2011 IEEE International Conference on Fuzzy Systems
June 27-30, 2011, Taipei, Taiwan

Fuzzy Folksonomy-based Index Creation for e-Learning Content Retrieval on Cloud Computing Environments

Wen-Chung Shih
Department of Applied Informatics and Multimedia, Asia University
Taichung, 41354, Taiwan
wjshih@asia.edu.tw

Chao-Tung Yang
Department of Computer Science
Tunghai University
Taichung, 40704, Taiwan
ctyang@thu.edu.tw

Shian-Shyong Tseng*
Department of Applied Informatics and Multimedia, Asia University
Taichung, 41354, Taiwan
sstseng@asia.edu.tw

Abstract—**Due to the trend of individualization and adaptation of e-Learning, more and more SCORM-compliant teaching materials are developed by institutes and individuals in different sites. Also, cloud computing environments are emerging as powerful infrastructures to support e-Learning applications. Therefore, how to rapidly retrieve SCORM-compliant documents on cloud computing environments has become an important issue. Creating an index from folksonomies has been investigated in previous researches; however, the involved uncertainty has not been addressed. This paper focuses on the fuzzy index creation problem for learning content retrieval. A bottom-up approach to constructing the fuzzy index is proposed. The index creation method has been implemented, and a synthetic learning object repository has been built on a Hadoop cloud platform to evaluate the proposed approach. Experimental results show that this method can increase precision of retrieval.**

Keywords- Folksonomy; Fuzzy Sets; Information retrieval; e-Learning; Cloud computing

I. INTRODUCTION

Due to the advances in information technologies and the requirements of courseware, more and more teachers are able and willing to design their own teaching materials and make them accessible on the Web [1-3]. In addition, a growing number of large-scale projects aim to construct learning content repositories [4, 5]. For example, in 2002, the National Science Council of Taiwan approved a resolution on the ‘‘National Science and Technology Program for e-Learning,’’ planning to spend $120 million within a five-year period [6]. These educational contents are mainly based on Sharable Content Object Reference Model (SCORM) [7], which has become a popular standard for creating sharable and reusable teaching materials for e-Learning. With the popularization of e-Learning, how to find and reuse these existing materials becomes an important issue.

Cloud computing systems [8, 9] are transparent resource-sharing infrastructures, which can overcome the limitations in traditional e-Learning platforms, such as scalability, interoperability, availability, etc. Also, Cloud computing technologies provide possibilities for supporting innovative applications of e-Learning. For example, a medical college can provide students with three-dimensional simulation of human body anatomy using high performance cloud computing systems, which is beyond the ability of traditional e-Learning platforms. Therefore, more and more effort has gone into the field of e-Learning cloud, using cloud technologies in the context of e-Learning.

With the promising development of e-Learning cloud, there will be a great demand to find desired teaching materials from multiple repositories in the e-Learning cloud. A traditional approach is to implement a meta-search engine on top of these distributed repositories [10]. When the meta-search engine receives a query, it distributes the query to the local repositories, and then collects the returned results and presents them to users. However, this traditional approach doesn’t consider characteristics of SCORM and cloud computing to speed up the retrieval process. For example, SCORM-compliant documents are associated with nine categories of metadata. When processing a query with a restrictive filter, such as ‘‘documents about insects,’’ the meta-searching approach will search the whole databases.

We classify the schemes of content retrieval into static and dynamic ones according to the adaptability of the retrieved results. For static CR, the retrieved result only depends on the query, independent of users and contexts. Dynamic CR can be further divided into personalized, context-aware and other schemes, according to the factors that are considered by the adaptive mechanisms of CR. To support context-aware CR, the teaching materials stored in the repositories have to be organized according to their contextual information, in order for efficient retrieval. However, it is almost impossible to request experts to annotate all these contents with suitable context-aware metadata. Therefore, content annotation based on folksonomies and automatic techniques in a collaborative way is a promising solution.

In well-known folksonomy applications, such as del.icio.us (http://del.icio.us/), Flickr (http://www.flickr.com), etc., folksonomies are organized into a flat structure, which consists of several catagories named by user-defined tags. This flat organization is suitable for users to manage their preferences. However, when the size of repositories gets larger and larger, a hierarchical organization becomes a better choice to organize the contents. To bridge the gap of the two structures, we formulate this index creation problem and propose a bottom-up approach to constructing an index from existing folksonomies according to the similarity between tags, which considers metadata and structural information of the teaching materials annotated by the tags [11]. Furthermore, creating an index from folksonomies often involves uncertainty. For example, which tags are associated with which categories. Therefore, a fuzzy representation is proposed to address this issue. The index creation method has been implemented, and a synthetic learning object repository has been built to evaluate the proposed approach. Experimental results show that this method can increase precision of retrieval.

The contributions can be summarized as follows. First of all, we propose a fuzzy folksonomy-based method for index creation, which can reduce the effort required in subsequent work of location-aware content retrieval. With this method, the heavy burden of experts for manually developing concept hierarchy can be significantly alleviated. Second, a similarity function is proposed to increase the precision of folksonomy fusion, which considers more characteristics of learning contents, including metadata and structural information of the teaching materials annotated by the tags. Next, a self-organizing mechanism can designed to balance the number of documents annotated by the tags, which can increase the performance of indexing structures. Finally, the proposed method is implemented and the built index is evaluated. Experimental results reveal that this method can improve the performance of retrieval.

The rest of this work is organized as follows. In Section II, we review the preliminaries and previous work on related researches. Then, the problem is formulated in Section III. Next, the method of index creation is described in Section IV. Section V illustrates experimental results. Finally, the concluding remarks are given in Section VI.

II. Preliminaries and Related Work

In this section, we review previous research related to this work. First, the SCORM model is introduced. Then, methods of information retrieval are described. Finally, we review methods of ontology building, which is related to index creation.

A. Sharable Content Object Reference Model (SCORM)

To share and reuse teaching materials, several standards have been proposed recently. Among these, SCORM (http://www.adlnet.org/) is the most popular standard for learning contents. It was proposed by the U.S. Department of Defense's Advanced Distributed Learning (ADL) organization in 1997. This standard consists of several specifications developed by IEEE LTSC (Learning Technology Standards Committee, http://ltsc.ieee.org/wg12/), IMS (Instructional Management System, http://www.imsproject.org/), AICC (Aviation Industry CBT Committee, http://www.aicc.org/), etc. The SCORM specifications are a composite of several specifications developed by international standards organizations. In a nutshell, SCORM is a set of specifications for developing, packaging and delivering high-quality education and training materials whenever and wherever they are needed [12, 13]. In SCORM, content packaging scheme is proposed to package the learning objects into standard teaching materials. The content packaging scheme defines a teaching materials package consisting of four components: 1) Metadata, which describes the characteristics or attributes of this learning content; 2) Organizations, which describe the structure of the teaching material; 3) Resources, which denote the physical files linked by each learning object within the teaching material; and 4) the (Sub) Manifest, which describes this teaching material, consisting of itself and other teaching materials. SCORM Metadata refers to the IEEE's Learning Object Metadata (LOM), and describes the attributes of teaching materials. IEEE LOM v1.0 includes nine categories: General, LifeCycle, Meta-Metadata, technical, educational, rights, relation, annotation, and classification.

Su et al. proposed a level-wise approach to SCORM content management [14]. In this two-phase scheme, structural information is considered. Also, both general and specific Learning Objects can be retrieved according to requests of users. However, the design of this structure doesn't consider its usability in ubiquitous learning environments.

There have been numerous studies on Structured Document Retrieval [15, 16]. Researches showed that structured searching can increase precision. Previous work mainly addresses XML and SGML documents. Besides, XML Query Languages, such as XIRQL, XQL, etc., were proposed. However, intra-document structural modeling is not suitable for SCORM-compliant documents.

B. Search Engine Technologies

Inverted file indexing has been widely used in information retrieval [17-20]. An inverted file is used for indexing a document collection to speed up the searching process. The structure of an inverted file consists of two components: the vocabulary and the posting list. The vocabulary is composed of all distinct terms in the document collection. For each term, a list of all the documents containing this term is stored. The set of all these lists is called the posting list. However, the structure of a document is not considered in this model.

Storage requirements of inverted indices [21] have been evaluated based on B+-tree and posting list. Five strategies of the index term replication were discussed. This approach is extended to analyze the storage requirement of the proposed approach in this proposal. In [22], 11 different implementations of ranking-based text retrieval systems using inverted indices were presented, and their time complexities were also investigated.

The meta-search approach has been studied in the context of distributed information retrieval [10]. This approach consists of Query Distribution and Result Merging phases. Furthermore, the Document Retrieval problem is divided into two sub-problems: Database Selection problem and Document Selection problem. However, Distributed indexing is not suitable for common cloud architectures. Also, distributed information retrieval is less efficient in searching.

The use of ontologies to overcome the limitations of keyword-based search has been put forward as one of the motivations of the Semantic Web since its emergence in the late 1990s. One way to view a semantic search engine is as a tool that gets formal ontology-based queries from a client, executes them against a knowledge base, and returns tuples of ontology values that satisfy the query. In this view, the information retrieval problem is reduced to a data retrieval task. While this conception of semantic search brings key advantages already, our work aims at taking a step beyond.

A purely Boolean ontology-based retrieval model makes sense when the whole information corpus can be fully represented as an ontology-driven knowledge base. But, there are well-known limits to the extent to which knowledge can be formalized this way. Boolean search does not provide clear ranking criteria, without which the search system may become useless if the retrieval space is too big.

C. Ontology Building Approaches

We review ontology building approaches because its construction process is similar to index creation in concept. Traditionally, ontology building mainly depends on the contribution of domain experts in the knowledge creation activity. Metadata extraction and merging is carried out manually by domain experts. Many tools have been developed for ontology developers to access ontologies, browse them, edit them, and propose modifications. However, some drawbacks do exist. This process is time-consuming and arduously. For this reason, recent researches turned to automatic and semiautomatic (as opposed to manually) ontology construction and maintenance. Automatic techniques for building and integrating ontologies have been studied for many years by the artificial intelligence community. More recently, several techniques specifically aimed at learning ontologies from text corpora or database repositories were proposed. Well-known research approaches to ontology building include [23]:

- Dictionary-based approach [24]: constructs the hierarchy of concepts based on a traditional dictionary, which presents the related concepts of words, including synonyms, etymology, etc.
- Conceptual clustering [25]: Concepts are grouped according to a semantic distance between each other to generate hierarchical relations.
- Association rules mining [26]: The frequency of an association of terms is computed in the text repositories. If the frequency of the association is close to the occurrence of individual terms, the association is transformed in an ontological relation.
- Formal concept analysis [23]: is a method for representation, analysis and management of data and knowledge. It can be used to build a hierarchy of terms and associated relations.

In [8], a fuzzy ontology extraction algorithm was proposed for adaptive e-learning. Folksonomies have not been widely applied to the subject of index creation. Although folksonomies are not formal models for knowledge representation, the collaborative wisdom of resource categorization can be a good start point from which effective indices can be built.

III. Problem Formulation

Before the index creation problem is described, some concepts and models are introduced. First, a level-wise structural model of learning contents is presented, which is intended to model the semantic features of different levels of a structural learning content.

Definition. Content Package (CP)

A Content Package represents a structural learning content, such as a SCORM-compliant teaching material. It is a rooted tree. The leaf node represents textual content, which is characterized by a feature vector = $<w_1, w_2, \ldots, w_n>$, where n is the size of the vocabulary and the weighting scheme is the widely used TF-IDF (Term Frequency – Inverse Document Frequency) [17, 19]. Internal nodes represent structural information of this teaching material. Each Content Package is associated with two types of attributes:

- Level j feature vector, L_j, $(0 < j < Height)$

where L_j = Average of feature vectors of nodes at level j, and Height is the height of the rooted tree;

- Metadata, $\{M_k \mid k = 1 \text{ to } m, m \text{ is the number of metadata}\}$.

A folksonomy is a user generated taxonomy used to categorize and retrieve web content such as Web pages, photographs and Web links, using user-defined labels called tags. Typically, a folksonomy has a flat structure, which consists of several user-defined categories. The folksonomic tagging is intended to make a body of information increasingly easy to search, discover, and navigate over time. One well-known website using folksonomic tagging is del.icio.us. In this work, fuzzy folksonomies are modeled as follows. The proposed model is based on Zadeh's theory of fuzzy sets [27].

Definition. Fuzzy Folksonomy

A fuzzy folksonomy defined by a user is a triple of (T, I, R), where T is a fuzzy set of tags, I is a fuzzy set of content packages and R is a fuzzy relation on $T \times I$. We say that tag t is related to a content package i if i is annotated by t. Each tag is associated with two types of attributes, which are derived from content packages annotated by this tag.

- Level j feature vector, L_j, $(0 < j < Height)$

where L_j = the average of the level j feature vectors of content packages annotated by the tag, and *Height* is the height of the rooted tree;

- Metadata, $\{M_k \mid k = 1 \text{ to } m, m \text{ is the number of metadata}\}$

where the value of M_k is defined by the most frequent M_k value of the content packages annotated by the tag.

The following example illustrates this definition. Figure 1 depicts two folksonomies defined by users A and B, respectively. The folksonomy of user A can be represented by

$T = \{t_1, t_2\}$

$I = \{i_1, i_2, i_3, i_4\}$

$R = \{(t_1, i_1), (t_1, t_2), (t_1, i_3), (t_2, i_1), (t_2, i_4)\}$

where i_1, i_2, i_3 and i_4 are content packages.

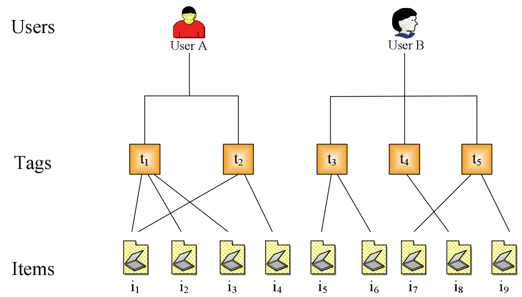


Figure 1. The folksonomy of user A

In the field of computer science, ontology is a data model that represents a set of concepts within a domain and the relationships between those concepts. It is used to understand the objects within that domain. However, there has not been a commonly acceptable definition for ontologies. The following definitions describe ontology-like indices referred to in this paper. First, the relations, general and specific, are stated.

Definition. General

A tag *a* is more general than a tag *b* if the set of teaching materials annotated by *a* includes those annotated by *b*.

Next, we define the index.

Definition. Index

An index is considered to be a rooted tree. The nodes represent concepts in the domain, and the edges represent relations between nodes. A parent node *p* is more general than its child node *c*. Each node is associated with a feature vector, which characterizes the semantic meaning of this concept.

Based on the definitions above, the index creation problem is stated as follows.

Definition. The Fuzzy Folksonomy-based Index Creation Problem (FICP)

Given a collection of content packages and corresponding folksonomies, generate an index. The objective is to improve performance of learning content retrieval.

IV. FUZZY FOLKSONOMY-BASED INDEX CREATION

Our idea to solve the FICP problem is based on the heuristic that existing folksonomies generated by users can be a good starting point from which to construct a location-aware index. While most folksonomies are organized into flat structures, we plan to build hierarchical indices to better organize the learning content. To bridge the gap of the two structures, folksonomies and indices, we propose a bottom-up approach to construct an index from existing folksonomies according to the similarity between tags, which considers metadata and structural information of the teaching materials annotated by the tags.

We design an algorithm to merge two folksonomies into one which is used for subsequent information retrieval. The idea is to make decisions of tag merging according to the similarity of the two tags. The main difficulty is how to choose a suitable similarity function for SCORM-compliant teaching materials, which are characterized by textual content, metadata and structural information. Here, a similarity measure for two tags, *a* and *b*, is proposed:

$$Sim(a,b) = (1-\beta)\sum_{i=0}^{Height}\alpha_i \times Sim_i(a,b) + \beta \times Sim_M(a,b) \qquad (1)$$

where the sum of $_i$ is equal to one, and 0< <1. The parameter, $_i$, is used to adjust the weighting of level-wise content vector. The parameter is used to adjust the weighting of metadata similarity. The similarity function consists of two parts:

- Sim_i: level *i* similarity function, which is *cosine* function, 0< *i* < *Height*. The similarity between two vectors v_k = <k_1, k_2, …, $k_{|V|}$> and v_p = <p_1, p_2, …, $p_{|V|}$> is measured by the following formula:

$$sim_i = \frac{\sum_{i=1}^{|V|} k_i \times p_i}{\sqrt{\sum_{i=1}^{|V|} k_i^2} \times \sqrt{\sum_{i=1}^{|V|} p_i^2}} \qquad (2)$$

- Sim_M: Metadata similarity function, which is (the number of matched attributes) / (the number of all attributes).

The Folksonomy-based Index Creation Algorithm consists of two steps: initializing the master folksonomy and iteratively merging tags of the other folksonomy into the master folksonomy, listed as follows.

Algorithm. Fuzzy Folksonomy-based Index Creation Algorithm (AlgFFIC)
Input:
  F_1, F_2: the two folksonomies to be merged
  Th_sim: the threshold for comparing similarity
  sim: the similarity function
Output:
  F: the merged folksonomy

Step 1. Initialization
  1.1 Each tag of F_1 and F_2 is represented by the average of its related teaching materials.
  1.2 F_1 is assigned as the Master folksonomy, *M*.
Step 2. for each tag *t* in F_2
  2.1 calculate the similarity of t and each tag of *M*.
  2.2 let *t_close* be the closest tag in *M* to *t*.
  2.3 if the *sim*(*t*, *t_close*) > *Th_sim* then
      add the tag *t* into *t_close*
    else
      add tag *t* into *M* as a new tag

Step 1 of the *AlgFFIC* algorithm evaluates attributes of the input folksonomies. Also, one of the input folksonomy is assigned as the Master folksonomy, which serves as the base for other tags to join. Next, tags of the other folksonomy are merged into the Master folksonomy one after another in Step 2. Finally, a merged folksonomy is generated.

In Step 2.3, a threshold, *Th_sim*, is set to decide whether to merge the tag or construct a new tag. When the similarity is greater than the threshold, the two tags are merged. Otherwise, a new tag is constructed in the master folksonomy for the dissimilar tag from the other folksonomy.

V. IMPLEMENTATION

In this experiment, we have implemented a cloud portal for query submission and results presentation. In addition, results are shown with estimated transmission time. The corpus is composed of synthetic SCORM-compliant documents. The experiments investigate searching performance in terms of query processing time.

A. *Prototype Design*

A folksonomy-based information retrieval system is implemented to evaluate the proposed method, as shown in Figure 2. The objective is to increase precision by folksonomy-based semantic search, and to reduce search time by folksonomy-based indexing. The phase of constructing a concept hierarchy is based on the proposed method, in order to alleviate the heavy burden of experts and knowledge engineers. This framework consists of three phases. In the concept hierarchy construction phase, users' folksonomies are merged into a hierarchy, which can be referred by the index creation phase. A bottom-up method is designed to organize teaching materials located in different repositories, according to the generated index. A global index is then created to facilitate semantic search. Finally, users' queries are submitted in the search phase, and desired teaching materials are retrieved fast and precisely.

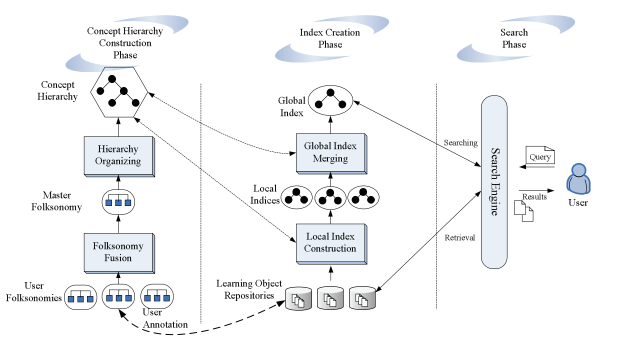
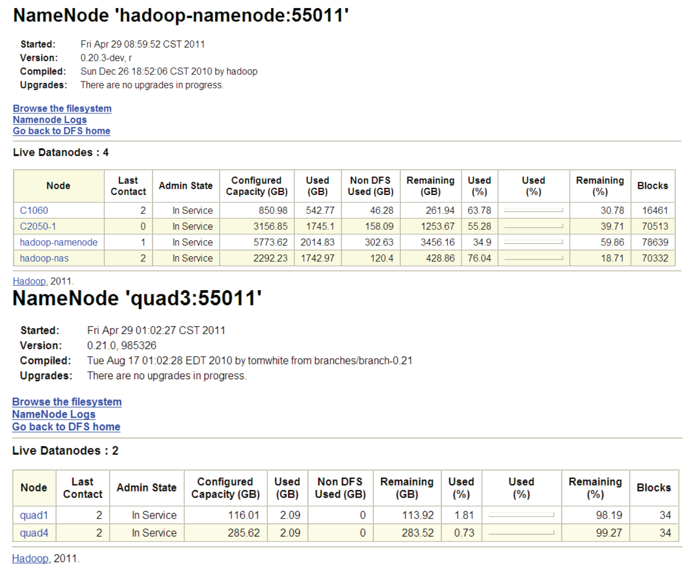


Figure 2. The overview of a folksonomy-based information retrieval system

A Cloud computing test-bed has been built by the High Performance Computing Lab. of Tunghai University, Taichung, Taiwan, using Hadoop. The summary of the nodes is shown in Figure 3.



NameNode 'hadoop-namenode:55011'

Started: Fri Apr 29 08:59:52 CST 2011
Version: 0.20.3-dev, r
Compiled: Sun Dec 26 18:52:06 CST 2010 by hadoop
Upgrades: There are no upgrades in progress.

Browse the filesystem
Namenode Logs
Go back to DFS home

Live Datanodes : 4

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
C1060	2	In Service	850.98	542.77	46.28	261.94	63.78		30.78	16461
C2050-1	0	In Service	3156.85	1745.1	158.09	1253.67	55.28		39.71	70513
hadoop-namenode	1	In Service	5773.62	2014.83	302.63	3456.16	34.9		59.86	78639
hadoop-nas	2	In Service	2292.23	1742.97	120.4	428.86	76.04		18.71	70332

Hadoop, 2011.

NameNode 'quad3:55011'

Started: Fri Apr 29 01:02:27 CST 2011
Version: 0.21.0, 985326
Compiled: Tue Aug 17 01:02:28 EDT 2010 by tomwhite from branches/branch-0.21
Upgrades: There are no upgrades in progress.

Browse the filesystem
NameNode Logs
Go back to DFS home

Live Datanodes : 2

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
quad1	2	In Service	116.01	2.09	0	113.92	1.81		98.19	34
quad4	2	In Service	285.62	2.09	0	283.52	0.73		99.27	34

Hadoop, 2011.

Figure 3. The node summary of the cloud test-bed

B. *Evaluation*

Two synthetic LORs are used in this experiment. The first LOR contains 1,200,000 SCORM-compliant documents, which are converted from Web pages related to educational domains. After stop-word cleansing, there remains 2,570,623 distinct index terms. The size of this LOR is around 20 GB. The other LOR contains 2,400,000 SCORM-compliant documents, which are converted from technical papers related to computer science domains. After stop-word elimination, there remains 4,730,384 distinct index terms. The size of this LOR is around 40 GB. We have implemented two algorithms for this experiment. The first algorithm is a traditional meta-searching approach, named DIR (Distributed Information Retrieval). The other is the approach using the proposed approach, named CIR (Cloud Information Retrieval).

Figure 4 presents the running time of two implementations for different numbers of query terms on LOR 1. In the three cases, CIR performs better than DIR. Figure 5 presents similar results on LOR 2. The main reason may be that CIR using the proposed approach can effectively speed the searching processing.

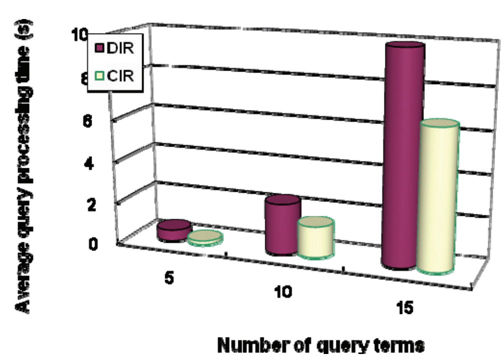


Figure 4. The running time for different numbers of query terms on LOR 1

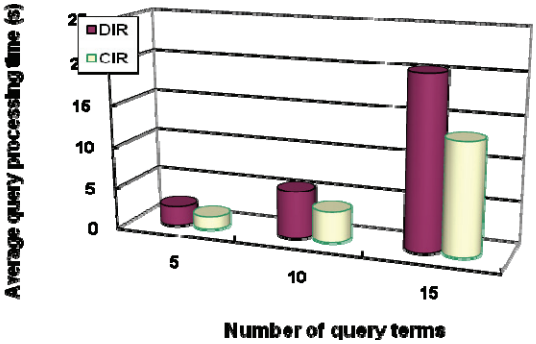


Figure 5. The running time for different numbers of query terms on LOR2

VI. CONCLUSIONS

This paper describes a fuzzy folksonomy-based approach to index creation for learning content retrieval. We divide the index creation problem into two sub-problems: folksonomy fusion and hierarchy organizing. The former problem is solved by merging tags according to a similarity function, which considers textual, metadata and structural information of teaching materials. Experimental results show the effectiveness of this approach. This folksonomy-based approach is characterized by a time-saving development process, minimal involvement of experts and high performance of information retrieval.

Learning content retrieval includes many important and challenging issues. After the study of this work, the future work will address the possibility of complementing traditional ontologies with folksonomies. In addition, it is a promising way to use Expert Systems technologies to facilitate the searching process.

ACKNOWLEDGMENT

This research was partially supported by National Science Council of Republic of China under the number of NSC97-2511-S-468-004-MY3, NSC98-2511-S-468-004-MY3, NSC98-2511-S-468-002 and NSC99-2511-S-468-003.

REFERENCES

[1] A. D. Iorio, A. A. Feliziani, S. Mirri, P. Salomoni, and F. Vitali, "Automatically Producing Accessible Learning Objects," *Journal of Educational Technology & Society,* vol. 9, pp. 3-16, 2006.

[2] R. Lanzilotti, C. Ardito, M. F. Costabile, and A. D. Angeli, "eLSE Methodology: a Systematic Approach to the e-Learning Systems Evaluation," *Journal of Educational Technology & Society,* vol. 9, pp. 42-53, 2006.

[3] J. L. Sierra, A. Fernández-Valmayor, M. Guinea, and H. Hernanz, "From Research Resources to Learning Objects: Process Model and Virtualization Experiences," *Journal of Educational Technology & Society,* vol. 9, pp. 56-68, 2006.

[4] C.-C. Kiu and C.-S. Lee, "Ontology Mapping and Merging through OntoDNA for Learning Object Reusability," *Journal of Educational Technology and Society,* vol. 9, pp. 27-42, 2006.

[5] A. Kassahun, A. Beulens, and R. Hartog, "Providing Author-Defined State Data Storage to Learning Objects," *Journal of Educational Technology & Society,* vol. 9, pp. 19-32, 2006.

[6] ELNP, "National Science and Technology Program for e-Learning." vol. 2007, 2002.

[7] SCORM, "Sharable Content Object Reference Model (SCORM)." vol. 2006: Advanced Distributed Learning, 2004.

[8] R. Y. K. Lau, D. Song, Y. Li, T. C. H. Cheung, and J.-X. Hao, "Toward a Fuzzy Domain Ontology Extraction Method for Adaptive e-Learning," *IEEE Transactions on Knowledge and Data Engineering,* vol. 21, pp. 800-813, 2009.

[9] N. Leavitt, "Is Cloud Computing Really Ready for Prime Time?," *IEEE Computer Magazine,* vol. 42, pp. 15-20, 2009.

[10] C. Yu, K. L. Liu, W. Meng, Z. Wu, and N. Rishe, "A Methodology to Retrieve Text Documents from Multiple Databases," *IEEE Transactions on Knowledge and Data Engineering,* vol. 14, pp. 1347-1361, 2002.

[11] W.-C. Shih and S.-S. Tseng, "Folksonomy-based Indexing for Location-aware Retrieval of Learning Contents," in *The 5th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education (WMUTE 2008)*, Beijing, China, 2008, pp. 143-147.

[12] E. D. Nitto, L. Mainetti, M. Monga, L. Sbattella, and R. Tedesco, "Supporting Interoperability and Reusability of Learning Objects: The Virtual Campus Approach," *Educational Technology & Society,* vol. 9, pp. 33-50, 2006.

[13] J.-M. Su, S.-S. Tseng, W. Wang, J.-F. Weng, J. T. D. Yang, and W.-N. Tsai, "Learning Portfolio Analysis and Mining for SCORM Compliant Environment," *Educational Technology & Society,* vol. 9, pp. 262-275, 2006.

[14] J. M. Su, S. S. Tseng, C. Y. Wang, Y. C. Lei, Y. C. Sung, and W. N. Tsai, "A Content Management Scheme in SCORM Compliant Learning Object Repository," *Journal of Information Science and Engineering,* vol. 21, pp. 1053-1075, 2005.

[15] A. Trotman, "Searching structured documents," *Information Processing and Management,* vol. 40, pp. 619-632, 2004.

[16] A. Trotman, "Choosing Document Structure Weights," *Information Processing and Management,* vol. 41, pp. 243-264, 2005.

[17] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval*. New York: ACM Press, 1999.

[18] A. Mittal, P. V. Krishnan, and E. Altman, "Content Classification and Context-Based Retrieval System for E-Learning," *Journal of Educational Technology & Society,* vol. 9, pp. 349-358, 2006.

[19] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw & Hill, 1983.

[20] I. H. Witten, A. Moffat, and T. C. Bell, *Managing gigabytes: compressing and indexing documents and images*, 2nd ed. ed. San Francisco, California: Morgan Kaufmann, 1999.

[21] Y. K. Lee, S.-J. Yoo, K. Yoon, and P. B. Berra, "Index structures for structured documents," in *the 1st ACM international conference on digital libraries*, 1996, pp. 91-99.

[22] B. B. Cambazoglu and C. Aykanat, "Performance of query processing implementations in ranking-based text retrieval systems using inverted indices," *Information Processing and Management,* vol. 42, pp. 875-898, 2006.

[23] S.-S. Weng, H.-J. Tsai, S.-C. Liu, and C.-H. Hsu, "Ontology construction for information classification," *Expert Systems With Applications,* vol. 31, pp. 1-12, 2006.

[24] L. Khan and F. Luo, "Ontology Construction for Information Selection," in *the 14th IEEE internation conference on tools with artificial intelligence*, Washington DC, 2002, pp. 122-127.

[25] A. Hotho, A. Maedche, and S. Staab, "Ontology-based Text Clustering," in *the IJCAI-2001 workshop text learning: Beyond supervision*, Seattle, 2001.

[26] A. Maedche and S. Staab, "Ontology Learning for the Semantiv Web," *IEEE Intelligent Systems,* vol. 16, pp. 72-79, 2001.

[27] L. A. Zadeh, "Fuzzy Sets," *J. Information and Control,* vol. 8, pp. 338-353, 1965.