

# Studying Relations Between E-learning Resources to Improve the Quality of Searching and Recommendation

Ngoc Chan Nguyen, Azim Roussanaly, Anne Boyer

► **To cite this version:**

Ngoc Chan Nguyen, Azim Roussanaly, Anne Boyer. Studying Relations Between E-learning Resources to Improve the Quality of Searching and Recommendation. Proceedings of the 7th International Conference on Computer Supported Education, May 2015, Lisbon, Portugal. pp.119-129, <10.5220/0005454301190129>. <hal-01253030>

**HAL Id: hal-01253030**

**<https://hal.inria.fr/hal-01253030>**

Submitted on 8 Jan 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Studying Relations Between E-learning Resources to Improve the Quality of Searching and Recommendation

Nguyen Ngoc Chan, Azim Roussanaly, and Anne Boyer  
*Université de Lorraine, LORIA UMR 7503, Vandœuvre-lès-Nancy, France*  
{ngoc-chan.nguyen, azim.roussanaly, anne.boyer}@loria.fr

**Keywords:** Online Education, Learning Resource Recommendation, Searching, PageRank

**Abstract:** Searching and recommendation are basic functions that effectively assist learners to approach their favorite learning resources. Several searching and recommendation techniques in the Information Retrieval (IR) domain have been proposed to apply in the Technology Enhanced Learning (TEL) domain. However, few of them pay attention on particular properties of e-learning resources, which potentially improve the quality of searching and recommendation. In this paper, we propose an approach that studies relations between e-learning resources, which is a particular property existing in online educational systems, to support resource searching and recommendation. Concretely, we rank e-learning resources based on their relations by adapting the Google's PageRank algorithm. We integrate this ranking into a text-matching search engine to refine the search results. We also combine it with a content-based recommendation technique to compute the similarity between user profile and e-learning resources. Experimental results on a shared dataset showed the efficiency of our approach.

## 1 INTRODUCTION

Online education has been progressively developed since the first videos of lectures published on the Internet by the University of Tübingen in Germany in 1999 and the appearance of the MIT OpenCourseWare in 2002. Along with the maturity of online education, numerous learning resources have been continuously published. Numerous portals of digital learning resources have been set up, such as MERLOT, OER Commons, LRE For Schools, Academic Earth, Organic Edunet, OCW France, and so on. These portals provide a variety of resources in different types, disciplines and levels. They allow teachers to share their lectures and a large number of learners to study and consolidate their knowledge. However, the diversity of e-learning resources probably lose learners' time in searching for expected resources. Therefore, supported tools are indispensable to assist learners to approach their favorite resources.

Searching is a fundamental function that is available on any e-learning portal to allow learners to find resources. However, it is passive function, which is only activated when the learner proceeds a search. In addition, it simply executes a text-matching mechanism, which is not able to detect learner's interest to provide them more interesting resources. There-

fore, recommendation is considered as a necessarily complementary function of searching as it is able to recommend dynamically resources that are close to learner's interests (Manouselis et al., 2011). Consequently, many recommendation techniques in the recommender systems (RS) domain have been applied in the Technology Enhanced Learning (TEL) domain. For example, techniques based on collaborative filtering (Lemire et al., 2005; Tang and McCalla, 2005), content-based filtering (Khribi et al., 2009; Koutrika et al., 2008), user ratings examination (Drachler et al., 2009; Manouselis et al., 2007), association rules (Lemire et al., 2005; Shen and Shen, 2004) or user feedback (Janssen et al., 2007) have been proposed. Model-based techniques such as Bayesian model (Avancini and Straccia, 2005), Markov chain (Huang et al., 2009), resource ontologies (Nadolski et al., 2009; Shen and Shen, 2004) and hybrid approaches (Nadolski et al., 2009; Hummel et al., 2007) have also been considered.

As focus on adapting techniques in the RS domain to the TEL domain, existing approaches take into account properties that can be applied by RS techniques. Data such as text description of resources (Khribi et al., 2009), resource rating (Lemire et al., 2005) or historical usage (Tang and McCalla, 2005) have been exploited. Few of approaches analyze particular char-

acteristics that exist in the online education systems such as learning context, learner levels or relations between e-learning resources. In addition, most of existing systems still remain at a design or prototyping stage (Manouselis et al., 2011).

In this paper, we present an approach that studies relations between e-learning resources to support searching and recommendation. We analyze different kinds of resource relations, such as ‘one resource can be a part of another resource’, ‘one resource can include other resources’, ‘one resource can be associated to other resources’ and so on. We adapt the Google’s PageRank algorithm, which was developed for web page ranking and searching, to rank e-learning resources according to their relations. We propose to combine this ranking with the text matching to refine the search results. We also propose to associate it with recommendation techniques to adjust the similarity between user profile and e-learning resources.

By examining resource relations, our objective is three-fold: (i) to study the coherence between e-learning resources via different kinds of links, (ii) to investigate the impact of a particular characteristic that exists in a specific domain, which are resource relations in online education systems, on the quality of searching and recommendation and (iii) to demonstrate an efficient algorithm to enrich experiments in the TEL domain.

As the recommended resources tend to be close to the learners’ interest, our approach potentially improve the learners’ learning performance and motivation, which certainly encourage learners to continue studying. Obviously, the more precise the recommendations are, the more efficiently the learners study.

The paper is organized as follows. The next section presents the resource ranking. Section 3 shows the application of resource ranking on searching and recommendation. Experiments are presented in section 4. Related work is discussed in section 5 and we conclude our approach in section 6.

## 2 RESOURCE RANKING

This section elaborates how we rank e-learning resources based on their relations. First, we introduce shortly the Google’s PageRank algorithm (subsection 2.1). Then, we identify basic relations between e-learning resources (subsection 2.2). Finally, we present the resource ranking based on their relations (subsection 2.3).

### 2.1 Google’s Page Ranking

PageRank is the core algorithm of the Google search engine. They proposed to rank web pages based on their interconnection, i.e. hyperlink, then integrate this ranking into their search engine to filter the search results. The high-ranked pages are important pages and will be put on the top of the returned list (Brin and Page, 1998; Page et al., 1999). The idea of Google’s page ranking is as follows.

Let consider a corpus that has  $N$  web pages. Ranking of these  $N$  web pages is defined by a vector  $v^*$  in the  $N$ -dimensional space that satisfies:

$$Gv^* = v^* \quad (1)$$

where  $G$  is the Google matrix, which is defined as:

$$G = \frac{1-d}{N}S + dM \quad (2)$$

where  $0 \leq d \leq 1$  is the damping factor,  $S$  is the matrix with all entries equal to 1 and  $M$  is the transition matrix (Page et al., 1999; Wills, 2006).

$M$  is a Markov matrix that presents links between pages. Value of an element  $M_{[i,j]}$  is the weight of the link from page  $j^{th}$  to page  $i^{th}$ . If a page  $j$  has  $k$  outgoing links, each of them has a weight  $\frac{1}{k}$ . So, sum of all weights of any column in  $M$  is always equal to 1.

For example, Figure 1 presents 4 web pages in a corpus and their hyperlink.  $A$  has 2 out-going links to  $B$  and  $D$ , so each link has a weight of  $\frac{1}{2}$ . Similarly, each link from  $B$  has a weight of  $\frac{1}{3}$ , and so on. The matrix  $M$  on the right is the transition matrix of the given corpus. All elements of  $M$  are non-negative and sum of each column is 1.

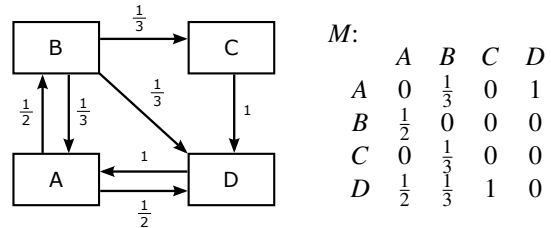


Figure 1: Example of hyperlink between pages

According to Eq. 2, as  $M$  is a Markov matrix and  $S$  is the matrix with all entries equal to 1, we easily prove that  $G$  is also Markov matrix. According to Eq. 1,  $v^*$  is the eigenvector of the Markov matrix  $G$  with the eigenvalue 1. Let  $v_0$  be the initial page rank vector. Elements in  $v_0$  are initialized by  $\frac{1}{N}$ .  $v^*$  is iteratively computed as following:

$$v_{i+1} = Gv_i \quad (3)$$

until  $|v_{i+1} - v_i| < \epsilon$  ( $\epsilon$  is a given threshold).

As  $G$  is a Markov matrix,  $v_{i+1}$  will converge to  $v^*$  after a finite number of iterations.  $v^*$  presents the ranking of web pages according to their hyperlink.

For example, in Figure. 1, if we initialize the page rank vector  $v$  of these web pages as:  $v_0 = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}$ , and calculate  $v_i$  by Eq. 3 with a threshold  $\epsilon = 0.01$ ,  $v$  will converge to the vector  $v^* = \{0.37, 0.20, 0.09, 0.34\}$  after 9 iterations.  $v^*$  present the ranking of the given pages. According to  $v^*$ ,  $A$  is the most important.

## 2.2 E-learning Resource Relations

E-learning resources are always described under a standard format, which allows presenting their metadata (i.e., title, abstract, keywords, disciplines, levels, etc.) under a well-structured form. Among a number of standards (such as Dublin Core, MPEG-7, MODS, and so on), the IEEE Learning Object Metadata (LOM) is dominant in use (Nilsson et al., 2006). According to the LOM standard<sup>1</sup>, a resource can have different kinds of relations with other resources. These kinds<sup>2</sup> include: *ispartof*, *haspart*, *isversionof*, *hasversion*, *isformatof*, *hasformat*, *references*, *isreferencedby*, *isbasedon*, *isbasisfor*, *requires* and *isrequiredby*.

Although the definition of resource relations was reported as a standard, there are still debates about its appropriateness and missing relationships. For example, (Seeberg et al., 2000) argued that the defined relations mix content-based and conceptual connections between resources, the 'requires/isrequiredby' is inappropriate, or there is no difference between a 'isbasedon' relation and a 'isrequired' one and so on. (Engelhardt et al., 2006) proposed additional relations, such as 'illustrates/isIllustratedBy', 'isLessSpecificThan/isMoreSpecificThan', etc., regarding to the semantic connections between learning objects. Or, the Open University of Humanities (UoH)<sup>3</sup> has defined a new relation, which is 'isassociatedto', in their LOM metadata to present the coherence between resources.

In our approach, we suppose that there are generally  $k$  kinds of resource relation, numbered from  $r_1$  to  $r_k$ , regardless their meanings. Each kind signifies a specific meaning of the relation. For example,  $a \xrightarrow{\text{ispartof}} b$  signifies that resource  $a$  is part of resource  $b$ , or  $a \xrightarrow{\text{hasversion}} a_1$  signifies that  $a_1$  is a version of  $a$ . Each kind of relation plays a role in presenting the co-

herence between two resources. It has a weight that indicates the importance of the corresponding relation. We suppose that a kind of relation  $r_i$  has a weight  $w_i, \forall i = 1..k$ . The concrete value of each weight can be flexibly assigned according to the set of kinds of relation used in a corpus.

For example, in the collection of resources published by the UoH<sup>4</sup>, there are only 3 kinds of relation (Figure. 2), which are: 'ispartof', 'haspart', and 'isassociatedto'. One can argue that: 'isassociatedto' indicates a set of coherent resources that supplement each other to present some knowledge, 'haspart' shows a set of resources to be involved within a subject and some of them are possibly not coherent, while 'ispartof' signifies a resource which is member of another resource but does not clearly present other related resources. Obviously, according to the coherence between resources, 'isassociatedto' should have greater weight than 'haspart' and 'haspart' should have greater weight than 'ispartof'. Consequently, he/she could assign the weights of these kinds of relation as 0.5, 0.3 and 0.2 respectively. However, this is just a specific observation. These weights can be varied according to other judgments. In our approach, we deal with relation weights regardless to their concrete values. Details of our resource ranking computation are presented in the next subsection.

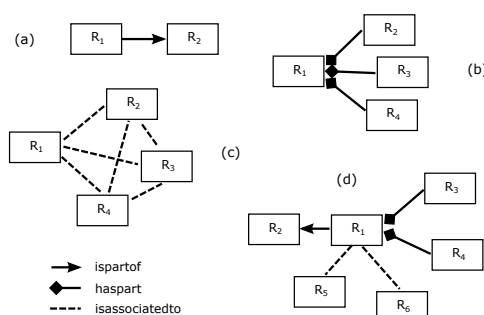


Figure 2: Example of relations between resources: a resource can be part of another resource (a), include other resources (b), associate to other resources (c) or have a mix of relations (d)

## 2.3 Resource Ranking

Inspired by the Google's PageRank algorithm, we propose an algorithm to compute the ranking of resources in a corpus. Instead of using the hyperlink, we take into account relations between resources, which are clearly defined in their metadata. This ranking presents the importance of each resource and can be

<sup>1</sup><http://standards.ieee.org/findstds/standard/1484.12.1-2002.html>

<sup>2</sup>based on the Dublin Core (<http://dublincore.org/documents/2012/06/14/dcmi-terms/>)

<sup>3</sup><http://www.uoh.fr/front>

<sup>4</sup>obtained in 06/2014

effectively used for resource filtering or recommendation.

The key step of our algorithm is presenting the resource relation by a transition matrix, which allows computing the Google matrix (Eq. 2) and resource ranking (Eq. 3). The transition matrix must be a Markov matrix, in which all elements are non-negative and sum of each column (or row) is 1. In our approach, we present each resource as a column and normalize relation weights so that the Markov matrix's conditions are satisfied. Details of our computation are as follows.

Consider a corpus that has  $N$  resources  $R_1, R_2, \dots, R_N$  and  $k$  kinds of relations  $r_1, r_2, \dots, r_k$ . Let  $w_t$  be the weight of the kind  $r_t$ ,  $1 \leq t \leq k$ .

Assume that a resource  $R_i$  has totally  $n_{i1}$  relations of the kind  $r_1$ ,  $n_{i2}$  relations of the kind  $r_2$ ,  $n_{i3}$  relations of the kind  $r_3$ , and so on ( $0 \leq n_{it} < N$ ,  $t = \overline{1..k}$ ). The weight of a relation kind  $r_t$  of  $R_i$ , denoted by  $w_{it}$ , is computed by Eq. 4.

$$w_{it} = \frac{w_t}{\sum_{t=1}^k n_{it} w_t} \quad (4)$$

Let  $M$  be the transition matrix. Element  $M_{[j,i]}$  presents the weight value of the relation from  $R_i$  to  $R_j$ . Assume that from  $R_i$  to  $R_j$ , there are  $n_{ij1}$  relations of the kind  $r_1$ ,  $n_{ij2}$  relations of the kind  $r_2$ ,  $n_{ij3}$  relations of the kind  $r_3$  and so on ( $0 \leq n_{ijt} < N$ ,  $t = \overline{1..k}$ ). The weight of the relation from  $R_i$  to  $R_j$  is given by Eq. 5.

$$M_{[j,i]} = \sum_{t=1}^k n_{ijt} w_{it} \quad (5)$$

where  $w_{it}$  is the weight of the relation kind  $r_t$  from  $R_i$ .

So, from Eq. 4 and Eq. 5, the sum of all elements in the  $i^{th}$  column of the matrix  $M$  is:

$$\begin{aligned} \sum_{j=1}^N M_{[j,i]} &= \sum_{j=1}^N \sum_{t=1}^k n_{ijt} w_{it} = \frac{\sum_{j=1}^N \sum_{t=1}^k n_{ijt} w_t}{\sum_{t=1}^k n_{it} w_t} \\ &= \frac{\sum_{t=1}^k \sum_{j=1}^N w_t n_{ijt}}{\sum_{t=1}^k n_{it} w_t} = \frac{\sum_{t=1}^k (w_t \sum_{j=1}^N n_{ijt})}{\sum_{t=1}^k n_{it} w_t} \quad (6) \end{aligned}$$

On the other hand, as  $n_{it}$  is the number of relations of the kind  $r_t$  from the resource  $R_i$ , we have:

$$n_{it} = \sum_{j=1}^N n_{ijt} \quad (7)$$

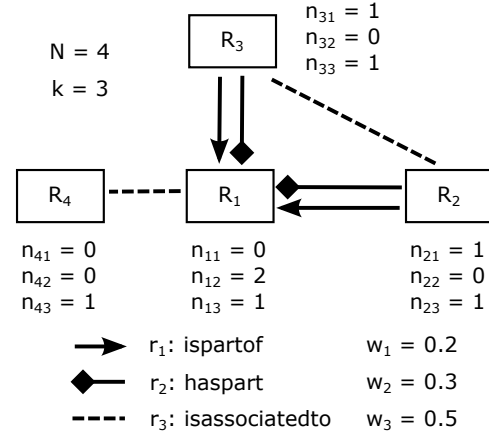


Figure 3: Example of resource relations in a corpus

From Eq. 6 and Eq. 7, we have:

$$\sum_{j=1}^N M_{[j,i]} = \frac{\sum_{t=1}^k w_t n_{it}}{\sum_{t=1}^k n_{it} w_t} = 1 \quad (8)$$

So, from Eq. 4, Eq. 5 and Eq. 8, we conclude that  $M$  is a Markov matrix and can be used by the Google's algorithm to rank resources.

For example, consider a corpus that has 4 resources and 3 kinds of relation. The relations between resources and their corresponding weights are presented in Figure 3. By applying Eq. 4, we have the weights of kinds of relation of these resources, which are given in Table 1.

Table 1: Weight of each kind of relations in Figure. 3

$R_1$ :	$w_{11} = 0.182$ ;	$w_{12} = 0.273$ ;	$w_{13} = 0.454$
$R_2$ :	$w_{21} = 0.286$ ;	$w_{22} = 0.429$ ;	$w_{23} = 0.714$
$R_3$ :	$w_{31} = 0.286$ ;	$w_{32} = 0.429$ ;	$w_{33} = 0.714$
$R_4$ :	$w_{41} = 0.4$ ;	$w_{42} = 0.6$ ;	$w_{43} = 1$

Then, we apply Eq. 5 to calculate the value of each element  $M_{[j,i]}$  in the transition matrix  $M$ . We get  $M$  as resulted in Table 2.  $M$  has only non-negative elements and sum of each column is 1. So,  $M$  is a Markov matrix.

Table 2: Transition matrix of the resources in Figure. 3

	$R_1$	$R_2$	$R_3$	$R_4$
$R_1$	0	0.286	0.286	1
$R_2$	0.273	0	0.714	0
$R_3$	0.273	0.714	0	0
$R_4$	0.454	0	0	0

After having  $M$ , we apply Eq. 2 and Eq. 3 to compute the Google matrix and the ranking of these re-

sources. For example, with  $d = 0.85$  and  $\epsilon = 0.01$ , the ranking vector of resources presented in Figure. 3 is converged to  $v^* = \{0.304, 0.272, 0.272, 0.152\}$  after 7 iterations. This result indicates that  $R_1$  is the most important resource w.r.t the given relations.

### 3 RESOURCE SEARCHING & RECOMMENDATION

In this section, we present an application of resource ranking in searching (subsection 3.1) and recommendation (subsection 3.2). We also present a basic scenario in which resource ranking is effectively used (subsection 3.3).

#### 3.1 Searching

E-learning portals always provide a search engine for resource searching. Most of search engines are implemented using text-matching techniques in order to match user's query and resource descriptions. Basically, when a user types keywords in the search box, the search engine will return a list of resources whose descriptions contain the provided keywords. The order of resources in the returned list can be arbitrary or based on certain criteria, such as most-clicked, recent-viewed or high-rated resources. In our approach, we propose to use resource ranking as another important criterion for the search result arrangement. We combine the resource ranking with a basic text matching technique. We sort the returned resources according to their ranking instead of the criteria above. Pseudo codes of our approach are presented in Algorithm 1.

---

#### Algorithm 1: Searching with resource ranking

---

**input** : keywords to search  
**output**: list of e-learning resources

- 1  $v^*$  : ranking vector of all resources ;
- 2  $L = \emptyset$  : list of resources to be returned ;
- 3 **foreach**  $R_i$  *in the corpus* **do**
- 4     **if**  $R_i$  *contains searching keywords* **then**
- 5          $L = L \cup R_i$
- 6     **end**
- 7 **end**
- 8 Sort  $R_i \in L$  by  $v^*[i]$  in descending order ;
- 9 **return**  $L$  ;

---

In line 1, resource ranking is computed and stored in vector  $v^*$ . All resources that contain the searching keywords are put into list  $L$  (lines 2-7). Finally, resources in  $L$  are sorted in descending order according

to their ranking values (line 8) and returned to the user (line 9).

As resource ranking allows search engine to return important resources (w.r.t to their coherent relations), it potentially improves the quality of search results. Apart from a basic combination presented in Algorithm 1, there can be other combinations with other sorting criteria, such as most-clicked and high-rated items. In these combinations, resource ranking can be applied in the last step to refine the search results.

#### 3.2 Recommendation

The goal of recommendation algorithms is to find items that are the most relevant to a particular user profile. Many algorithms have been developed on exploiting different aspects of user profile, such as preferences, rating, comments, behavior, social networks, trusted friends, and so on. They score the relevance between items and users based on a similarity metric. According to similarity scores, they generate a short list of recommended items for each user. In the TEL domain, recommendation algorithms target to provide for each user a list of suitable e-learning resources. They also use a similarity metric to evaluate the relevance between user profile and e-learning resources.

In our approach, we propose to combine recommendation techniques with resource ranking. We multiply the similarity between user profile and e-learning resources, which is evaluated by a recommendation technique, with the ranking of the corresponding resources to compute the final matching score. The list of recommended resources is created based on this score. Concrete computation is given in Eq. 9.

$$scr(U_i, R_j) = sim(U_i, R_j) \times v^*[j] \quad (9)$$

where  $sim(U_i, R_j)$  is the similarity between user  $U_i$  and resource  $R_j$  computed by a recommendation technique and  $v^*[j]$  is the ranking score of  $R_j$  in the corpus.

Pseudo codes of our approach are presented in Algorithm 2. Ranking of resources is stored in vector  $v^*$  (line 1). Lines 2-4 compute the final matching score between the active user profile and all resources. After all, resources are sorted by their final matching scores (line 5) and top-K resources are selected for recommendation (line 6).

#### 3.3 Scenario

Two basic interactions of a user when visiting an e-learning website are: *searching for resources* and *selecting a resource* to learn. These interactions are re-

---

**Algorithm 2:** Combination of resource ranking with a recommendation technique

---

**input** :  $U_i$ : active user,  
 $R$ : e-learning resources in the corpus  
**output**:  $rec(U_i)$ : recommended resources for  $U_i$

- 1  $v^*$  : ranking vector of all resources ;
- 2 **foreach**  $R_j$  *in the corpus* **do**
- 3      $sim(U_i, R_j)$  : similarity between  $U_i$  and  $R_j$   
       computed by a recommendation technique  
        $scr(U_i, R_j) = sim(U_i, R_j) \times v^*[j]$
- 4 **end**
- 5 Sort  $R_j \in R$  by  $scr(U_i, R_j)$  in descending order.  
    ;
- 6  $rec(U_i) \leftarrow$  top-K resources in the sorted list. ;

---

peated during a learning session. As resource ranking can be associated with a search engine (subsection 3.1) and recommendation techniques (subsection 3.2), it can be effectively used to refine the search results or the recommendation list. Figure 4 shows a scenario in which resource ranking can be integrated to support the two use-cases of the basic user interactions.

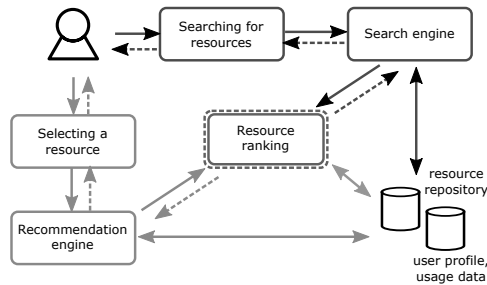


Figure 4: A scenario in which resource ranking is effectively used

In the first use case (Figure 4, top-right), when a user performs a search, the search engine retrieves from the repository resources that match to the provided keywords. Then, it sorts the retrieved resources according to their rankings learned from the ‘Resource ranking’ component. Finally, it returns the sorted list to the user. In the second use case (Figure 4, left-bottom), when the user selects a resource to learn, the recommendation engine studies the user profile to generate a recommendation list. Then, it refers to the ranking given by the ‘Resource ranking’ component to refine the recommendation list. Finally, it recommends the top-K resources in the refined list to the user.

## 4 EXPERIMENTS

We performed experiments on a dataset which is shared by the Open University of Humanities<sup>5</sup> (<http://www.uoh.fr/front>). We measure the performance of resource ranking based on the computation time with difference values of the convergence factor. Then, we evaluate the efficiency of our approach in two use-cases: searching and recommendation. We use Precision, Recall and Accuracy as metrics in our evaluation. Details of the obtained dataset (subsection 4.1), the metrics (subsection 4.2), our implementation (subsection 4.3) and experimental results (subsection 4.4) are as follows.

### 4.1 Dataset

We crawled all resource descriptions that are published on the website of the French Open University of Humanities. Each description is presented under the LOM format and provides basic information of the resource such as title, abstract, keywords, discipline, types, creator and relations to other resources. After parsing the crawled data, we obtained 1294 descriptions, which indicate 62 publishers (universities, engineering schools, etc.), 14 pedagogic types (slide, animation, lecture, tutorial, etc.), 12 different formats (text/html, video/mpeg, application/pdf, etc), 10 different levels (school, secondary education, training, bac+1, bac+2, etc.), 2 classification types (dewey, rameau) and 3 kinds of relation: ‘ispartof’, ‘haspart’ and ‘isassociatedto’. Among 1294 resources, 880 resources have relations with other resources, in which 692 resources have relation ‘ispartof’, 333 resources have relation ‘haspart’ and 573 resources have relation ‘isassociatedto’.

We have also obtained a shared package of anonymized usage logs from the university for our experiments. The shared logs consist of 415031 records with 70824 digitized IDs, in which 8844 IDs performed at least one ‘search’ action and 68658 IDs perform at least one ‘view’ action. Totally, there are 18677 search strings but only 7292 search strings that have results and followed by at least one ‘view’ action.

### 4.2 Metrics

Precision and Recall are the most frequently used measures in information retrieval for evaluating the efficiency of a system. Meanwhile, Accuracy is an alternative that judges the fraction of correct classification (Manning et al., 2008). These metrics are com-

<sup>5</sup>In French: Université ouverte des Humanités

puted based on the matching between retrieved data (such as recommended items) and relevant data, i.e. ground-truth data (such as actual used items). This matching can be summarized by the following contingency table (Table 3).

Table 3: Contingency table of retrieved and relevant items

	Relevant	Non-relevant
Retrieved	true positives (tp)	false positives (fp)
Not retrieved	false negatives (fn)	true negatives (tn)

Based on the number of items classified by Table 3, Precision, Recall and Accuracy are computed as follows.

$$\begin{aligned}
 Precision &= \frac{tp}{tp+fp}; & Recall &= \frac{tp}{tp+fn}; \\
 Accuracy &= \frac{tp+tn}{tp+fp+tn+fn}
 \end{aligned}
 \tag{10}$$

$Precision=1$  means that all retrieved items are relevant,  $Recall=1$  means that all relevant items are retrieved, and  $Accuracy=1$  means that retrieved items and relevant items are perfectly matched.

For example, consider a system that has 10 items {A, B, C, D, E, F, G, H, I}. An algorithm predicts that user X will use items {A, B, C} (retrieved) but actually X uses items {A, C, F, G, H} (relevant). So, in this prediction, tp = 2 (A, C), fp = 1 (B), fn = 3 (F, G, H) and tn = 4 (D, E, I, J). Therefore,  $Precision=\frac{2}{3}=0.67$ ,  $Recall=\frac{2}{5}=0.4$  and  $Accuracy=\frac{6}{10}=0.6$ .

### 4.3 Implementation

We developed a Java program to crawl and parse the obtained dataset. We used Apache Lucene<sup>6</sup> to remove stop words. We evaluate the efficiency of resource ranking in two cases: searching and recommendation.

In the case of searching, we developed a search function that matches a search string with resource descriptions in two cases: matching one of words (using OR operator) and matching all of words (using AND operator) that appear in the search string. We evaluated our approach by comparing the search results with and without resource ranking to the actually selected resources after each search action.

In the case of recommendation, we developed a content-based recommendation technique using the vector space model (VSM) and cosine similarity (Salton et al., 1975). We assumed that recently viewed resources reflect user interest. So, we

<sup>6</sup><http://lucene.apache.org>

built user profile with the keywords of her/his recent viewed resources. We presented each user profile as a vector of words. Resource descriptions were also presented as vectors. Then, these vectors were weighted using term frequency, inverse document frequency (TF-IDF) metrics. Finally, the similarity between a user profile and all resources were computed based on the cosine value of their vectors. For each user, we predicted her/his next viewed resources with and without resource ranking according to the computed similarity. Then, we compared our prediction to the list of resources that were actually selected by that user to compute Precision, Recall and Accuracy.

### 4.4 Results

In the first experiment, we target to measure *the performance of resource ranking*. We set the damping factor  $d = 0.85$  (like Google) and vary the convergence factor  $\epsilon =$  from  $10^{-4}$  to  $10^{-9}$ . Figure 5 shows the number of iterations needs to be performed to compute the ranking vector  $v^*$ , the corresponding computation time and the convergence of resource ranking. These results show that our approach can rapidly rank resources based on their relations, for instance, we can rank 1294 resources within 180ms with a very small threshold  $10^{-9}$ .

In the second experiment, we evaluate *the efficiency of our approach in the case of searching*. We use Precision, Recall and Accuracy as metrics in our evaluation (see section 4.2). For each search record of a user ID, we consider its followed viewed resources as relevant items. The resources returned by the search engine (with and without ranking) are consider as retrieved items. We compute the Precision, Recall and Accuracy by applying Eq. 10 with different top-N retrieved resources. We set the convergence factor  $\epsilon = 10^{-9}$  and measure the Precision, Recall and Accuracy with different top-N returned items.

Figure 6 gives an insight of the number of computed records in two cases of matching: one of words (using OR operator) and all of words (using AND operator) that appear in the search string. With different top-N values, the ‘all of words’ matching always returns a lower number of records than the ‘one of words’ matching.

Figure 7 and Figure 8 show the average results in the two searching cases with and without resource ranking. In both cases, searching with resource ranking achieves better results of Precision, Recall and Accuracy than searching without resource ranking for all top-N returned items. This means that resource ranking effectively improves the quality of search results.



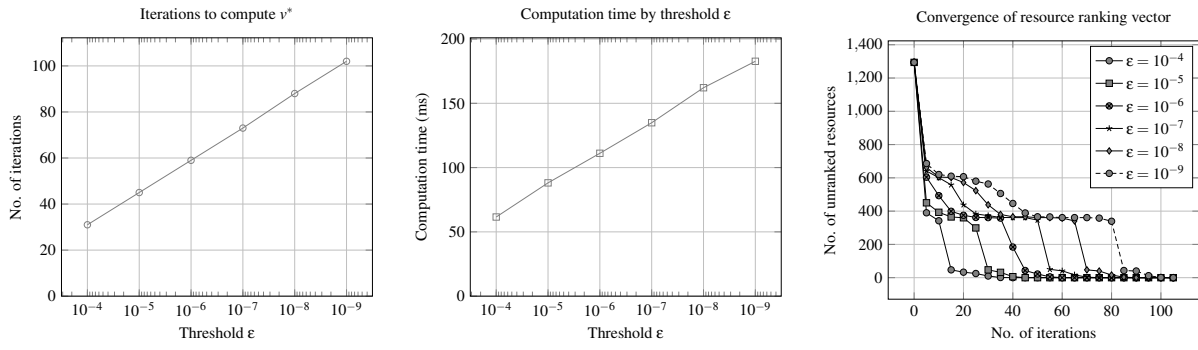


Figure 5: Experiments on resource ranking

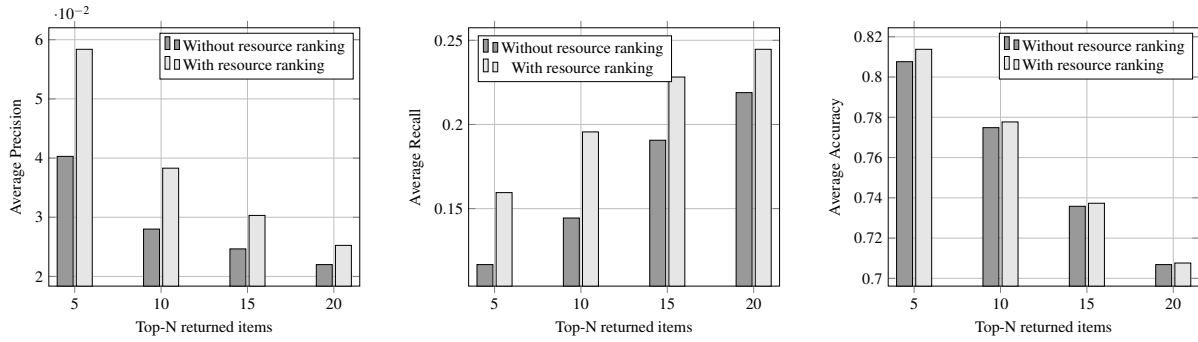


Figure 7: Average Precision, Recall and Accuracy in the case of searching with OR operator

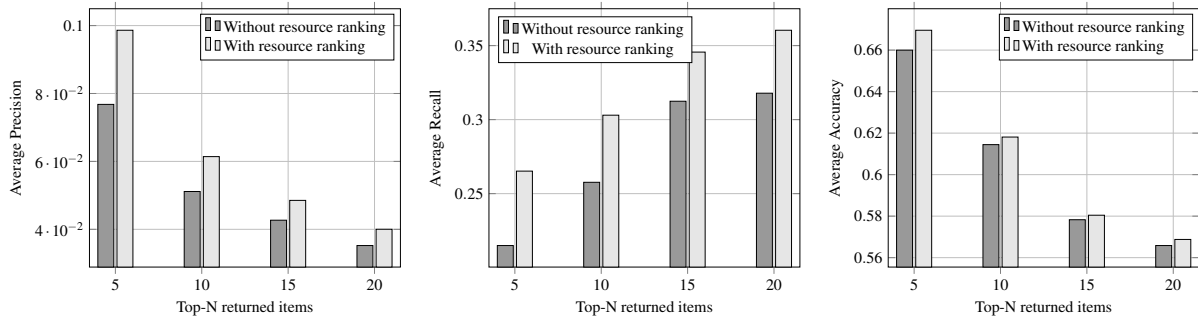


Figure 8: Average Precision, Recall and Accuracy in the case of searching with AND operator

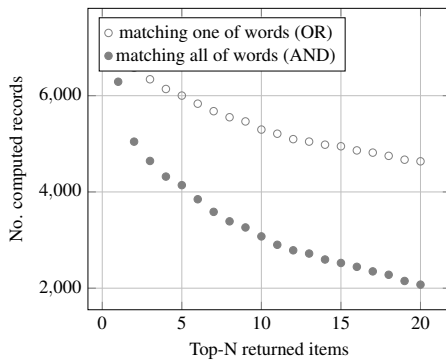


Figure 6: No. records computed based on search actions

In the third experiment, we evaluate *the efficiency of resource ranking in the combination with a content-*

*based recommendation technique.* We present user profile by keywords of the 5 recent viewed resources. We consider only user IDs that have viewed at least 10 and at most 100 different resources. For each user IDs, based on her/his first 5 viewed resources, we recommend 10 resources and compare them with her/his next viewed resources to compute the Precision, Recall and Accuracy.

Table 4 shows the average results of our experiment. The average values of Precision, Recall and Accuracy in the case of recommendation with resource ranking is a bit higher than their average values in the case of recommendation without resource ranking. This means that resource ranking improves slightly the quality of recommendation.

Table 4: Average Precision, Recall and Accuracy with and without resource ranking

	Without ranking	With ranking
Avg. Precision	0.016609784	0.016666667
Avg. Recall	0.01760926	0.017673905
Avg. Accuracy	0.982187556	0.982188438

## 5 RELATED WORK

In order to support e-learning resource discovery, the Advanced Distributed Learning has proposed a framework named Content Object Repository Discovery and Registration/Resolution Architecture (COR-DRA). This is an architecture that enables the interoperability among heterogeneous repositories, which allows facilitating the discovery and sharing of learning objects (LOs). However, relations among reusable LOs and the history of using these LOs are not maintained (Yen et al., 2010).

Data preprocessing has been also considered to improve the searching quality. The authors in (Hendez and Achour, 2014) have proposed an approach that extracts and indexes keywords of e-learning resources, while in (Saini et al., 2006), the authors proposed to automatically generate metadata for learning objects according to the taxonomic descriptions of learning domains. A metadata domain-knowledge search engine has also been proposed by (Zhuhadar et al., 2008). Each learning object (LO) in their approach is labeled with certain information such as the college's name, the course's name and the professor's name. Although the domain-knowledge is considered, the extraction of knowledge is done manually and repeated for all LOs. In addition, metadata about relations between LOs has not been considered.

In (Yen et al., 2010), the authors have proposed a guidance search engine for LO retrieval. They attempted to suggest learners to revise their search string for better approaching their favorite resources. For each provided search string, they suggest certain keywords that can be added to obtain better search results. The suggestion of each search string is specified according to the ranking of LOs, which is computed based on their download frequency, author reference and timescale. In our approach, instead of statistical metadata, we compute the ranking based on the relation between LOs. Although the input and the objective of the two approaches are different, their common concern, which is LO ranking, is comparable. However, the limitation and availability of a common dataset do not allow us to perform comparable experiments. Our dataset do not contain statistical metadata,

whereas their dataset is private and the system<sup>7</sup> is inaccessible.

On the research stream of recommendation for e-learning, authors in (Manouselis et al., 2011) and (Verbert et al., 2012) have made deep surveys on existing approaches that apply recommendation techniques to support online education. Common used techniques such as collaborative filtering (Lemire et al., 2005; Tang and McCalla, 2005), content-based filtering (Khribi et al., 2009; Koutrika et al., 2008), association rules (Lemire et al., 2005; Shen and Shen, 2004), user ratings (Drachsler et al., 2009; Manouselis et al., 2007), context aware (Verbert et al., 2012), feedback (Janssen et al., 2007) analysis and ontological structure (Tsai et al., 2006) have been exploited. However, none of existing approaches considers the resource ranking. In addition, most of them still remain at a design or prototyping stage (Manouselis et al., 2011). In our approach, we compute the resource ranking based on their relations. This ranking can be integrated into a search engine or a recommender system to improve the quality of results.

Our previous work (Chan et al., 2014) has mentioned the resource ranking. However, it placed the resource ranking in a combination with two other recommendation techniques and has not yet provided experiments on a real historical usage dataset. In this work, we focus specially on the resource ranking. We study deeply its impact on the quality of searching and recommendation and provide several experiments on a real dataset to show the efficiency of our approach.

## 6 CONCLUSION

In this paper, we present an approach that studies relations between e-learning resources to improve the quality of resource searching and recommendation. We propose to adapt the Google's PageRank algorithm on different kinds of relation between resources to compute their ranking. This ranking can be integrated into a search engine or combined with existing recommendation techniques to retrieve relevant resources. Experimental results showed that our approach improves the quality of searching and recommendation.

In the future work, we intend to perform more experiments with different combinations and metrics to deeply study the impact of resource ranking. We will also consider other criteria to improve the performance of our approach. For example, we will take

<sup>7</sup><http://mine.tku.edu.tw/>, last access: Dec. 18, 2014

into account the correspondence between the learner's level and the resource's prerequisites. We will also pay attention on the similar behaviors of users in the same communities (such as class, course, group of discussion or social network).

## ACKNOWLEDGEMENTS

This work has been fully supported by the French General Commission for Investment (Commissariat Général à l'Investissement), the Deposits and Consignments Fund (Caisse des Dépôts et Consignations) and the Ministry of Higher Education & Research (Ministère de l'Enseignement Supérieur et de la Recherche) within the context of the PERICLES project (<http://www.e-pericles.org>).

## REFERENCES

- Avancini, H. and Straccia, U. (2005). User recommendation for collaborative and personalised digital archives. *Int. J. Web Based Communities*, 1(2):163–175.
- Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International Conference on World Wide Web 7, WWW7*, pages 107–117, Amsterdam, The Netherlands, The Netherlands. Elsevier Science Publishers B. V.
- Chan, N., Roussanaly, A., and Boyer, A. (2014). Learning resource recommendation: An orchestration of content-based filtering, word semantic similarity and page ranking. In Rensing, C., de Freitas, S., Ley, T., and Muñoz-Merino, P., editors, *Open Learning and Teaching in Educational Communities*, volume 8719 of *Lecture Notes in Computer Science*, pages 302–316. Springer International Publishing.
- Drachsler, H., Pecceu, D., Arts, T., Hutten, E., Rutledge, L., Rosmalen, P., Hummel, H., and Koper, R. (2009). Remashed — recommendations for mash-up personal learning environments. In *Proceedings of the 4th European Conference on Technology Enhanced Learning: Learning in the Synergy of Multiple Disciplines, EC-TEL '09*, pages 788–793, Berlin, Heidelberg. Springer-Verlag.
- Engelhardt, M., Hildebrand, A., Lange, D., and Schmidt, T. C. (2006). Reasoning about elearning multimedia objects. In *First International Workshop on Semantic Web Annotations for Multimedia (SWAMM), joint with the 15th World Wide Web (WWW) Conference*, Edinburgh, Scotland.
- Hendez, M. and Achour, H. (2014). Keywords extraction for automatic indexing of e-learning resources. In *Computer Applications Research (WSCAR), 2014 World Symposium on*, pages 1–5.
- Huang, Y.-M., Huang, T.-C., Wang, K.-T., and Hwang, W.-Y. (2009). A markov-based recommendation model for exploring the transfer of learning on the web. *Educational Technology & Society*, 12(2):144–162.
- Hummel, H. G. K., van den Berg, B., Berlanga, A. J., Drachsler, H., Janssen, J., Nadolski, R., and Koper, R. (2007). Combining social-based and information-based approaches for personalised recommendation on sequencing learning activities. *IJLT*, 3(2):152–168.
- Janssen, J., Tattersall, C., Waterink, W., van den Berg, B., van Es, R., Bolman, C., and Koper, R. (2007). Self-organising navigational support in lifelong learning: How predecessors can lead the way. *Comput. Educ.*, 49(3):781–793.
- Khribi, M. K., Jemni, M., and Nasraoui, O. (2009). Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. *Educational Technology & Society*, 12(4):30–42.
- Koutrika, G., Ikeda, R., Bercovitz, B., and Garcia-Molina, H. (2008). Flexible recommendations over rich data. In *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys '08*, pages 203–210, New York, NY, USA. ACM.
- Lemire, D., Boley, H., McGrath, S., and Ball, M. (2005). Collaborative filtering and inference rules for context-aware learning object recommendation. *International Journal of Interactive Technology and Smart Education*, 2(3).
- Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., and Koper, R. (2011). Recommender systems in technology enhanced learning. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 387–415. Springer US.
- Manouselis, N., Vuorikari, R., and Assche, F. V. (2007). Simulated analysis of maut collaborative filtering for learning object recommendation. In

- In Workshop proceedings of the EC-TEL conference: SIRTEL07 (EC-TEL '07, pages 17–20.*
- Nadolski, R. J., van den Berg, B., Berlanga, A. J., Drachsler, H., Hummel, H. G., Koper, R., and Sloep, P. B. (2009). Simulating light-weight personalised recommender systems in learning networks: A case for pedagogy-oriented and rating-based hybrid recommendation strategies. *Journal of Artificial Societies and Social Simulation*, 12(1):4.
- Nilsson, M., Johnston, P., Naeve, A., and Powell, A. (2006). The future of learning object metadata interoperability. In Koohang, A., editor, *Principles and Practices of the Effective Use of Learning Objects*. Informing Science Press.
- Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.
- Saini, P., Ronchetti, M., and Sona, D. (2006). Automatic generation of metadata for learning objects. In *Advanced Learning Technologies, 2006. Sixth International Conference on*, pages 275–279.
- Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.
- Seeberg, C., Steinacker, A., and Steinmetz, R. (2000). Coherence in modularly composed adaptive learning documents. In Brusilovsky, P., Stock, O., and Strapparava, C., editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 1892 of *Lecture Notes in Computer Science*, pages 375–379. Springer Berlin Heidelberg.
- Shen, L.-p. and Shen, R.-m. (2004). Learning content recommendation service based-on simple sequencing specification. In Liu, W., Shi, Y., and Li, Q., editors, *Advances in Web-Based Learning – ICWL 2004*, volume 3143 of *Lecture Notes in Computer Science*, pages 363–370. Springer Berlin Heidelberg.
- Tang, T. and McCalla, G. (2005). Smart recommendation for an evolving e-learning system: Architecture and experiment. *International Journal on E-Learning*, 4(1):105–129.
- Tsai, K. H., Chiu, T. K., Lee, M. C., and Wang, T. I. (2006). A learning objects recommendation model based on the preference and ontological approaches. In *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies, ICALT '06*, pages 36–40, Washington, DC, USA. IEEE Computer Society.
- Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I., and Duval, E. (2012). Context-aware recommender systems for learning: A survey and future challenges. *IEEE Trans. Learn. Technol.*, 5(4):318–335.
- Wills, R. S. (2006). Google’s pagerank: The math behind the search engine. *Math. Intelligencer*, pages 6–10.
- Yen, N. Y., Shih, T. K., Chao, L. R., and Jin, Q. (2010). Ranking metrics and search guidance for learning object repository. *IEEE Trans. Learn. Technol.*, 3(3):250–264.
- Zhuhadar, L., Nasraoui, O., and Wyatt, R. (2008). Metadata domain-knowledge driven search engine in “hypermanymedia” e-learning resources. In *Proceedings of the 5th International Conference on Soft Computing As Transdisciplinary Science and Technology, CSTST '08*, pages 363–370, New York, NY, USA. ACM.