Ranking Metrics and Search Guidance for Learning Object Repository

Neil Y. Yen, Timothy K. Shih, Senior Member, IEEE, Louis R. Chao, and Qun Jin, Member, IEEE

Abstract—In line with the popularity of the Internet and the development of search engine, users request information through web-based services. Although general-purpose searching such as one provided by Google is powerful, searching mechanism for specific purposes could rely on metadata. In distance learning (or e-learning), SCORM provides an efficient metadata definition for learning objects to be searched and shared. To facilitate searching in a federated repository, CORDRA provides a common architecture for discovering and sharing Learning Objects. We followed SCORM and CORDRA specifications to develop a registry system, called the MINE Registry, for storing and sharing 20,738 Learning Objects created in the past five years. As a contribution, we propose the concept of “Reusability Tree” to represent the relationships among relevant Learning Objects and enhance CORDRA. We further collect relevant information, while users are utilizing Learning Objects, such as citations and time period persisted. The feedbacks from the user community are also considered as critical elements for evaluating significance degree of Learning Objects. Through these factors, we propose a mechanism to weight and rank Learning Objects in the MINE Registry, in addition to other external learning objects repositories. As a practical contribution, we provide a tool called “Search Guider” to assist users in finding relevant information in Learning Objects based on individual requirements.

Index Terms—CORDRA, LOM, learning object repository, ranking metrics, search guidance, reusability tree, social feedback, information retrieval, distance learning.

1 INTRODUCTION

In the 21st century, worldwide competition is on the rise. To efficiently retrieve and utilization of high-quantity information. The competition includes the creation of a huge volume of knowledge on the Internet and the rapid incrementation of personalized learning to acquire knowledge. Hence, how to enhance the learning efficiency and to help world citizens to achieve high quality of life-long learning is the key research domain of education. Computer-Based Training (CBT) and Web-Based Training (WBT) had provided a different learning style from traditional education since late 1960s. With the improvement of the Internet and communication devices, people are eager for grabbing knowledge anytime everywhere. Hence, distance/mobile learning has become more and more important nowadays.

The development of information systems (and thus, knowledge system) can be briefly categorized into three aspects: user interface, process logic, and data storage. From the perspective of e-learning [33], technologies include Authoring Tool, Learning Management System (LMS), and Repository. Broadly speaking, websites for online learning with functionalities of administration, course management, and assessment could be seen as an LMS. Authoring tool is used for the creation and management of courseware and metadata. And an open database that provides data storage, searching, delivery, and exchange functions is called a repository. A repository in distance learning not only provides a distributed storage mechanism but also emphasizes on the shareability and reusability of Learning Objects (LOs) [17], [35], [39]. Although the issues of common repository for online learning were addressed [18], [34], [40], representation of LOs is another key issue that affects the design of repository architecture [41], [45].

The Advanced Distributed Learning (ADL) initiated a distance learning de facto standard named Sharable Content Object Reference Model (SCORM). SCORM includes the Content Aggregation Model (CAM) to enhance the reusability of courseware, Runtime Environment (RTE) to support learning activities, and Sequence and Navigation (SN) to provide a series of sequence rules for adaptive learning purposes. SCORM-compliant LOs can easily achieve shareability and reusability through the use of Learning Object Metadata (LOM) [20] described in CAM. With the increase of LOs (or training materials), a repository was developed in the Department of Defense (DoD) of the US military. Meanwhile, ADL with other research organizations proposed a common framework named Content Object Repository Discovery and Registration/Resolution Architecture (CORDRA) to enhance the interoperability and data exchange among distributed repositories.

To ensure reusability, the metadata (i.e., IEEE LOM) have been utilized to help LO discovery [22], [44]. To
enhance the annotations in a traditional metadata format, ontological approaches [15], [43] were proposed to achieve efficient classification of LOs and assist the users in searching probable LOs. As a consequence and general requirement, an LO not only needs a common representation of metadata, but also a storage infrastructure to allow public searching, which had brought an important open issue for e-learning research [13], [47].

In earlier work [26], Lin et al. proposed a Metadata Wizard framework for automatic metadata generation. They utilize course creators’ work and fill the missing parts of metadata, which, in turn, can enhance the searchability. After that, as a significant extension, Shih et al. [36] go further to find out the relations among different LOs and assist users in creating courseware by providing necessary information. As a short summary, these articles followed SCORM to develop a repository system (i.e., the MINE Registry), and provided a search mechanism with a series of search criteria for users to obtain LOs.

In this paper, based on a systematic reexamination of reuse scenarios from users, a series of metrics have been proposed to enhance the reusability of LOs. This work aims at providing a way to reduce the efforts users may have to make while accessing the repository for searching LOs that may possibly satisfy their needs. The overall scenario is that when an LO has been registered in our repository, system will calculate the importance degree (or weight) of it based on its implicit information (e.g., the creator, duration time, citations, etc.). Then, the LOs can be ranked in a specific order through the cross calculation of their weights and the relevance between LOs based on users’ input query. After obtaining this information, the system will also provide possible suggestions to users for assisting them in revising their original query. For this reason, we name the process flow “Guided Search.”

The organization of this paper is as follows: After describing our motivation and contribution, Section 2 will give a brief introduction to the related technologies we adopt. The core methodologies including weighting, ranking, and guiding formulae are discussed in Section 3. Section 4 shows the evaluation results for the proposed methodologies. We conclude this work and address our future works in Section 5. In the appendix, we demonstrate example of practical usages to illustrate how our repository works, how the search process represents, and how to make use of the service.

1.1 Motivation and Our Contributions

Although SCORM and CORDRA provide preliminary solutions for searching and reusing LOs, a few important points were missing. For instance, CORDRA pays more attention on sharing and discovering LOs as compared with SCORM. After LOs were created and registered to the repository, they might be reused by other users. In this situation, the external information, such as citations or persisted time period, may change the significance of LOs. A huge amount of past LOs may increase the difficulty to discover, and thus, to share more recent and useful LOs. For instance, the training courseware with topic “Grid Computing” may have lower priority than the one which instructs “Cloud Computing” recently, even though the amount of former LOs is much greater than the later ones. Our contribution has two main parts. First, we utilize the external information of LOs to provide an efficient mechanism for evaluating LOs based on the CORDRA framework. We give each LO a weight and calculate the relevance between LOs. Furthermore, we proposed a set of search guidance rules to assist users in finding relevant LOs by providing them necessary suggestions. The search process we proposed is different from ones in other repositories. We emphasize on assisting users to revise original search criteria rather than asking them to input blind or misleading queries caused by other irrelevant results.

2 BACKGROUND TECHNOLOGIES

We will give a brief introduction to the related technologies and methodologies including the IEEE LOM and CORDRA, Reusability Tree, Similarity Calculation Methodologies, and Data Mining Technologies that we used in this research.

2.1 IEEE LOM and CORDRA

The IEEE Learning Technology Standard Committee (LTSC) proposed a five layered architecture to describe the possible information for available LOs. In the third level, the architecture focuses on the precise definition of system components and the related repositories. The committee also introduced LOM (IEEE 1484.12.1-2002 LOM v1.0) to provide a unified description (or metadata) of learning resources. By using LOM, the LOs can be retrieved and acquired easily and precisely among various e-learning systems. The LOM is now serving as the principal standard internationally to specify LOs. In this research, we utilize LOM as the basis to calculate the relevance between LOs. LOM mainly composed of nine categories as follows: General, Life Cycle, Meta-Metadata, Technical, Education, Rights, Relation, Annotation, and Classification, to annotate LOs in a comprehensive manner. Each category has its own subcategories and specific vocabularies to describe LOs in details.

CORDRA is “an open, standards-based model for how to design and implement software systems for the purposes of discovery, sharing, and reuse of learning content through establishment of interoperable federations of learning content repositories” [2]. Its architecture aims to provide a way to resolve the conflict in the name space by means of a unique handler for each LO. It also provides a way to allow discovery and sharing of LOs. However, relations among reusable LOs and the history of using these LOs are not maintained. As a consequence, if a course creator obtains a large number of LOs in a particular search, he/she needs to look at them one by one to find out their relations and the usage history. This tedious process will discourage reusing LOs. A similar situation occurs when utilizing an ordinary Internet search engine, if the large amounts of search results are not properly organized for the user.

2.2 Reusability Tree

The reusability tree is conceptually similar to a version derivation tree. It consists of nodes and links, where a node at one level is a parent LO, and child nodes of the parent node represent LOs created by reusing. A child LO thus contains properties copied from its parent LO and its own properties. When reusing an LO, several types of changes
may be made,\textsuperscript{1} and the changes are captured in the reusability tree. Taking Fig. 1 as an example, there are four different LOs in this scenario. LO\textsubscript{1} represents the original learning object with three nodes (i.e., N\textsubscript{1}, N\textsubscript{2}, N\textsubscript{3}). LO\textsubscript{2}, LO\textsubscript{3}, and LO\textsubscript{4} are created by modifying parts of LO\textsubscript{1}. The new learning objects LO\textsubscript{2}, LO\textsubscript{3}, and LO\textsubscript{4} can be considered as the nodes in the derivation tree from the LO\textsubscript{1}. As an example, it is not difficult to find that LO\textsubscript{4} has a higher similarity with LO\textsubscript{1} (the original LO), as compared with others. The similarity degree can assist users to get the relevant information of LOs and reuse them.

Fig. 1. An example of reusability tree.

2.3 Similarity Calculation Methodologies
In the literature of Information Retrieval (IR), a common method is to use keywords which creators select as the basic attribute of specific documents. Lots of mechanisms \cite{15,17} are utilized to assist systems in classifying the documents into several groups, and can help users to find relevant resources. Obviously, the document in the same group will have a higher similarity. Example methods to calculate similarity degrees include “Simple Matching Coefficient,” the “Jaccard Coefficient,” and others \cite{18}. These methods calculate similarity between different documents mainly through keywords. However, inaccuracy may occur while a huge amount of keywords are set in specific documents. In distance learning, the metadata (LOM) contain lots of categories and their corresponding values. These values can be regarded as the elements for us to calculate the similarity between LOs. In this situation, a similarity calculation method not influenced by the numbers of keywords is needed. The Cosine Coefficient which focuses on the angle between vectors regarded as the set of keywords will be the optimal choice for us to utilize.

2.4 Data Mining Technologies
The time series problems \cite{16} are considered in this research. And in this case, we will introduce some related data/web mining technologies widely used as the bases of our proposed mechanisms.

2.4.1 The Google PageRank
Larry Page and Sergey Brin \cite{12,24} consider the popularity of a website that depends on internal/external links. They propose an algorithm to calculate a numerical weighting, called PageRank (\textit{PR}), to each element of a hyperlinked set of documents. The formula is as follows:

\[
PR(T_i) = \left(1 - d\right) + d \sum_{l=1}^{n} \left[ \frac{PR(T_l)}{L(T_l)} \right],
\]

where the \(PR(T_i)\) represents the PageRank of page \(T_i\) which has links \(L(T_i)\) to other \(n\) pages. The variable \(d\) with the default value 0.85 is the probability coefficient of specific webpage.

2.4.2 Mining Methodologies for Time Series Data
Landmark \cite{30} represents the beginning of a system time. In this model, data will be processed from the time a system starts until the present time. Due to the consistent speed of time, we can easily analyze and compare the process history in the time model. The disadvantage is that it will cause extreme workload to systems when over time processing occurs. Besides, not all of data in each timescale are useful. We need to take garbage data processing into consideration when utilizing this model. The Sliding Window Model \cite{27} improves the disadvantages of the Landmark Model. It helps analyzing the data stream in a specific timescale. Each sliding window contains a fixed width of data elements. The data will be loaded and processed in a specific timescale ahead of the current time. After that, data elements are implicitly deleted from the specific sliding window, when it moves out of the window scope. However, the use of such a model focuses on the basis of timescale and must be adjusted based on different conditions and circumstances.

The data stream is considered as the same in each timescale in the models presented. To represent the

\textsuperscript{1} All the changes that may be made will be recorded by the Hard SCORM authoring tool \cite{37} while reusing the specific LOs.
importance of data in each timescale clearly, the Time-Fading Model [7] was proposed. As such, the mentioned distance of time is also a key point of data mining. It separates the time into several blocks and gives each timescale a different decreasing weight progressively, from the current to the past. It improves the relationship between data and timescale, especially to those timeliness data. Taking Fig. 2 as an example, the data in the later timescale will have higher weights than the past ones.

Using the presented models for mining time series data, the quantity of data has to be concerned because of the memory size. In Landmark Model, it may take

\[ 60 \text{minutes} \times 24 \text{hours} \times 31 \text{days} = 44,640 \text{units} \]

to record the data in one month with the smallest measurement unit: 1 minute. In order to overcome this storage problem, the Tilted-Time Window [9] was proposed. The timescale is divided into different sections from the nearest one to the farthest one. The nearer sections will be given in more details, as shown in Fig. 3.

With the same example, the total memory costs will be:

\[ 60 \text{minutes} + 24 \text{hours} + 31 \text{days} = 115 \text{units}. \]

In this research, we pay emphasis on the focus-to-date LOs, and to estimate their importance degree by using the mining methodologies, especially the Time-Fading Model and the Tilted-Time Window Model presented above. The LOs over a long period of time will be considered in a macroperspective through the integration of these methodologies.

3 THE PROPOSED MECHANISMS

As described in the previous sections, the basic concept of our proposed mechanisms is applied to assist users in retrieving useful LOs in repository system. Three steps are conceived to achieve this goal. First, the weighting metrics, as the basis of this research, will be discussed in Section 3.1 on how to assign different important values to the LOs according to the time series. Second, as all search systems do, our ranking metrics are utilized to provide the retrieved results in a specific order in accordance with users’ query and the weights that LOs have. The details of our ranking metrics will be described in Section 3.2. Finally, Section 3.3 will show how our search guidance mechanism works to give possible query revision suggestions to users’ original query. The suggestions will also be affected by the weight/rank values the LOs have. Furthermore, the concrete instances of these metrics and mechanism are also provided in the respective sections.

3.1 The Weighting Metrics for LOs

It is meaningful to use metadata of LOs in their life cycle to rank and recommend LOs, as described in [28]. In addition, it is possible to improve user experiences through metadata calculation. The incidence graph of the LOs is used to find out the relations among LOs.

We propose a mechanism for weighting and ranking LOs by recording the citation impacts from users (i.e., users of authoring tool and users of LMS). In this research, the citations can be considered as the download frequency of specific LOs. Thus, for learning object \((LO_i)\), we have the Citation Reference \((CR(LO_i))\) from the MINE Registry system [10], and the value will be a nonnegative integer. According to this citation value, we can realize the importance degree of an LO. The higher citation value the LO has, the more popular it is. Then, we further use the following methods:

- **Author reference** \((AR)\): System will collect citations of LOs created by same authors and sum them up. We could keep track of the status of LOs according to the relationship between authors and LOs through how many times the specific LO is downloaded. The citations of a newly created LO are defaultly set to zero.

- **Time reference** \((TR)\): It represents the number of citations in a specific timescale. If the citation of a specific LO increases suddenly in a timescale but it is only utilized just a few times in the following days, it may not be evaluated simply through \(AR\). We have to consider the citations in a specific timescale that LO has. Hence, \(TR\) can be utilized to record the time LO persisted and its corresponding citations to improve the accuracy of the weight of LO.

In accordance with the parameters \((CR, AR, and TR)\) discussed above, we can give weights to the LOs in our system (and other LOs imported). It is similar to any search mechanism (e.g., Google search engine), which utilizes some rules to make search results accurate. We make use of the three parameters above with different thresholds, according to the following formula:

\[
Ref(LO_i) = \alpha \cdot CR(LO_i) + \beta \cdot AR(LO_i) + \gamma \cdot TR(LO_i),
\]

where \(\alpha + \beta + \gamma = 1\).

(2)

However, (2) has two problems: 1) The value of \(CR(LO_i)\) may be extremely high (e.g., 9,999) or extremely small (e.g.,
1) without normalized measurement. 2) We have to withdraw some old data according to the TR. But we also have to modify CR and AR according to the changes of TR. The revised formula is shown as follows:

\[
\text{Let } CR(T_s, LO_i) \subseteq CR(LO), \\
AR(T_s, LO_i) \subseteq AR(LO), \\
\text{where } |T_s| \geq 0; \\
Ref(LO_i) = \alpha \frac{CR(T_s, LO_i)}{CR(LO)} + \beta \frac{AR(T_s, LO_i)}{AR(LO)},
\]

where \(\alpha + \beta = 1\),

where \(CR(T_s, LO_i)\) represents the citations for a specific learning object \(i\) in a selected timescale \(TS\). The measurement unit for \(TS\) can be assumed to be a year, and set by an administrator or instructor. \(AR(T_s, LO_i)\) stands for the author references in \(TS\). With normalization, (3) can provide a preliminary solution which may be caused by problem 1) mentioned above. There is no evidence showing that the citations in certain years are, in fact, within the optimal timescale. In this research, we use three years as the default setting.

To normalize the proposed formula, it is reasonable to revise (3) by integrating the Tilted-Time Window Model and separating specific timescale in different length. We separate time into units like “half day (12 hours),” “one day (24 hours),” “one month (31 days),” “one year (12 months),” and “10 years” as shown in Fig. 4.

After that, we integrate the Time-Fading Model to calculate the weights of LOs as follows:

\[
W(LO_i) = \frac{D_i - j + 1}{D_j},
\]

where \(W(LO_i)\) represents the weight of \(LO_i\) in a specific timescale \(Di\), and \(n\) is the number of exact time units we set for obtaining different weight of a specific \(LO_i\). For instance, the weight of \(LO_i\) in \(D_2\) will be \(W(LO_i)\), and the weight of \(LO_i\) in \(D_1\) will be \(W(LO_i)\).

According to Time-Fading Model, the weight of the latest information shall be the greatest. Thus, as shown in Fig. 4, the weight of the smallest unit (one to the right side in each section) will be greater than or equal to the sum of previous ones. Similarly, the weight of the latest LO will be set to the largest. Thus, we have to change the order of the sections (i.e., parameters of \(n\)) in (4). For instance, if we have three sections (“half day,” “one day,” and “one month”), the \(n\) shall be “3,” “2,” and “1,” respectively.

According to the citations and weights in a specific timescale, we can define the following formula for the weight of LOs:

\[
Ref(LO_i) = \alpha \frac{\sum W(LO_i)_j \cdot CR(T_s, LO_i)}{CR(LO)} + \beta \frac{\sum W(LO_i)_j \cdot AR(T_s, LO_i)}{AR(LO)},
\]

where \(\alpha + \beta + \gamma = 1\).

According to the calculation for \(LO_i\), in (5), the parameters are retrieved based on the citation numbers provided by our repository. Inspired by the concept in Web 2.0 and social network, we also take end-user evaluation into account. This strategy is similar to the evaluation mechanism used in the YouTube and the Google Social Search [42]. Thus, we add a social evaluation mechanism, the response feedbacks for a specific learning object \(FB(LO_i)\), into our repository to collect subjective opinions from users. The time series problem is also considered. Therefore, we revise our formula to the following:

\[
Ref(LO_i) = \alpha \frac{\sum W(LO_i)_j \cdot CR(T_s, LO_i)}{CR(LO)} + \beta \frac{\sum W(LO_i)_j \cdot AR(T_s, LO_i)}{AR(LO)} + \gamma \cdot FB(LO_i),
\]

where \(\alpha + \beta + \gamma = 1\).

The feedbacks in a different timescale, \(FB(T_s, LO_i)\), are also considered in (6). Users will be asked to leave feedbacks to the results computed by their input queries. After continuous calculation, the feedbacks of \(LO_i\) can be categorized between two results (Negative and Positive) from previous users. Specifically, we separate the feedback information into five categories and give each of them a different relevancy coefficient. The instance of our interface for collecting the feedbacks of LOs is shown in the Appendix. In our repository, the algorithm we utilize to calculate the feedback from users is shown in Fig. 5.

We utilize a temporary threshold \(R_t\) to represent the search results with three variables \((Rv, Po, Ne)\) to examine the accuracy of the feedbacks for \(LO_i\) in specific timescale \(TS\). \(Po\) stands for the positive feedbacks the \(LO_i\) has. On the contrary, \(Ne\) is for the negative feedbacks. The variable \(Rv\) is the current scores \(LO_i\) has. Two thresholds \((Po, Ne)\) are used to normalize the polarization feedbacks which might cause the imbalance. The value of \(Rv\) will be mapped to \(FB(T_s, LO_i)\).

For instance, Fig. 6 shows the citations in the life cycle of \(LO_i\). The age of \(LO_i\) is 2 month and 6.5 days (68.5 days in total). The citations are 650 times in two month, 250 times in six days, and 100 times in the last half day. In this situation, we can obtain:

\[
D_1 = 0.5, D_2 = 6, D_3 = 62, n = 3.
\]

According to (4), we can obtain the weight of each timescale as follows:

\[
W_1 = \frac{D_1}{\sum D_i} = \frac{D_2 - 1 + 1}{D_1 + D_2 + D_3} = \frac{62}{68.5} = 0.91.
\]
Similarly, we calculate the values of $W_2$ and $W_3$, and can obtain 0.08 and 0.01, respectively. In this example, $LO_1$ is the only LO created by a specific author, the $\beta$ shall be 0. Another key point to obtain the weight of $LO_1$ is the feedback from past users. In accordance with the record from our repository, there are 375 feedbacks in recent 68.5 days where there are totally 500 feedbacks related to $LO_1$, which has been created for a year (365 days). Thus, the selected computation timescale for $LO_1$ will be 0.19 (68.5 days in 365 days). The calculation process is shown in Fig. 7.

We calculate the citations of LOs in a certain timescale. But it does not mean that selected timescale will be the most appropriate one. System administrators or instructors can change it to meet different scenario’s situations.

### 3.2 The Ranking Mechanism

After obtaining the weight of LOs, inspired by the concept of Google PageRank algorithm, we try to rank the LOs in our repository.

As shown in Fig. 1, it is obvious that three LOs ($LO_2$, $LO_3$, $LO_1$) have certain relations (or similarity) with $LO_1$. In the e-learning field, the most efficient way to compare LOs is to utilize the metadata known as LOM to assist users in searching for a specific LO. But in LOM, not all of the elements are feasible for similarity comparison. Thus, we only select nine representative elements [1], [3], [29] in two main categories to strengthen the educational usage:

- **General**: Title, Language, Keyword, Coverage.
- **Educational**: LearningResourceType, IntendedUserRole, TypicalAgeRange, Difficulty, TypicalLearningTime.

As shown in Fig. 8, we utilize the Cosine Similarity Coefficient as the major calculation methodology and make use of the selected LOM elements ($SE$) to adjust the original similarity between LOs (based on the existence of selected elements ($SE$)). The results emphasize more on the educational usage, because lots of these selected elements are required fields in the IEEE LOM specification. It may reduce the difficulty while calculating the similarity when there are less empty fields in their metadata.

### The Weight of $LO_1$

1. The citations and frequency:
   - **Period** | **Citations**
     - 0.5 days | 100
     - 6 days | 250
     - 62 days (2 months) | 650

2. Weight value for each section:
   - **Period** | **Weight**
     - $W_1$ (0.5 days) | 0.91
     - $W_2$ (6 days) | 0.08
     - $W_3$ (62 days) | 0.01

3. Threshold for our proposed formula:
   - $\beta = 0$
   - Assumed that $\alpha = 0.5$ and $\gamma = 0.5$

4. The feedback information of $LO_1$ will be:
   - $FB(0.19, LO_1) = 375$
   - $FB(1, LO_1) = 500$

5. Placing the thresholds and parameters into our formula

$$Ref(LO_1) = 0.5 \cdot (0.91 \cdot 100 + 0.08 \cdot 250 + 0.01 \cdot 650)/(100 + 250 + 650) + 0 + 0.5 \cdot 375/500$$

Therefore, the final weight of $LO_1$ is 0.43375.

---

**Fig. 7.** The calculation process for obtaining the weight of $LO_1$.

---

**Fig. 8.** The algorithm for obtaining the similarity between two LOs.

---

**Fig. 6.** The life cycle of $LO_1$.  

**Fig. 5.** The algorithm for calculating the feedbacks.
According to (6) and the similarity algorithm, the ranking formula for LOs in this paper is proposed as follows:

$$\text{Rank}(\text{LO}_i) = \text{Ref}(\text{LO}_i) + \frac{\sum \text{Sim}(\text{LO}_i, \text{LO}_j) \cdot \text{Ref}(\text{LO}_j)}{n}.$$  \hspace{1cm} (7)

Equation (6) shows how to weight LOs. After getting the weight of each LO, we rank these LOs through (7). The main purpose is to assist users in retrieving relevant LOs when they use the search service provided in our repository. The list of search results will appear with relevant degree. In this situation, the learners can get the relevant resources to learn; and, on the other hand, the authors can spend less time on finding resources to help creating learning materials.

We take two LOs in Fig. 1 as examples. Their metadata are shown in Tables 1 and 2. We assume that LO2 is a derivated learning object from LO1. Thus, LO2 has certain relation with LO1. In this example, we do not list all of the metadata description of specific LOs. We only list the necessary elements selected.

In the Information Retrieval literature, researchers utilize the Term Frequency-Inverse Document Frequency (TF-IDF) \cite{4, 48} to get the weight of test data. The weight for a term often indicates the frequency that the query terms used. The weight also shows the relevant degree between query terms and the test data. However, we do not utilize TF-IDF as the methodology for full text scanning in e-learning content. The purpose of LOM is for searching. Thus, we have to transform the query terms into index array and compare the metadata to obtain similarity values. The detail process for similarity calculation is shown in Fig. 9.

After obtaining the similarity between LOs, we calculate the ranking values. Since LO2 was derived from LO1, LO2 should return part of its weight back to LO1 because of the hierarchical relationships between LOs. To rank these LOs, our first step is to calculate the weight of LO1 and LO2, as shown in Fig. 11.

In Fig. 10, the sub-LOs of LO2 are not taken into consideration, and thus, the order will be: LO1 $\rightarrow$ LO2 (in descending order). However, two sub-LOs (LOg, LO7) are derived from LO2, and the similarities of them are 0.5916 and 0.6429.

**Similarity between LO1 and LO2:**

Assuming the weight of each index term is “1”, and the selected elements for matching are:

**General:** title, language, keyword, coverage

**Educational:** learningresourcetype, typicalagerange, difficulty, typicalearningtime

1. The similarity in Category “General” will be:

$$\text{Sim}_{\text{General}} = \frac{1}{3} + \frac{1}{4} \cdot \frac{3}{4} = 0.7708$$

2. According to the same algorithm, we can also obtain the similarity in “Lifecycle”, “metadata”, “Technical”, “Educational”, “Rights”, “Relation”, “Annotation”, and “Classification”, which are “0”, “0”, “1”, “0.5667”, “0”, “0”, “0.5556”, “0”, respectively.

By this algorithm, some of the elements will be filled in by string such as <size>, <date>, etc. We also set up a tolerance range for calculating their similarity. According to the above, \text{Sim}(\text{LO}_1, \text{LO}_2) will be 0.6429.

Fig. 9. The similarity between LO1 and LO2.
0.423. The weights of LO6 and LO7 are 0.40833 and 0.31719, respectively. In this situation, the rank value of LO2 will be 0.63908, and the details are shown in Fig. 11.

In the revised example, it is obvious that LO2 has a higher ranking value than LO1. Thus, the order of search results will be: LO2 > LO1 > LO6 > LO7 (in descending order). Generally speaking, the ranking value of LOs will be dynamically changed based on their weights and the sub-LOs they have.

According to our methodologies, both weights and ranking values can help representing the significant degree of retrieved learning objects. These methodologies help users (both learners and authors) to find what they want. However, it is time-consuming to go through any searching process if no guidance is utilized. Thus, in the next section, we will introduce the proposed search guidance mechanism to enhance search efficiency.

3.3 The Search Guidance Mechanism

Guided Search could be considered as a derivation path according to the experience that a user utilized the search service in a federated repository. The search space consists of nodes and links, where the node represents the course unit (or LO) and the link represents the relationship between the parent node and child node. In order to find the relevant LOs in a repository, the use of similarity is necessary.

To find the similarity between LOs, we have selected the most commonly used elements (nine elements) from LOM (77 elements in total). Through these selected elements, we can find out the relationship between the one searched by the users and the one stored in our repository.

Fig. 12 shows an example of search scenario. The red box represents the initial query. The green box represents the revised query based on the original query. The revised query will retrieve what the user wants. The key issue is to guide the initial query toward the revised query in a fast way.

To achieve our goal, we revised the Relevance Feedback algorithm [31], [32], [46] and integrated it with our proposed mechanism. The revised algorithm is shown in Fig. 13.

The process flow can mainly be separated into the following steps:

1. A user follows the search criteria selected to find relevant LOs in the repository. The search criteria we set can be classified into following groups: Precise Criteria, Incremental Criteria, Precedence Criteria, Time/Duration Criteria, and Single/Multiple Choice Criteria, which have already addressed in [36].

2. An alternative way is to allow users to input one or several keywords to start the first query. The query criteria ($C_i$) here include the definition of LOM (keywords, language, difficulty, etc.) The system will generate the query vector ($Q_m$) to proceed the
queries and return the relevant results \( D_m \), shown in a reusability tree to users.

3. We calculate the similarity of each search result and take the intersection to generate a representative set. This set is called positive union \( Z_p \). After that, we compare the elements in \( Z_p \) with the first-query result lists. Thus, we can retrieve the irrelevant elements. They are called negative union \( Z_n \).

4. Then, we make use of the weighting mechanism we proposed and the diversity match function in previous work [25] to calculate the suggestion coefficient \( S_{coe} \) and the irrelevant degree \( S_{roe} \). The system will reset \( Z_p \) based on \( S_{coe} \) to become a new suggestion list \( Z_{pr} \). After that, we check the elements both in \( Z_{pr} \) and \( Z_n \) to see if there are any corresponding elements.

5. To filter out the irrelevant query results [8], we select 10 elements in descending order from \( Z_{pr} \) and check them with \( Z_n \). Then, we add them into the suggested revised query vector \( Q_{r} \).

Our goal is to utilize users’ operation, with our repeated suggestions, and prune off several irrelevant results. For example, a user inputs the keywords “Data Structure” and the system can return the search results to him/her. After that, the system will follow the algorithm to create the suggestion lists. Suppose that the system returns only one suggestion with an index term “Hashing.” Then, the suggested revised query will be “Data Structure + Hashing.” The user can choose them by following the suggestion or to input a new query criterion.

The steps listed above will be in a loop with interaction with users. It will continue until users stop the query process or start a new query process. We show an example of the revision process for a user who was a tester in our experiment introduced in Section 4.3. The detail process is shown in Table 3.

In this example, user inputs the query three times and obtained the target LOs we asked him to search. The symbol “+” represents the query condition “and,” and the “j” stands for the condition “or.” In “Retrieved Objects,” the retrieved results are represented in the view of reusability tree, and the amounts of LOs are also shown to users (i.e., 5,199 retrieved results were represented in 84 reusability trees). In “Revised Query,” we only list two major factors that are “keywords” and “languages.” The similarity degree calculated through our proposed algorithm will affect the order of suggested item in specific factors.

### 3.4 The Comparisons

In the literature, developers of e-learning systems mainly pay emphasis on authoring tools and learning platforms (LMSs). Relatively, few articles addressed repository and searching issues. However, in the e-commerce literature, recommendation systems provide efficient ways for...
customers to search for products. The same mechanisms can also be adopted in distance-learning-related systems. In this paper, we provide a novel mechanism for repository to calculate the significant degree of LOs. In addition, we also provide a set of guidance algorithms to assist users in retrieving relevant information by revising their queries. As shown in Table 4, we compare several existing techniques with our proposed approach. We mainly focus on the difference of strategies and mechanisms to show our unique contribution.

### 4 System Evaluation

We separate the evaluation process into three parts: 1) to assess the overall performance of the MINE Registry system; 2) to compare the performance with other acknowledged methods; and 3) to evaluate search guidance.

#### 4.1 Overall Performance of MINE Registry System

The MINE Registry, up to the date of this experiment, has around 21,000 LOs with unique IDs obtained from the CNRI system, and these LOs can be accessed by external repositories. Most of the LOs in our repository were released and collected from ADL and universities. And these LOs have been utilized for a specific training program. We conduct the system evaluation to assess the overall performance of our repository system by obtaining the accuracy of the search results. We choose three topics and their corresponding queries (using only keywords, for example) to perform this experiment. The first topic (T1) is in the scope of “Algorithm and Data Structure,” and the input query is “algorithm + ds.” The second one (T2) is about “Photoshop Introduction,” and the query is “photoshop + intro.” The third one (T3) is about “Multimedia Computing,” and its query is “mm + computing.” Before the evaluation, we define the accuracy functions [21] that will be used in this experiment as follows:

\[
\text{Precision}' = \frac{\text{Number of relevant LOs retrieved}}{\text{Number of relevant LOs}},
\]

\[
\text{Recall}' = \frac{\text{Number of relevant LOs retrieved}}{\text{Number of relevant LOs}},
\]

\[
F - \text{Score}' = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.
\]

Table 5 revealed the evaluation results of our repository. “Relevant LOs in Repository” represents the numbers of learning objects relevant to each target topic we observed before evaluation. “Retrieved LOs” represents the search results through selected queries, and “Relevant LOs Retrieved” stands for the search results that are in the scope of “Relevant LOs in Repository.” Since there are lots of derivated LOs in the repository, it may reduce the average precision which is about 85 percent. It is not the optimal results we expect. However, it is worth noting that the average recall value is about 94 percent. And this may prove that our approach (considering the time series problem to rank the LOs) can achieve the same performance as the one we used before.
4.2 Comparison with Related Approaches

We also conducted an experiment to compare our proposed approach with related works addressed in Section 3.3. We utilized the evaluation results obtained in the previous section. The precision-recall graph is shown in Fig. 14. And it is worthy to mention that the evaluation results between different approaches are close. However, our approach, especially after the integration of ranking mechanism and search guidance, still keeps the overall search performance. Furthermore, this evaluation also shows that the use of metadata as the basis of searching criteria, in some situation, will lead to a better result than the ontological approach which describes the LO and provides search criteria in a nonunited way.

4.3 Evaluation of Search Guidance

To prove the effectiveness of our proposed mechanisms, we conducted a pre-post-experiment assessing 40 users who utilized our repository to find useful LOs and create a specific courseware. In this experiment, the users included six professors and 34 Teaching Assistants (TAs) in both Tamkang University, Taiwan, and Waseda University, Japan. To get the objective results, we asked users to search for a specific LO which is the only one whose title is “Introduction to Photoshop” in the selected testing data set, and to calculate the time they spent and the query frequencies in both situations (with system’s guidance or without guidance). First, we made two substantial assumptions before the experiment:

- Assumption 1: The resource retrieval time in the posttest is not better than one in the pretest.
- Assumption 2: The resource retrieval time in the posttest is better than one in the pretest.

To verify our assumptions, we utilized an One-Tailed Test \( (\alpha = 0.05) \) and set the degree of freedom \( (df) \) at 39. Fig. 15 illustrates the pretest and posttest comparison and the average retrieval time of the two tests. In Fig. 15, the \( d \)-value, that is, the pretest score minus the posttest score of user identified as number 1, is 24 (i.e., 152-128). According to the \( t \)-distribution table, which assesses whether the means of two groups statistically differ from each other, \( df = 39 \) maps to \( t \) value 1.685. According to the formula, we obtained the value \( t = 10.098 \), which is greater than 1.685. Thus, Assumption 2 is acceptable and tells us that users who utilize our guidance mechanism will reduce the cost of time, around 40 seconds in this case, in looking for specific LO. The detail evaluation result is revealed in Table 6.

In addition, we also considered the query frequency users have while searching for the target LO. However, it is unobvious because the average frequency is 3.5, where the least is 2 and the most is 7, in this experiment.

5 Conclusions

The construction of a federated search and sharing architecture is important for distance learning. And it is
particularly important, in such architecture, to provide a mechanism that can assist course creators in finding LOs for reuse. In this paper, we enhanced our previous works, especially on the Reusability Tree, based on SCORM and CORDRA. First, we proposed to utilize data mining technologies for time series data to gather the relevant information, such as citations, of specific LOs in different timescales. That is, citations in different timescales represent different meaning of the LOs. We revised the Time-Fading Model and Tilt-Time Window Model to measure the weight of LOs. In addition, we provided a mechanism to rank these LOs. Utilizing the proposed mechanism, it can enhance the reusability of LOs. Furthermore, to assist users in the searching phrase, we revised the algorithm of Relevance Feedback and combined it with the weight of LOs that we proposed. We do not provide actual items, like the recommendation systems, to users. Instead, we provide suggestions that can guide them to revise the query process especially in the input query terms. We believe that with the proposed mechanisms and the distance learning standard that focuses on describing LOs (i.e., IEEE LOM), LOs can be searched in an efficient way, which will help the promotion of SCORM and CORDRA specifications in the international community of distance learning.

As stated above, in this research, we proposed an evaluation mechanism based on the ranking metrics and a search guidance algorithm to make search process in current federated Learning Object Repository (LOR) more efficiently. Although we can benefit from some useful distance learning standards like IEEE LOM or system architecture like CORDRA, we still face some critical challenges as listed below:

- **Quantities of LOs.** Although there are totally 20,738 learning objects in our registry system collected in the past years. Some of them are updated LOs based on existing ones. The number of learning objects we have can be regarded as a large amount of data. However, we still cannot reach a good sampling coverage as compared to some IR research domains.
- **External Connection.** According to the definition of federated CORDRA, every subrepository will have certain connections with others. Actually, most of them only have relation with the central registry system which belonged to CNRI. The search performance among a large scale of federated repositories will be taken into consideration though they proceed efficiently in our current repository.

<table>
<thead>
<tr>
<th>The MINE Registry</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="MINE Registry Image" /></td>
<td>1. <strong>Registration / Language Selection</strong>&lt;br&gt;The user has to login to the system for utilizing the advanced functionalities such as upload/download learning objects, personalized search history, and search path analysis, etc.&lt;br&gt;2. <strong>Basic Search Toolbar</strong>&lt;br&gt;The user can utilize this toolbar to search learning objects only based on keywords.&lt;br&gt;3. <strong>Side Toolbar</strong>&lt;br&gt;The user can click on this toolbar to open the sitemap or other related services developed by our research group.&lt;br&gt;4. <strong>Search Service</strong>&lt;br&gt;The user can click this tab to change the settings, such as destination repository path or query criteria, for search services.&lt;br&gt;5. <strong>Advanced Search Service</strong>&lt;br&gt;This service allows users to search for learning objects based on IEEE LOM.&lt;br&gt;6. <strong>Query Criteria Selection</strong>&lt;br&gt;The users can choose the criteria and input corresponding values to search for LOs in the repository.</td>
</tr>
</tbody>
</table>

Fig. 16. Layout of the MINE Registry and the interface of Advanced Search.
7. Resources Type
The icon shows the type of resource. The first icon represents the format of course package (SCORM format or Common Cartridge format), and the second icon stands for the web link.

8. Reusability Tree View
The reusability tree shows the relation between those learning objects. The values in the angle brackets are the similarity value and the diversity value.

9. The Weight of LO
Here we show the weight of learning objects in the last three years based on our proposed methodologies.

10. Corresponding Functionalities
The user can click here for more functions like “View Details,” “Give Feedbacks,” or “Cite It.”

11. Search Guidance
As stated in this paper, the system will automatically generate several search suggestions according to users’ query results.

12. LO Information Toolbar
We provide the detail information to users. Our system also has a simple LMS Run-Time for users to view the content of learning objects.

13. LOM Description
The user can view the metadata description of a chosen learning object.

14. The LO’s structure
Here it shows the structure of learning objects. Users can click on an item to see its content.

- **Selected Timescale.** We have to set the timescale for evaluating the representative degree of specific learning objects. However, a variation of timescale may cause different results. It is hard to define a standard evaluation timescale. Our registry system has been developed for five years. In this research, we utilize three years as the default timescale. It is necessary to consider different time scales to make our results more precise according to the practical situations.

In the future, we will continue this work to find out the optimal solutions for the difficulties and the restrictions that we have mentioned above.
APPENDIX

THE SYSTEM DEMONSTRATION

This appendix shows a concrete example of our proposed works based on the example course which introduces the basic concept of Photoshop released from the Open University, United Kingdom (see Figs. 16 and 17). According to our previous work, the reusability tree keeps track of the association among the LOs. Our repository, also known as the MINE Registry, will record the citations for each LO. The demo LOs we adopt here have been modified and reused for several times. Besides, lots of these LOs have been stored in our repository for more than three years. In this example, we set the default calculation timescale to three years and utilize our proposed mechanisms to obtain the weight for each query LOs. Furthermore, the search guidance mechanism will provide necessary suggestions based on the users’ input queries.

This research is partially supported by the Institute for Information Industry, Taiwan. Under the cooperation provision, we are not allowed to publish the testing LOs without permission. Interested readers are welcome to visit our website at http://www.mine.tku.edu.tw for more detail information.

ACKNOWLEDGMENTS

The research is supported by the National Research Council, Taiwan, under the grant “A QTI-Based Authoring and Item Bank System (NSC 96-2520-S-152-002-MY3),” and is also partially supported by the Institute for Information Industry, Taiwan.

REFERENCES

Timothy K. Shih is a professor with the National Central University, Taiwan. He was the dean of the College of Computer Science, Asia University, Taiwan, and the department chair of the Computer Science and Information Engineering Department at Tamkang University, Taiwan. His current research interests include multimedia computing and distance learning. Dr. Shih has edited many books and published more than 440 papers and book chapters, and has participated in many international academic activities, including the organization of more than 60 international conferences. He was the founder and co-editor-in-chief of the International Journal of Distance Education Technologies, published by Idea Group Publishing, United States. He is an associate editor of the ACM Transactions on Internet Technology and an associate editor of the IEEE Transactions on Learning Technologies. Dr. Shih has received many research awards, including research awards from the National Science Council of Taiwan, the IIAS research award from Germany, the HSSS award from Greece, the Brandon Hall award from the United States, and several best paper awards from international conferences. He has been invited to give more than 30 keynote speeches and plenary talks in international conferences, as well as tutorials at IEEE ICME 2001 and 2006 and ACM Multimedia 2002 and 2007. Dr. Shih is a fellow of the Institution of Engineering and Technology (IET) and a senior member of the ACM and the IEEE. He also joined the Educational Activities Board of the IEEE Computer Society.

Louis R. Chao is a professor in the Department of Computer Science and Information Engineering at Tamkang University, Taiwan. In addition to being the founder of the Association of E-Learning, he is also the director of the Department of Computer Science, the chief of the Engineering Institute, the dean of academic affairs, and chancellor at Tamkang University. He has been involved with international conferences such as the International Conference on Computers in Education, the International Computer Symposium, and the National Computer Symposium, and established the International Journal of Information and Management during his time at the university. He is not only a pioneer of informationization and internationalization at Tamkang University, but also a pioneer of computer-assisted instruction. His research interests are in the fields of distance learning, networking and communication, information security, multimedia, neural networks, and fuzzy theory. He also has experience in enterprise organization, management, and tactics application.

Qun Jin is a tenured full professor in the Networked Information Systems Laboratory, Department of Human Informatics and Cognitive Sciences, Faculty of Human Sciences, Waseda University, Japan. He was engaged extensively in research work in computer science, information systems, and social and human informatics. He seeks to exploit the rich interdependence between theory and practice in his work with interdisciplinary and integrated approaches. His recent research interests include user-centric ubiquitous computing, sustainable and secure information environments, user modeling, behavior informatics, information search and recommendation, human-computer interaction, e-learning support, and computing for well-being. He is a member of the IIEE.