Automated Discovery, Categorization and Retrieval of Personalized Semantically Enriched E-learning Resources

Leyla Zhuhadar
Knowledge Discovery and Web Mining Lab
Department of Computer Engineering and Computer Science
University of Louisville, Louisville, KY 40292, USA

Olfa Nasraoui
Knowledge Discovery and Web Mining Lab
Department of Computer Engineering and Computer Science
University of Louisville, Louisville, KY 40292, USA

Robert Wyatt
The Office of Distance Learning
Division of Extended Learning and Outreach
Western Kentucky University, Bowling green, KY 42101, USA

Abstract—As the number of online students has been growing significantly for the last couple of years, generic Web Information Retrieval methods have either maintained an emphasis on serving the general population, or have been lagging in integrating the power of adaptive semantics and personalization via knowledge discovery into real-life working E-learning applications. In this paper, we describe an integrated and working E-learning search system for retrieving personalized semantically enriched learning resources. The first part of this paper is related to Information Filtering, which despite its improvements for the past decade, has focused on models that are “good for all users” 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. Thus the need for an infrastructure that can provide, manage, and collect data that permits high levels of adaptability and relevance to the learner’s profiles. Within this context, this work proposes an architecture divided into four layers: (1) Semantic Representation (knowledge representation), (2) Algorithms, which are the core engine of this study, (3) Personalization Interface to deal with information filtering, and (4) Dual representation of the semantic user profile. This architecture is implemented and evaluated on a real E-learning platform named “HyperManyMedia”. We use Cluster Analysis in support of an adaptive personalized search for E-learning, where Cluster Analysis is used to divide the documents into an optimal categorization that is not influenced by the hand-made taxonomy of the colleges and course titles. In other words, clustering is used to both refine the college-based ontology 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 second part of this paper proposes a dual representation of the semantic user profile: (1) a lower-level semantic representation that consists of the accumulated gathering of a user’s interests over a long time window-frame that uses a standard convergence type machine learning algorithm, (2) a higher-level semantic representation algorithm that detects shifts in user’s activities. This work ends with an experimental evaluation of the results and an overview of future research. Evidence is found that both personalization and semantic enrichment are potential elements for improving an E-learning Information Retrieval System.

I. Introduction

A. Information Filtering and User Profiles

“The personalization aspect in IR is already an active field of research. However, most of the current research only focuses on investigating the behaviors of users in respect to the relevance of information objects where there is strong evidence that other factors can affect the relevance, such as factors associated with goals, tasks, the individual knowledge, and a variety of other contextual features [1]”. We can summarize Information Filtering (IF) as Information Retrieval that 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 [2]. Information Filtering deals with user profiles which represent long term interests in a specific domain. Various features and characteristics govern IF systems. [2] differentiates between IR and IF as follows:

- IF considers a user as a long-term user. Therefore, the system accumulates user activities and creates his/her profile. Whereas, Information Retrieval considers a user as a single-time user. The system is only interested in providing relative information to the user query for a specific single task [2].

- IF makes an assumption that a user profile is accurate, and, therefore, the information presented to a user matches his/her interests. On the other hand, Information Retrieval knows that finding the needed information that matches the requested query is a difficult task and that representing the requested information as a query is not sufficient to find accurate results [2].

- IF’s main concern is how to disseminate information to individuals or groups, while Information Retrieval’s main concern is how to organize and collect information (indexing, representation) [2].

- IF deals with the problem of filtering/deleting unneeded information, while Information Retrieval works on finding relevant information from the data [2].

[3] made a comparative study of some of the existing work in information filtering systems. The following list is not exhaustive, but provides a good summary:
- A user-model based IF, such as Cite-Seer [4], uses the personal profile to recommend only relevant documents to the user’s research interests by downloading only relevant documents from digital libraries. Cite-Seer uses statistical relatedness measures, based on constraint-based relatedness [3].

- A rule-based IF, such as GHOSTS [5], filters e-mails according to a set of user-defined rules. GHOST requires explicit user input in order to generate the filtering process. In addition, the user needs to update his/her set of rules in order to cope with changes that might have happened to the system such as new senders, new messages, etc. Therefore, this system requires the user’s involvement in the process and does not provide automatic updates [3].

- A content-based recommender system for books on the Web, known as LIBRA, was based on training examples supplied by a particular user. The experiments using data from the Amazon.com Web book store showed that the system provided accurate recommendations without using any information about other users [3].

- A semantic-nets filtering based IF, SiteIF [6], used a personal agent that takes into account the user’s browsing. It learns the user’s interests from the requested pages, and uses natural language processing (NLP) techniques to build the user model and the WordNet hierarchy to provide the semantic meaning [3].

Our proposed approach has been implemented on the HyperManyMedia1 platform, was based on training examples supplied by a particular user. As of 2006 the “Hypermanymedia” search engine has been ranked number 24 on “The Ultimate Guide to students at WKU. As of 2006 the “Hypermanymedia” search engine has been ranked number 24 on “The Ultimate Guide to students at WKU. was of many improvements in IF systems, the majority of existing real-life (working) 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, one approach is GHOSTS [5], filters e-mails according to a set of user-defined rules. GHOST requires explicit user input in order to generate the filtering process. In addition, the user needs to update his/her set of rules in order to cope with changes that might have happened to the system such as new senders, new messages, etc. Therefore, this system requires the user’s involvement in the process and does not provide automatic updates [3].

- A content-based recommender system for books on the Web, known as LIBRA, was based on training examples supplied by a particular user. The experiments using data from the Amazon.com Web book store showed that the system provided accurate recommendations without using any information about other users [3].

- A semantic-nets filtering based IF, SiteIF [6], used a personal agent that takes into account the user’s browsing. It learns the user’s interests from the requested pages, and uses natural language processing (NLP) techniques to build the user model and the WordNet hierarchy to provide the semantic meaning [3].

The research field of Semantic E-learning covers a wide range of research problems. To conclude this Section, we divided research in Semantic Web E-learning into four categories:

- Ontologizing Knowledge Flows: Adding an ontology layer to the learning content or creating learning paths on top of domain ontologies [7].

- Semantic Modeling for E-learners: The development of programs based on learning experiences and directed to learners’ profiles [8].

- Semantic Annotation in E-learning Content: Semantic Web technologies, such as OWL and RDF, can integrate learning object components in a hierarchical structure.

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

1HyperManyMedia: We created this term to refer 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).

2Open Courseware: http://www.collegedegree.com/library/collegelife/the_ultimate_guide_to_using_open_courseware

3openlearn.open.ac.uk
looking for. This system uses an ontology recommendation structure. When a user queries for specific information, the retrieval system brings the resources related to this query and also shows the user all the concepts/subconcepts related to the submitted query. Building the ontology is done separately and does not affect the organization of the repository. The process of navigating through the ontology can be understood as a traversal of a directed graph. The linked nodes of the graph represent concepts that are semantically related. When the user hits a node in the ontology, the user can see the children nodes under that concept. If the users are interested in navigating through the deeper level of the hierarchy, they can continue navigating through these nodes, and at the moment they are interested in navigating in the opposite direction (upper concept), they can just click back to go to the parent node. In this way, the users are not restricted to the design of the system, and can freely navigate between concepts and relations that are represented by the Ontology. Our proposed approach implemented on HyperManyMedia [17] platform.

A. Objectives

Our objectives are aligned with the following phases of realizing the proposed system:

- Phase I: Designing and implementing a personalized cluster-based semantic search engine.
- Phase II: Designing a dual representation of the semantic user profile for personalized web search in an evolving domain.

B. Phase I: System Architecture

This phase represents the upper three layers in Figure 1, which are: (1) Semantic representation (knowledge representation), (2) Algorithms (core software), and (3) Personalization interface.

C. Phase I: Methodology

1) 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:

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

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

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

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: $<\text{parent node, concept node, visited node, child node}>$, while a leaf node holds $<\text{parent node, visited node, document, nil}>$. Figure 2 illustrates part of the tree structure generated from the OWL file.

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 $docs(U_i) = \bigcup_{k=1}^{l} d_{ki}$ 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.
Fig. 1. E-learning Personalization Framework with Shifts Detection of Interests

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

Input: docs(Ui) = \{d1, d2, ..., dn\}; // I = # of visited documents by user Ui
Output: BUi = \bigcup_{k=1}^{n} \sum_{r=1}^{q} d_{ki}; // User Ontology Tree(learner’s semantic profile)
R = \bigcup_{k=1}^{n} Ci_{k}; // Domain Ontology Tree
DomainConcept = root;
CollegeConcept = root.child;
While(CollegeConcept <> nil) do
  if (CollegeConcept.child) <> nil
    remove(CollegeConcept, DomainConcept);
  else
    CollegeConcept = 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);
          else
            ParentConcept = SubConcept;
            SubConcept = SubConcept.next;
        end
      end
      UpperConcept = CollegeConcept;
      CourseConcept = CollegeConcept.next;
    end
  end
end

DomainConcept = CollegeConcept;
CollegeConcept = CollegeConcept.next;
BUi = DomainConcept;

Algorithm 2 Best Cluster Mapping Algorithm for a Learner Ui

Input: D = \{d1, d2, ..., dn\}; // I = # of visited docs
Output: BestCluster: // most similar cluster
CL = \bigcup_{k=1}^{n} CL_{k}; // n = # of clusters
BestCluster = CL_1
for each CL_k in CL
  if Sim(D, CL_k) > Sim(D, BestCluster) then
    BestCluster = CL_k
end

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 used the entropy measure [7], [7] 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

\[ E(S_r) = -\frac{1}{n_r} \sum_{i=1}^{q} n_{ri} \log \frac{n_{ri}}{n_r} \] [7], [7]

where \( q \) is the number of classes in the dataset, and \( n_{ri} \) is the number of documents of the \( r \)th class that where 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

\[ E(T) = \sum_{r=1}^{p} \frac{1}{n_r} E(S_r) \] [7], [7]

4) Cluster to Profile Ontology Mapping: Each learner’s profile \( U_i \) is considered as a set \( D \) of documents \( docs(U_i) = \{d_{1}, d_{2}, ..., d_{n}\} \). The domain clusters \( CL = \bigcup_{k=1}^{n} CL_{k} \) are obtained from the clustering in Sec III-C3. 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. 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.

5) Changing the Learner’s Semantic Profile: After extracting the most similar cluster \( CL_{i} = BestCluster \) (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. Figure 4 is a mountain view visualization of the clustering solution, in addition to the features (frequent terms) that represent a descriptive information about each cluster. These features have been used in our ontology.

6) Re-ranking the Learner’s Search Results: We start by representing each of the \( N \) documents as a term vector \( \overrightarrow{d} = \langle w_1, w_2, ..., w_n \rangle \), where \( w_i = tf_i \times \log \frac{N}{n_i} \) is the term weight for term \( i \), combining the term frequency, \( tf_i \), and the term’s Inverse Document Frequency \( (IDF_i = \log \frac{N}{n_i}) \) if 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 cluster that is most similar 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 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) have the lowest priority ($\gamma = 1.0$). All the documents, in each category, are then re-ranked based on 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 4. We modified the boosting score as follows: $doc.setBoost() = \alpha$, in case of Category1, $doc.setBoost() = \beta$, in case of Category2, and $doc.setBoost() = \gamma$, in case of Category3.

**Algorithm 3 Re-ranking a Learner’s Search Results**

Input: $q$ //Keyword search Output: Rank = $\{d_1, d_2, ..., d_n\}$ //Rn – rank

```
Rank = $\{d_1, d_2, ..., d_n\}$ // default search results for query $q$

$UR_1$ = $\cup_{j=1}^{n}$ $RC_j$ = $\cup_{l=1}^{m}$ $d_k$

$RC = \cup_{j=1}^{n}$ $d_j$ // l = # of documents in Recommended Cluster

foreach $d_j$ $\in$ Rank

if $d_j$ $\notin$ $UR_1$ then

$d_j$.boost $\leftarrow$ $\alpha$; // document is in user profile

else if $d_j$ $\in$ $RC$ then

$d_j$.boost $\leftarrow$ $\beta$; // document is in recommended cluster

else

$d_j$.boost $\leftarrow$ $\gamma$;

end

end

Sort Rank based on boost field $d_j$.boost
```

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 Sec III-C5. This framework allows the learner to navigate through the semantic structure of his/her query, as shown in Figure 5, 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.

8) Representing the Semantic Domain: Since November 2006, Western Kentucky University5 has hosted a “HyperManyMedia” open-source repository of lectures6. These resources (lectures) are available in different formats: text, powerpoint, audio, video, podcast, videocast, 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”, “Architect 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.

9) Constructing the Dataset: Our methodology for this phase consists of (1) 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 resulting corpus consists of a total of 2,812 documents indexed under different concepts and sub-concepts. We used these documents to mine the clusters, as explained in Sec III-C3. We constructed an RDF-based (OWL) ontology for our entire E-learning domain based on the hierarchical structure explained in Sec III-C8. Then 10 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 Sec III-E1. We finally constructed keyword queries related to each learner’s profile using terms from the subconcepts (courses name) and lecture names, and a combination of the most significant terms under each concept. Three sets of queries have thus been used (single-keyword, two keywords, three keywords).

---


5http://www.wku.edu

6http://blog.wku.edu/podcasts
D. Phase II: Dual Representation of the Semantic User Profile for Personalized Web Search in an Evolving Domain

In phase II, we added layer 4: Updating the user’s semantic profile, as shown in Figure 1 to detect shifts in the user’s interests.

E. Phase II: Methodology

1) Building the Lower-level of a Learner’s Semantic Profile: 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/revisited. 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 [15], [18]. 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.

2) Building the Higher-level of a Learner’s Semantic Profile: The main purpose of this Section is to extract the user’s weighted interests in lectures in order to detect the user’s shift in interests. We assume that each learner has a relatively stable collection of interests that might change over time. 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 Sec III-C4 and term recommendations Sec III-C7. But if the user (learner) completely shifted his/her interests from concept to concept, a system with a long history of accumulated interests Sec IV-B1 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: VU1 = ∪i=1 l dki // User Concept interests
Output: VU1 = ∪i=1 l SCi // User Concept interests
Algorithm 4 Tracking The User’s History of Interests
Input: doc(Uj) = ∑i=1 l dki // l = # of visited documents by user Uj
Output: VUj = ∪i=1 l SCi // User Concept interests
Call Algorithm 2; // Best cluster Mapping for the user
If (VU1.CollegeConcept <> nil) do
  FindTopInterest(ConceptCollege); // Find user’s College Concept
  While (CollegeConcept.child <> nil) do
    If (CourseConcept.count <> 0) then
      AddSubConceptToVectorWithWeights;
      AddSubConceptToVectorWithWeights;
    End
  End
End
```

IV. Evaluation

A. Phase I: Evaluating the Personalized Cluster-based Semantic Search Engine

We used clustering to divide the documents into an optimal categorization that is not influenced by the hand-made taxonomy of the colleges and course titles. Our main goal is to evaluate the effectiveness of re-ranking based on a learner’s semantic profile.

1) Clustering/Text Analysis: We implemented three different clustering algorithms that are based on the agglomerative, partitional, and graph partitioning paradigms [19]. We compared different hierarchical algorithms for a dataset consisting of 2,812 documents using the clustering package Cluto. We repeated each clustering algorithm with all possible combinations of clustering criterion functions for different number 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. that each cluster contains documents mainly from the same category. However, since a class may be partitioned into several clusters (as was the case here), the clusters are more refined versions of the college categories, which was our desired goal. Of all the algorithms mentioned so far, graph-partitioning produced the best clustering results 1, with K=35 clusters and the lowest entropy. We relabeled each cluster, based on the majority of assigned documents in each college and from each course. After extracting the most similar cluster Ci = BestCluster (recommended-cluster), which is summarized by the 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.

2) Personalized vs. Non-Personalized IR: 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 Sec III-C9, and built the semantic profile for each learner using Algorithm 1, from Sec III-E1. A total of 1,406 lectures (documents) represented the profiles, with the size of each profile varying from one learner to another. Learner1 (English)= 86 lectures, Learner2 (and

7http://glaros.dtc.umn.edu/gkhome/chuto/cluto/overview
Table I: Clustering Entropy Measures for Various Algorithms and Partitioning Criteria

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>I1</th>
<th>I2</th>
<th>E1</th>
<th>G1</th>
<th>G1*</th>
<th>H1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct k-way Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2 (Slink)</td>
<td>0.040</td>
<td>0.042</td>
<td>0.059</td>
<td>0.06</td>
<td>0.043</td>
<td>0.024</td>
</tr>
<tr>
<td>Repeated Bisection Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2 (Slink)</td>
<td>0.045</td>
<td>0.043</td>
<td>0.040</td>
<td>0.086</td>
<td>0.038</td>
<td>0.051</td>
</tr>
<tr>
<td>Graph Partitional Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>log maxtf</td>
<td>0.048</td>
<td>0.038</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agglomerative Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pe</td>
<td>0.031</td>
<td>0.023</td>
<td>0.034</td>
<td>0.098</td>
<td>0.028</td>
<td>-</td>
</tr>
<tr>
<td>pH1</td>
<td>-</td>
<td>0.54</td>
<td>0.54</td>
<td>0.05</td>
<td>0.05</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Family Sciences) = 74 lectures, Learner3 (Communication Disorders) = 160 lectures, Learner4 (Engineering) = 210 lectures, Learner5 (Architect and Manufacturing Sciences) = 119 lectures, Learner6 (Math) = 374 lectures, Learner7 (Social Work) = 86 lectures, Learner8 (Chemistry) = 58 lectures, Learner9 (Accounting) = 107 lectures, Learner10 (History) = 132 lectures. We finally used our semantic search engine 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 6, 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 for single-keyword and two-keywords queries. The recall results show a noticeable improvement in recall between top-20 and top-40. Overall, these results show the effectiveness of the re-ranking based on the learner’s semantic profile.

B. Phase II: Evaluating the Dual Representation of the Semantic User Profile for Personalized Web Search in an Evolving Domain

The main goal of this phase is to find the timeframe where the user’s interests converge. Our assumption is that user’s interests must converge after a period of time. We assume that each person has a relatively stable collection of interests which converges over time. The following sections show the evaluation methodology used and the results.

1) Semantic User Profile Evaluation: Convergence: The time in Figure 7 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 other 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 Sec III-E2, 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 7.

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, as explained in Sec III-C9. Each profile is generated using Algorithm 1. Figure 8 shows the users’ activities over a period of one year. Ordering these lectures was based on concepts, as we 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

8http://blog.wku.edu/podcasts
appears to stabilize after a period of time. We modified Algorithm 1, in this stage, as shown in Algorithm 4. Based on our experimental results, the users’ interests converge after a period of one month, as shown in Figure 7.

Note 1: The list of queries and the developed E-learning Ontology Structure that have been used to evaluate this research are provided on this website http://blog.wku.edu/podcasts/Evaluations.php

V. Conclusion

We presented an information retrieval platform for personalized search that took advantage of the semantic Web standards (RDF and OWL) to represent the content and the user profiles. This framework used a combination of an authoritatively supplied taxonomy by the colleges, with the adaptive 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. The experimental results of this phase showed 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. The second phase of this work dealt with the problem of detecting shifts in users’ interests for personalized search to cope with a domain of information that evolves and user interests that change over time. In this model, the lower-level semantic representation consisted of an accumulated gathering of user activities over a long period of time that used a standard machine learning algorithm to detect user convergence where the higher-level semantic representation detected the shifts in the user activities, and in case this shift was detected, the higher-level semantic representation automatically updated the user profiles and reinitialized the system. The experimental results showed that the users’ interests converge after a period of time and during this period, if they changed their interest, the system can detect this change and adapt the user profile. This research has been implemented on a real platform called “HyperManyMedia” at Western Kentucky University. Over all, the system provides satisfactory retrieval experience to online learners.

References


