2008 20th IEEE International Conference on Tools with Artificial Intelligence

Semantic Information Retrieval for Personalized E-learning

Leyla Zhuhadar
Knowledge Discovery and Web Mining Lab
Department of Computer Engineering
and Computer Science
University Of Louisville
Louisville, KY, USA
l0zhuh01@Louisville.edu

Olfa Nasraoui*
Knowledge Discovery and Web Mining Lab
Department of Computer Engineering
and Computer Science
University Of Louisville
Louisville, KY, USA
olfa.nasraoui@Louisville.edu

Abstract

We present an approach for personalized retrieval in an e-learning platform, that takes advantage of semantic Web standards to represent the learning content and the user/learner profiles as ontologies, and that re-ranks search results/lectures based on how the contained terms map to these ontologies. 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 content retrieval, particularly by re-ranking the search results based on the learner's past activities.

1. Introduction and Related Work

Personalizing the retrieval of 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 semantic user interests or profiles can form a good representation of the learning context, that promises to enhance the results of retrieval via personalization. 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:

*This work is supported by National Science Foundation CAREER Award IIS-0133948 to Olfa Nasraoui.

1- Ontologizing knowledge flows: Adding an ontology layer to the learning content [1] or creating learning paths on top of domain ontologies [4].

2- Semantic Modeling for e-learners: The development of programs based on learning experiences and directed to learners' profiles[5].

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-Network: Promoting a semantic vision of e-learning that can change the view from individual learning to collaborative learning,. For example, OpenLearn[1], at the Open University, UK, enables users to learn together.

Of all related work, the idea in [6] 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 and the specific topics being learned), and **(2)** our search engine provides re-ranking based not only on the user's profile, but also on cluster-based similarity metrics.

2. 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.

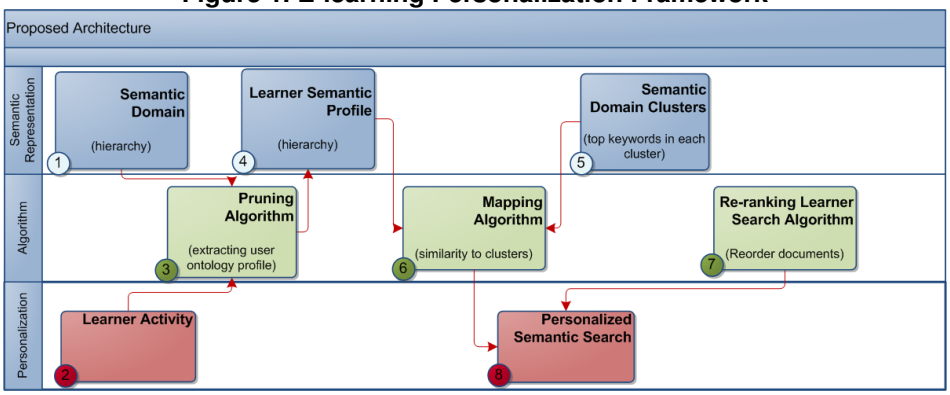
2.1. Semantic Domain Structure

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:

[1] openlearn.open.ac.uk

1082-3409/08 $25.00 © 2008 IEEE
DOI 10.1109/ICTAI.2008.130

Figure 1. E-learning Personalization Framework



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

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>***.

2.2. 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 $docs(U_i) = \cup_{k=1}^{l} d_{ki}$ 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 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. 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) = ∪_{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;
```

2.3. Cluster-based Semantic Profiles

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 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 [2]. 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. 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 is our desired aim. We used the *entropy measure* [8] 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 [8]. We implemented three different clustering algorithms that are based on the agglomerative, partitional, and graph partitioning paradigms [7]. Each algorithm uses a different algorithm for clustering. In agglomerative algorithms, starting 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, using different criterion functions [7]: I1, I2, E1, G1, G1*, H1, H2, 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 H2 criterion, with u and v, being documents and S_i being the i^{th} cluster, containing n_i documents, while $sim(u, v)$ denotes the similarity between u and v [8]. 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 [8]. 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 [8] performed best, whereas H1 [8] 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 [3] [2]. 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.

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

Algorithm 2 Best Cluster Mapping algorithm for a learner U_i

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

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	-	-	-	-	-

2.4. 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 2.3. 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.

2.5. 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.

2.6. Re-ranking the Learner's Search Results

We start by representing each of the N documents as a term vector $\overrightarrow{d} =< w_1, w_2, ...w_n >$, 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 ($IDF_i = log\frac{N}{n_i}$) 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 relationship 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 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 [3]. 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.

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

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

3. Implementation

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 2.3. We constructed an RDF-based (OWL) ontology for our entire e-learning domain based on the hierarchical structure. Then 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 2.2. 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 datasets of queries have thus been used (single-keyword, two keywords, three keywords).

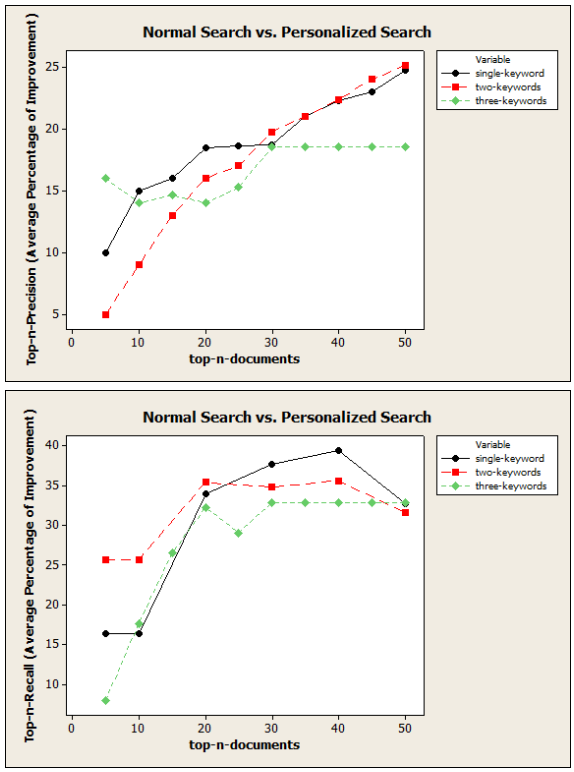
4. Experimental Evaluation

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 sub-concepts (courses), and a total of 2,812 lectures (documents) that were indexed under various concepts\sub-concepts. After constructing the domain ontology, we selected 10 learner profiles, as explained in Section 3, and built the semantic profile for each learner using Algorithm 1, from Section 2.2. 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* (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, *Learner10* (History) =132 lectures. We finally used our semantic search engine [4] 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 2, show the Average Percentage of Improvement in Top-n Re-

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

[4] http://blog.wku.edu/podcasts

Figure 2. Average Percentage of Improvement in Top-n Recall and Top-n Precision



call 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.

5. 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 forgetting factor to gradually forget older user activity is another interesting and adaptive approach.

6. 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).

References

[1] L. Aroyo, P. Dolog, G.J. Houben, M. Kravcik, A. Næve, M. Nilsson, and F. Wild. Interoperability in personalized adaptive learning. *Educational Technology & Society*, 9(2):4–18, 2006.

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

[3] B.M. Sarwar, G. Karypis, J.A. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system–a case study. *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, 2000.

[4] M.A. Sicilia. Metadata, semantics, and ontology: providing meaning to information resources. *International Journal of Metadata, Semantics and Ontologies*, 1(1):83–86, 2006.

[5] M.A. Sicilia and M.D. Lytras. The semantic learning organization. *The Learning Organization*, 12(5):402–410, 2005.

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

[7] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. *Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524, 2002.

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