's semantic profile to deal with this problem based on (1) a lower-level semantic representation that consists of the accumulated gathering of the user's activities, and (2) a higher-level semantic representation algorithm that detects shifts in the user's interests. We have shown that extracting the semantic interests in the user profiles can form a reasonable way to represent the learning context, and that the semantic profile, coupled with a semantic domain ontology which represents the learned content, can enhance the retrieval results on a real e-learning platform. We demonstrated the notion of convergence in an information filtering system, and we discovered that all the lower-level semantic profiles converged after a short period of time. Finally, we dealt with the problem of evolving user profiles and an evolving domain. In case the system detects changes in the user's interests, our method re-initializes the user profiles based on the new interests. In our future work, we will investigate different granularity levels in examining the convergence. In addition, we plan to study mixed techniques for detecting shifts in interests that can provide a more accurate prediction and a faster response.

References

Allan, J.; Aslam, J.; Belkin, N.; Buckley, C.; Callan, J.; Croft, B.; Dumais, S.; Fuhr, N.; Harman, D.; Harper, D.; et al. 2003. Challenges in Information Retrieval and Language Modeling. In *ACM SIGIR Forum*, volume 37, 31–47.

Barry Crabtree, I., and Soltysiak, S. 1998. Identifying and tracking changing interests. *International Journal on Digital Libraries* 2(1):38–53.

Belkin, N., and Croft, W. 1992. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM* 35(12):29–38.

D., B., and J., P. M. 1999, pp. 99-108. A hybrid user model for news classification. In *Proceedings of the Seventh International Conference on User Modeling Springer-Verlag*.

Gauch, S.; Chaffee, J.; and Pretschner, A. 2003. Ontology-Based User Profiles for Search and Browsing. *J. User Modeling and User-Adapted Interaction: The Journal of Personalization Research, Special Issue on User Modeling for Web and Hypermedia Information Retrieval*.

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

Karypis, G.; Aggarwal, R.; Kumar, V.; and Shekhar, S. 1999. Multilevel hypergraph partitioning: applications in VLSI domain. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 7(1):69–79.

Koychev, I., and Schwab, I. 2000. Adaptation to Drifting User's Interests. In *Proc. of ECML2000 Workshop: Machine Learning in New Information Age*, 39–45.

Lam, W.; Mukhopadhyay, S.; Mostafa, J.; and Palakal, M. 1996. Detection of shifts in user interests for personalized information filtering. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 317–325. ACM New York, NY, USA.

Mitchell, T.; Caruana, R.; Freitag, D.; McDermott, J.; and Zabowski, D. 1994. Experience with a learning personal assistant. *Communications of the ACM* 37(7):80–91.

Nasraoui, O.; Soliman, M.; Saka, E.; Badia, A.; and Germain, R. 2008. A Web Usage Mining Framework for Mining Evolving User Profiles in Dynamic Web Sites. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING* 202–215.

Pretschner, A., et al. 1999. Ontology based personalized search. Master's thesis, University of Kansas, Electrical Engineering and Computer Science.

Sheth, A.; Bertram, C.; Avant, D.; Hammond, B.; Kochut, K.; and Warke, Y. 2002. Managing semantic content for the Web. *Internet Computing, IEEE* 6(4):80–87.

Sieg, A.; Mobasher, B.; and Burke, R. 2007. Ontological user profiles for representing context in web search. *Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences on* 91–94.

Trajkova, J., and Gauch, S. 2004. Improving ontology-based user profiles. In *Proceedings of RIAO*, volume 4, 380–389.

Zhao, Y., and Karypis, G. 2002. Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, 515–524. New York, NY, USA: ACM.

Zhuhadar, L., and Nasraoui, O. Personalized cluster-based semantically enriched web search for e-learning. In *CIKM 08: ONISW The 2nd International workshop on Ontologies and Information Systems for the Semantic Web.*