Learning Resource Recommendation: An Orchestration of Content-Based Filtering, Word Semantic Similarity and Page Ranking

Chan Nguyen Ngoc, Azim Roussanaly, Anne Boyer

Learning Resource Recommendation: An Orchestration of Content-Based Filtering, Word Semantic Similarity and Page Ranking

Nguyen Ngoc Chan, Azim Roussanaly, and Anne Boyer

Université de Lorraine
LORIA UMR 7503, France

Abstract. Technologies supporting online education have been abundantly developed recent years. Many repositories of digital learning resources have been set up and many recommendation approaches have been proposed to facilitate the consummation of learning resources. In this paper, we present an approach that combines three recommendation technologies: content-based filtering, word semantic similarity and page ranking to make resource recommendations. Content-based filtering is applied to filter syntactically learning resources that are similar to user profile. Word semantic similarity is applied to consolidate the content-based filtering with word semantic meanings. Page ranking is applied to identify the importance of each resource according to its relations to others. Finally, a hybrid approach that orchestrates these techniques has been proposed. We performed several experiments on a public learning resource dataset. Results on similarity values, coverage of recommendations and computation time show that our approach is feasible.

Keywords: Learning Resource · Technology Enhanced Learning · Content-based Filtering · Word Semantic Similarity · TF-IDF · VSM · PageRank

1 Introduction

During the past few years, along with the development of technologies supporting online education, numerous repositories of digital learning resources have been set up, such as MERLOT[1], OER Commons[2] and LRE For Schools[3] [21]. They provide open learning resources of various disciplines (such as arts, humanities, science and technologies, etc.), levels (such as primary school, secondary school, high school, higher education, etc.) and types (such as lab, lecture note, exercise, tutorial, etc.). These resources allow users to self-study or to consolidate their knowledge on different domains. However, the variety of these resources, in contrast, easily discourage users to continue studying. For instance, whenever users want to study a subject, they search or browse resources related to that subject

[1] http://www.merlot.org

[2] http://www.oercommons.org

[3] http://lreforschools.eun.org

and preview them using try-and-error method. They spend much time to reach to their expected resources. In addition, after learning a resource, they should redo the search/browse process if they want to find other related resources.

In order to encourage the usage of online learning resources, recommender systems are considered as a pivotal solution, especially on the Technology Enhanced Learning (TEL) domain [14]. Many approaches that apply recommendation techniques to support resource recommendation have been proposed. For example, they apply collaborative filtering [12, 20], content-based filtering [9, 11], examine user ratings [5, 15], study association rules [12, 19] or analyze user feedback [8]. Bayesian model [1], Markov chain [6], resource ontologies [16, 19] and hybrid models [16, 7] were also proposed. However, most of existing systems still remain at a design or prototyping stage. Only few of them have been reported to be evaluated through trials that involved human users [14].

In our work, we also target to encourage the usage of online learning resources with recommendations. However, different from existing approaches, we propose an innovative solution that orchestrate 3 recommendation techniques: *content-based filtering*, *word semantic similarity* and *page ranking*. Content-based filtering is applied to filter syntactically learning resources that are similar to user profile. Word semantic similarity is applied to consolidate the content-based filtering with word semantic meanings. Page ranking, which is inherited from the Google PageRank algorithm [2], is applied to identify the importance of each resource according to its relations to others. By hybridizing these techniques, our objective is three-fold: (i) to present an important application of recommendation techniques on a specific domain, which is online education, (ii) to show a possible combination of existing techniques to develop a hybrid recommendation approach, and (iii) to demonstrate a retrieval of important items that are not only syntactically but also semantically relevant to a request.

As keywords present concisely and precisely the content of resource, whereas recent viewed resources present recent user interest, we propose to build implicitly user profile based on keywords of recent viewed resources. By building user profile based on historical keywords, our approach is able to make recommendations that are close to user recent interest. In addition, it does not ask any effort from users such as completing registration form, specifying preferences, etc.

The paper is organized as follows: the next section elaborates in detail recommendation techniques applied in our approach and their combination. Experiments are presented in section 3. Related word is discussed in section 4 and we conclude our approach in section 5.

2 Learning Resource Recommendation

In this section, we present in detail approach to recommend learning resources for an active user. We firstly introduce a basic CB filtering approach that applies vector space model (section 2.1). Then, we show a refinement of text vectors based on word semantic similarity (section 2.2) and the ranking of resources based on their relations (section 2.3). Finally, we present a hybrid approach that combines these techniques (section 2.4).

2.1 Content-Based Filtering with Vector Space Model

Vector Space Model (VSM) is a method popularly used in Information Retrieval to compute the similarity between documents. It considers each word as a dimension. It presents each document as a vector in the n-dimensional space of words. Elements of each vector are weights of the corresponding words in the document, which present their importance in the corpus. These weights are computed by the term-frequency (TF) and inverse-document-frequency (IDF). Concretely, consider a documents d_i, which is presented by a vector $\overrightarrow{d_i} = \{w_{i1}, w_{i2}, \ldots, w_{in}\}$, where n is the number of words in the corpus and w_{ik} $(k = \overline{1..n})$ is the weight of the k^{th} element in the vector. w_{ik} is computed by Eq. 1

$$w_{i,k} = TF(i,k) \times IDF(k) = \frac{|w_k|}{|d_i|} \times \log \frac{n}{|D_k|} \tag{1}$$

where $|w_k|$ is the occurrence of the word w_k in d_i, $|d_i|$ is the number of words in d_i, and $|D_k|$ is the number of documents containing w_k.

Then, similarity between documents is computed by the cosine of the angle between their representative vectors. For example, similarity between two documents d_i and d_j is computed by Eq. 2.

$$sim(d_i, d_j) = cosine(\overrightarrow{d_i}, \overrightarrow{d_j}) = \frac{\overrightarrow{d_i} \cdot \overrightarrow{d_j}}{|\overrightarrow{d_i}| \times |\overrightarrow{d_j}|} \tag{2}$$

In our approach, we apply VSM to compute the similarity between a user profile and a resource description. User profile is defined by the set of keywords of his recent viewed resources, whereas resource description is all the text that are used to describe a resource[4].

Concretely, consider a corpus that consists of m learning resources and an active user u_a is viewing a resource r_a. Assume that m resources contain n different words. Let d_i be text description of resource r_i $(i = \overline{1..m})$. We present d_i as a vector $\overrightarrow{d_i} = \{w_{i1}, w_{i2}, \ldots, w_{in}\}$. w_{ik} $(k = \overline{1..n})$ is the TF-IDF weight (computed by Eq. 1) of the k^{th} corresponding word in the resource description.

Let $\{k_1, k_2, \ldots, k_t\}$ be t keywords of h recent viewed resources of u_a. We consider these historical keywords as a query q_a. We present q_a as a vector $\overrightarrow{q_a} = \{w_{a1}, w_{a2}, \ldots, w_{an}\}$ in the same space with resource descriptions. Elements of $\overrightarrow{q_a}$ are TF-IDF weights of corresponding words in the query.

Then, according to Eq. 2, similarity between q_a and d_i is given by Eq. 3.

$$sim(q_a, d_i) = \frac{\overrightarrow{q_a} \cdot \overrightarrow{d_i}}{|\overrightarrow{q_a}| \times |\overrightarrow{d_i}|} = \frac{\sum_{k=1}^{n} w_{ak} w_{ik}}{\sqrt{\sum_{k=1}^{n} w_{ak}^2} \times \sqrt{\sum_{k=1}^{n} w_{ik}^2}} \tag{3}$$

[4] In our experiments, d_i includes words in the title, abstract, keywords, discipline and classification of the resource

We apply Eq. 3 for all $d_i \in \{1, 2, \ldots, N\}$. Then, we sort resources in descending order according to the computed similarity with q_a. Finally, top-K resources are selected as the relevant resources of the active user's historical keywords.

As words can appear in different forms (singular, plural) and tenses (present, past, future), we proceed words stemming and stop-words removing before performing the similarity computation. In addition, as keywords can be singular words or compound words, we preprocess resource descriptions by identifying compound words that are matched to those in the historical keywords. By this identification, we can treat compound words as singular words.

For example, consider a query and a resource r_i with description d_i as follows:

q_a ={recommender system, technology enhanced learning, learning resources}

d_i ={On the technology enhanced learning domain, recommender systems are consider as a pivotal solution to recommend learning resources.}

After words stemming and stop-words removing, singular word treatment identifies the word occurrence in q_a and d_i as follows.

q_a: {recommend(1), system(1), technology(1), enhance(1), learn(2), resource(1)}

d_i: {technology(1), enhance(1), learn(2), domain(1), recommend(2), system(1), consider(1), pivot(1), solution(1), resource(1) }

meanwhile, compound word treatment identifies the word occurrence in q_a and d_i as follows.

q_a: {recommend system (1), technology enhance learn (1), learn resource (1)}

d_i: {technology enhance learn (1), domain(1), recommend system (1), consider(1), pivot(1), solution(1), recommend (1), learn resource (1)}

Based on these word occurrences, TF-IDF (Eq. 1) and VSM (Eq. 3) are applied to computed the similarity between q_a and d_i.

2.2 Query-Resource Matching based on Word Semantic Similarity

Polysemy and synonymy are common problems facing in text processing. If we deal with only word syntactical matching, without considering the semantic similarity, we easily miss potential matchings of different words which expose the same meaning. In this section, we present an integration of word semantic similarity in our approach in order to recommend more precisely learning resources.

There exist many research on the word semantic similarity that can be applied in our approach, such as [10, 4, 13, 18]. However, as we focus on the matching between resources instead of semantic similarity, discussion about this topic is out of scope of our paper. In our experiment, we adopt the work of Peter Kolb [10], which is a high accurate approach based on the co-occurrence of words in the Wikipedia dataset, to compute the word semantic similarity.

Consider an active user u_a who has recently viewed h resources which have a list of keywords $q_a = \{k_1, k_2, \ldots, k_t\}$. We consider this list as a query. For each resource r_i, which is considered to match with q_a, we propose to replace each word in the resource description, i.e d_i, by its most *semantically* similar word in the query if this word does not appear in the query. We update the weight of words in the resource description according to their semantical similarity with the selected words in the query . Finally, we weight words in both resources and query by TF-IDF and compute their similarity by applying VSM.

Concretely, consider a word v_x in resource description d_i with an occurrence o_x. Suppose that v_x is most semantically similar to a word k_y in q_a with a similarity value $s(v_x, k_y) \in (0, 1)$. We substitute v_x in d_i by k_y and update its weight to $w_{xy} = o_x s(v_x, k_y)$. This means that o_x times that v_x appears in d_i is considered as $o_x s(v_x, k_y)$ times that k_y appears in d_i. We repeat this substitution for all words in d_i.

Recall the example of q_a and d_i in section 2.1. The substitution of words in d_i by the most semantically similar words in q_a is given in Table 1. For instance, the word 'domain' in d_i is the most similar to the word 'resource' in q_a (similarity=0.015), we replace the word 'domain' by 'resource' and update its weight to $1 \times 0.015 = 0.015$ and so on.

	Similarity with words in q_a						Substitution	
v_x	recommend	system	technology	enhance	learn	resource	k_y	w_{xy}
domain	0.002	0.007	0.009	0.005	0.001	**0.015**	resource	0.015
consider	**0.056**	0	0.002	0.004	0.035	0.003	recommend	0.056
pivot	0.001	0.003	0.002	**0.0043**	0.001	0.0039	enhance	0.0043
solution	0.003	0.015	**0.017**	0.005	0.001	0.014	technology	0.017

Table 1: Example of word substitution based on semantic similarity

Assume that n_1 words in d_i are replaced by k_1 with the updated weights are $\{w_{11}, w_{21}, \ldots, w_{n_1 1}\}$, n_2 words in d_i are replaced by k_2 with updated weights are $\{w_{12}, w_{22}, \ldots, w_{n_2 2}\}$, and so on, and n_0 words in d_i are not replace by any word in the query.

The resource description d_i becomes $d'_i = \{k_1, k_2, \ldots, k_t, k_{t+1}, \ldots, k_{t+n_0}\}$ with corresponding weights are $\{\sum_{j=1}^{n_1} w_{j1}, \sum_{j=1}^{n_2} w_{j2}, \ldots, \sum_{j=1}^{n_k} w_{jk}, 0, \ldots, 0\}$.

Similarity between l_a and d_i is calculated by similarity between q_a and d'_i. As weights of $k_{t+1}, \ldots, k_{t+n_0}$ are 0, we can remove them in d'_i. d'_i becomes a vector in the same space with q_a. We, then, compute TF-IDF (Eq. 1) for words in both q_a and d_i. Finally, we apply VSM (Eq. 3) to calculate their similarity.

For example, after substitute words in d_i according to Table 1, we obtain d'_i={recommend(2.056), system(1), technology(1.017), enhance(1.0043), learn(2), resource(1.015)}, which is a vector in the same space with q_a. Then, we can apply TF-IDF and VSM to compute their similarity.

According to similarity between the query and all resources, we make recommendations by selecting the top-K most similar resources. In experiment, we run our approach and compare results in two cases: recommendations with and without word semantic similarity.

2.3 Resource Ranking Inspired from Google PageRank Algorithm

The importance of a resource in a corpus can be evaluated by different criteria such as the knowledge provided by that resource, its applications on different domains, its relations to other resources, or just the number of users viewing that resource. In this section, we present a ranking algorithm to evaluate the importance of resources based on their relations. This algorithm is inspired from the Google PageRank algorithm [2].

According to [2], rank of a page A is computed by Eq. 4.

$$PR(A) = (1-d) + d \times \sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)} \tag{4}$$

where $0 \leq d \leq 1$ is a damping factor (in [2], $d = 0.85$), $T_1, T_2, \ldots, T_n$ are pages which point to A and $C(T_i)$ is the number of links going out of T_i. Initial page rank of each page is $\frac{1}{N}$ where N is the number of pages. Page ranks of all pages are iteratively updated by Eq. 4 until they achieve stable values within a given threshold.

By another point of view[17, 22], page ranks are defined as a vector v^* that satisfies:

$$Gv^* = v^* \tag{5}$$

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

$$G = \frac{1-d}{N} S + dM \tag{6}$$

where S is the matrix with all entries equal to 1 and M is a transition matrix.

The transition matrix M 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}. $M_{[i,j]}$ satisfies $\sum_{i=1}^{N} M_{[i,j]} = 1, \forall j = \overline{1..N}$. According to [2], M is a Markov matrix and if a page j has k out-going links, each of them a weight $\frac{1}{k}$.

According to Eq. 5, 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 set to $\frac{1}{N}$. v^* is iteratively computed as following:

$$v_{i+1} = Gv_i \tag{7}$$

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

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

Inspired by the Google PageRank algorithm, we propose an algorithm to compute the ranking of learning resources. In our algorithm, we take into account resource relations instead of hyperlink between pages.

Basically, a resource can be a part of another resource, include other resources or associate to other resources. We define these relations respectively *'is part of'*, *'contains'* and *'associates to'*, in which 'is part of' is a 1-1 relation, 'contains' is a 1-n relation and 'associates to' is an n-n relation (Fig. 1).

Each relation not only presents a hyperlink between resources, but also exposes a particular semantic meaning. For instance, 'associates to' indicates a set of coherent resources that supplement each other to present some knowledge;

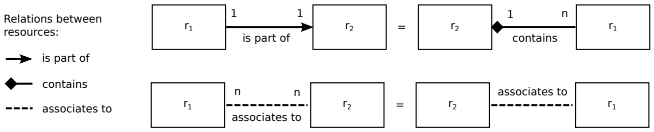


Fig. 1: Relations between resources

'contains' lists out a set of resources to be involved within a subject and some of them could be not coherent; 'is part of' signifies a resource which is a member of another resource but does not clearly present involved related resources. According to these meanings, we propose to assign a relation weight to each relation type. Concretely, 'associated to' has more weight than 'contains' and 'contains' has more weight than 'is part of'.

Let w_{ra}, w_{rc} and w_{rp} be weights of 'associates to', 'contains' and 'is part of', we have $w_{ra} > w_{rc} > w_{rp}$. For simplicity, we set[5]:

$$w_{ra} = \alpha w_{rc} = \alpha^2 w_{rp}, 0 < \alpha \leq 1 \tag{8}$$

In Google PageRank algorithm, weights of all hyperlink are set to be equal. In our approach, we weight relations between resources according to their types instead of giving an average weight for all relations. Concretely, assume that a resource r_i has a 'associates to' relations, b 'contains' relations and c 'is parts of' relations, we have:

$$a w_{ra} + b w_{rc} + c w_{rp} = 1 \tag{9}$$

From Eq. 8 and Eq. 9, we have:

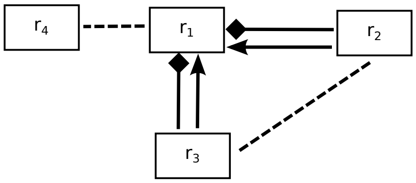
$$w_{rp} = \frac{1}{a\alpha^2 + b\alpha + c}; \quad w_{rc} = \frac{\alpha}{a\alpha^2 + b\alpha + c}; \quad w_{ra} = \frac{\alpha^2}{a\alpha^2 + b\alpha + c} \tag{10}$$

Eq. 9 ensures that the matrix M and G in Eq. 6 are Markov matrices. Hence, the multiplication of these matrices with an initial weighted-vector will converge to an eigenvector vector. It means that we can compute the PageRank vectors of resources based on the new weights. The weights calculated by Eq. 10 are used to initialize the matrix M and G in Eq. 6. Then the ranking vector v^* is calculated by Eq. 7.

For example, Fig 2 presents relations of 4 resources (r_1, r_2, r_3, r_4) in a corpus. Based on these relations, we can compute the weights of each relations (w_{ra}, w_{rc}, w_{rp}) and create the relation matrix M (the right column in Fig 2). Each element $M_{[i,j]}$ presents the weight of relation from r_j to r_i and the matrix satisfy that sum of all elements in each column is equal to 1. Based on M, we can easily compute G and v^* by applying Eq. 6 and Eq. 7.

The PageRank vector presents the importance of resources according to their relations. Their rankings can be used in a resource searching application, similar to page rankings are used in the Google search engine. They can also be considered as a parameter in a recommendation application in order to refine the recommendation result.

[5] In our experiment, we set $\alpha = 0.9$.



r_1: $a = 1, b = 2, c = 0,$
r_2: $a = 1, b = 0, c = 1,$
r_3: $a = 1, b = 0, c = 1,$
r_4: $a = 1, b = 0, c = 0,$

With $\alpha = 0.9$, we have:
r_1: $w_{ra} = 0.31, w_{rc} = 0.345$;
r_2: $w_{ra} = 0.45, w_{rp} = 0.55$;
r_3: $w_{ra} = 0.45, w_{rp} = 0.55$;
r_4: $w_{ra} = 1$;

and the matrix M:

$$\begin{array}{ccccc} & r_1 & r_2 & r_3 & r_4 \\ r_1 & 0 & 0.55 & 0.55 & 1 \\ r_2 & 0.345 & 0 & 0.45 & 0 \\ r_3 & 0.345 & 0.45 & 0 & 0 \\ r_4 & 0.31 & 0 & 0 & 0 \end{array}$$

Fig. 2: Example of resource relations and the corresponding matrix

2.4 A Hybrid Recommendation Approach

The content-based filtering algorithm (section 2.1) and its refinement with word semantic similarity (section 2.2) enable the retrieval of resources that are syntactically and semantically similar to user profile. Meanwhiles, resource ranking (section 2.3) helps to identify the importance of resources based on their relations. Therefore, their combination possibly retrieves resources that are both important and relevant to a user.

In our approach, we propose to multiply the similarity between user profile and resources with the ranking of resources to infer their final matching scores. Concretely, consider an active user u_a who has a profile q_a and a resource r_i which has a description d_i. Let d'_i be the refined resource of d_i by applying word semantic similarity according to the query q_a. The final matching score between q_a and r_i is given by Eq. 11.

$$scr(q_a, r_i) = sim(q_a, d'_i) \times v^*(i) \tag{11}$$

where $sim(q_a, d'_i)$ is the similarity between q_a and d'_i given by Eq. 3, $v^*(i)$ is the ranking of r_i in the corpus.

We compute the matching scores (Eq. 11) between q_a and all resources. We sort these scores in descending order and select top-K corresponding resources for recommendations.

Pseudo codes of the hybrid algorithm is described in Algorithm 1. In line 1, resource ranking is computed and stored in vector v^*. From line 2 to line 9, similarity between user profile q_a and each resource r_i (line 3-7) and their final matching score (line 8) are computed. After all, resources are sorted by their final matching scores (line 10) and top-K resources are picked up for recommendations (line 11).

3 Experiments

We performed experiments on the learning resources that are published by the Open University of Humanities[6] (`http://www.uoh.fr/front`). However, due to

[6] In French: Université ouverte des Humanités

Algorithm 1: Hybrid of content-based filtering and resource ranking

```
input  : $q_a$: user profile of $u_a$, $R$: set of learning resources
output: $rec(a)$: recommendations for $u_a$
$v* \leftarrow$ PageRank ($R$) ;
foreach $r_i \in R$ do
    $d_i \leftarrow$ Text description ($r_i$) ;
    $d'_i \leftarrow$ Refinement of $d_i$ by $q_a$ by word semantic similarity ;
    $\overrightarrow{q_a} \leftarrow$ TF-IDF vector of $q_a$ ;
    $\overrightarrow{d'_i} \leftarrow$ TF-IDF vector of $d'_i$ ;
    $sim(q_a, d'_i) \leftarrow$ cosine ($\overrightarrow{q_a}$, $\overrightarrow{d'_i}$) ;
    $scr(q_a, r_i) \leftarrow sim(q_a, d'_i) \times v^*(i)$ ;
end
Sort $r_i \in R$ by $scr(q_a, r_i)$ in descending order. ;
$rec(a) \leftarrow$ top-K resources in the sorted list. ;
```

the university privacy, historical usage data is not shared. So, we could not evaluate our approach based on ground-trust based metrics such as Precision/Recall, MAE, RMSE, etc. Instead, we measured the *similarity values*, the *coverage* of recommendations, the *convergence* of ranking vector and the *computation time* of our proposed algorithms in order to evaluate the *feasibility* of our approach. We elaborate in the following the collected dataset (section 3.1), our implementation (section 3.2) and experimental results (section 3.3).

3.1 Dataset

The Open University of Humanities is a French numerical university that provides open access to learning resources related to human science. These resources are created by teachers, lecturers of French higher educational schools. Each resource is published together with its description under the Learning Object Metadata (LOM) format. This description provides basic information of the resource such as title, abstract, keywords, discipline, types, creator, relations to other resources, etc.

As resource descriptions are public under a standard format, we crawled and parsed them to extract necessary information for our experiments. We collected 1294 resource 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.) and 2 classification types (dewey, rameau). Among 1294 resources, 880 resources have relations with other resources, in which 692 resources have relation 'is part of', 333 resources have relation 'contains' and 573 resources have relation 'associates to'.

The collected dataset contains essential information for our proposed algorithms, including resource descriptions, which are used in the content-based filtering and word semantic similarity algorithms, and their relations, which are used in the resource ranking algorithm.

3.2 Implementation

We developed a Java program to crawl and extract the public resources. We used Apache Lucene[7] for word stemming and stop words removal. We used DISCO library[8] for word semantic similarity. We simulated 1000 queries, each of which includes keywords of 10 resources. We assumed that these resources are recently viewed by an active user. For each query, we computed recommendations in 6 different cases:

1. *SingularKW*: we consider keywords and resource descriptions as sets of singular keywords and run the content-based filtering algorithm.
2. *CompoundKW*: we preprocessed resource descriptions by identify their compound words that are matched with compound words in the query. Then, we consider each compound word as a singular word and run the content-based filtering algorithm.
3. *SemanticCB*: we replace words in resource descriptions by the most semantically similar words in the query. Then, we run the content-based filtering algorithm with the new resource descriptions.
4. *SingularKW-PageRank*: we run the hybrid algorithm that combines the content-based filtering with singular word matching and resource rankings based on their relations.
5. *CompoundKW-PageRank*: we run the hybrid algorithm that combines the content-based filtering with compound word matching and resource rankings.
6. *SemanticCB-PageRank*: we run the hybrid algorithm that combines the content-based filtering with semantic word similarity and resource ranking.

We run our proposed algorithms on a Mac laptop with the configuration as follows: CPU 2GHz Core i7, Memory 8G 1600 MHz, SSD 251G and OS X 10.9.2.

3.3 Results

We set damping factor $d = 0.85$ (like Google) and differential parameter $\alpha = 0.9$ (see section 2.3). We run our algorithms on 1000 different queries and obtained results as follows.

In the first experiment, we target to measure *similarity values* between queries and resources. For each query, we compute the average similarity values of top-K resources that are selected for recommendations. Fig. 3 shows experimental results of the 6 different cases mentioned above. The SemanticCB and SemanticCB-PageRank cases achieve the highest average similarity values as they take into account both syntactic and semantic word similarity. Meanwhile, The CompoundKW and CompoundKW-PageRank have the lowest similarity values as they consider only syntactical matching between compound words which leads to the smallest number of matching pairs. The cases with PageRank algorithm have very small similarity values because the ranking values of resources are very small to satisfy that sum of all of them is equal to 1 (Max.=7.16×10^{-3}, Min.=1.16×10^{-4}).

[7] `http://lucene.apache.org`

[8] `http://www.linguatools.de/disco/disco_en.html`

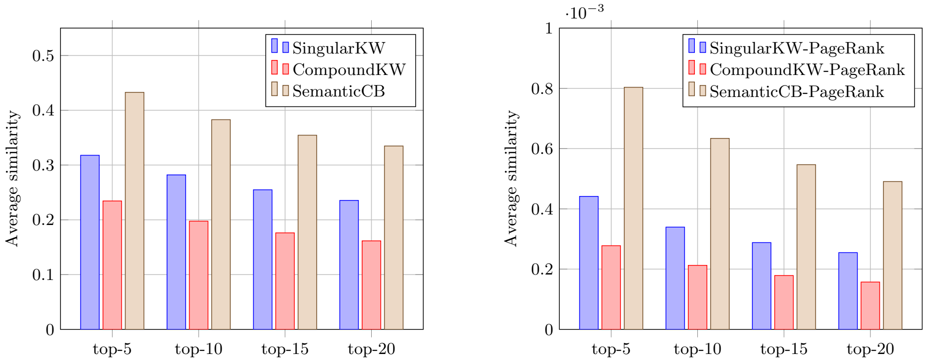


Fig. 3: Average similarity values with top-K selections

In the second experiment, we measure the *coverage* of recommendations, i.e. the percentage of resources that are considered for the top-K selection. We obtained that the CompoundKW and CompoundKW-PageRank cases (notated by CompoundKW*) have the lowest coverage (Fig. 4). It is because the number of compound word matchings is much smaller than the number of singular word matchings and word semantic matchings. We also obtained that the coverage of the SemanticCB* cases is always 1. It means that for each query, we always find at least a word in a resource description that is semantically matched to a word in the query with a matching value greater than 0.

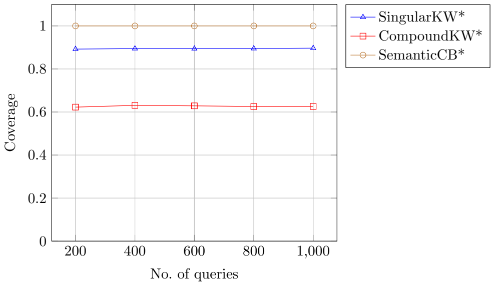


Fig. 4: Coverage of recommendations

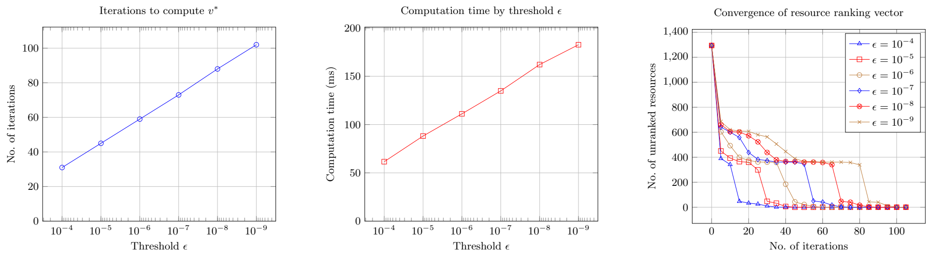


Fig. 5: Experiments on resource ranking

In the third experiment, we target to measure the *convergence* of resource rankings with different thresholds (from 10^{-4} to 10^{-9}). Fig. 5 shows the number of iterations needs to be performed to compute the ranking vector v^*, the corresponding computation times and the convergence of resource rankings. Theses 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}.

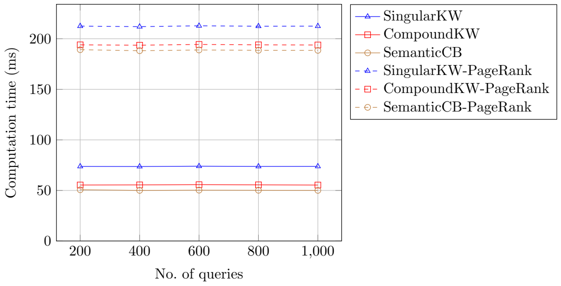


Fig. 6: Computation time with $\epsilon = 10^{-6}$

In the last experiment, we target to measure the *computation time* of our algorithms, without the data preprocessing time. Fig. 6 shows that the SemanticCB* cases have the smallest computation time while the SingularKW* cases have the highest computation time. It is because the number of dimensions in the resource vector space in the SemanticCB* cases is the smallest and in the SingularKW* cases is the highest. The cases with PageRank have much more higher computation time than others as they include the computation time of resource ranking. Fig. 6 also shows that our algorithms can make recommendations in very short time (around 200 ms with 1294 resources).

According to the limitation of dataset, we do not evaluate our approach using ground-trust based metrics. However, experimental results on the *similarity values*, the *coverage* of recommendations, the *convergence* of ranking vector and the *computation time* showed that our approach is able to make recommendations in a *very short time* and *for all queries*. This means that our approach is *feasible* in reality.

4 Related Work

On the TEL domain, a number of recommendation approaches have been proposed to encourage the usage of learning resources for online education. Manouselis et. al. [14] have made a deep survey on these existing approaches, which apply recommendation techniques on different online education contexts. Common used techniques such as collaborative filtering [12, 20], content-based filtering [9, 11], association rules [12, 19], user ratings [5, 15] and feedback [8] analysis have been exploited. However, none of existing approaches considers a combination of syntactic and semantic matching. In addition, most of them still remain at a design or prototyping stage. Only few of them have been reported to be evaluated through trials that involved human users [14]. In our approach, we take into account both syntactic and semantic matching together with resource ranking. We also provide experiments on a dataset of real online learning resources.

A related work that applied content-based and collaborative filtering on recent viewed resources has been proposed by Khribi et. al. [9]. They also presented experiments on resources that are presented in the standard Learning Object Metadata (LOM) format. However, different from them, we present another combination of existing recommendation techniques and we consider historical keywords as user profile instead of entire resource content. Although we performed experiments on LOM formatted resources, our approach can be applied on different resource formats as we take in to account resource descriptions instead of their formats.

Another related work that considered word similarity has been proposed by Chen et. al. [3]. They apply word similarity to compute the matching between user query and web contents. However, in their approach, they proposed to expand the user query by including all of their similar words. In our approach, instead of expanding the user query, we replacing words in a resource description by their most semantically similar words in the query.

5 Conclusion

Recommender systems have been considered as a pivotal solution to encourage the usage of online learning resources. In this paper, we present an innovative hybrid approach that combines three recommendation techniques: collaborative-filtering, semantic similarity and page rankings to generate resource recommendations. By this combination, our approach is able to recommend important resources that are syntactically and semantically relevant to user requests. In our approach, user profile is implicitly built based on keywords of recent viewed resources. Hence, we do not ask any effort from users. In addition, as recent viewed resources present recent interest of user, our approach is able to recommend resources that are close to user interest.

In future work, we will validate our approach on other datasets using ground-truth based metrics such as precision/recall, MAE and RMSE. We will take into account resource levels in order to filter resources that are best fitted to the learner's level. We also plan to integrate collaborative filtering and clustering techniques to improve the quality of recommendations.

6 Acknowledgments

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

1. Avancini, H., Straccia, U.: User recommendation for collaborative and personalised digital archives. Int. J. Web Based Communities 1(2), 163–175 (Jan 2005)
2. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: WWW7. pp. 107–117. Elsevier Science Publishers B. V. (1998)

3. Chen, L., Sycara, K.: Webmate: A personal agent for browsing and searching. In: AGENTS '98. pp. 132–139. ACM, New York, NY, USA (1998)
4. Danushka, B., Yutaka, M., Mitsuru, I.: Measuring semantic similarity between words using web search engines. In: WWW '07. pp. 757–766. ACM (2007)
5. Drachsler, H., Pecceu, D., Arts, T., Hutten, E., Rutledge, L., Rosmalen, P., Hummel, H., Koper, R.: Remashed — recommendations for mash-up personal learning environments. In: EC-TEL '09. pp. 788–793. Springer-Verlag (2009)
6. Huang, Y.M., Huang, T.C., Wang, K.T., Hwang, W.Y.: A markov-based recommendation model for exploring the transfer of learning on the web. Educational Technology & Society 12(2), 144–162 (2009)
7. Hummel, H.G.K., van den Berg, B., Berlanga, A.J., Drachsler, H., Janssen, J., Nadolski, R., Koper, R.: Combining social-based and information-based approaches for personalised recommendation on sequencing learning activities. IJLT (2007)
8. Janssen, J., Tattersall, C., Waterink, W., van den Berg, B., van Es, R., Bolman, C., Koper, R.: Self-organising navigational support in lifelong learning: How predecessors can lead the way. Comput. Educ. 49(3), 781–793 (Nov 2007)
9. Khribi, M.K., Jemni, M., Nasraoui, O.: Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. Educational Technology & Society 12(4), 30–42 (2009)
10. Kolb, P.: Experiments on the difference between semantic similarity and relatedness. In: Jokinen, K., Bick, E. (eds.) NODALIDA'09. vol. 4, pp. 81–88 (2009)
11. Koutrika, G., Ikeda, R., Bercovitz, B., Garcia-Molina, H.: Flexible recommendations over rich data. In: RecSys '08. pp. 203–210. ACM (2008)
12. Lemire, D., Boley, H., McGrath, S., Ball, M.: Collaborative filtering and inference rules for context-aware learning object recommendation. International Journal of Interactive Technology and Smart Education 2(3) (August 2005)
13. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans. on Knowl. and Data Eng. 15(4), 871–882 (Jul 2003)
14. Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., Koper, R.: Recommender systems in technology enhanced learning. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 387–415 (2011)
15. Manouselis, N., Vuorikari, R., Assche, F.V.: Simulated analysis of maut collaborative filtering for learning object recommendation. In: SIRTEL'07 (2007)
16. Nadolski, R.J., van den Berg, B., Berlanga, A.J., Drachsler, H., Hummel, H.G., Koper, R., Sloep, P.B.: Simulating light-weight personalised recommender systems in learning networks: A case for pedagogy-oriented and rating-based hybrid recommendation strategies. J. of Artificial Societies and Social Simulation (2009)
17. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (1999)
18. Richardson, R., Smeaton, A., Murphy, J.: Using wordnet as a knowledge base for measuring semantic similarity between words. In: AICS'09 (1994)
19. Shen, L.p., Shen, R.m.: Learning content recommendation service based-on simple sequencing specification. In: ICWL'4, pp. 363–370 (2004)
20. Tang, T., McCalla, G.: Smart recommendation for an evolving e-learning system: Architecture and experiment. IJ. on E-Learning pp. 105–129 (2005)
21. Tzikopoulos, A., Manouselis, N., Vuorikari, R.: An overview of learning object repositories. In: Erickson, J. (ed.) Database Technologies: Concepts, Methodologies, Tools, and Applications, pp. 362–383. IGI Global (2009)
22. Wills, R.S.: Google's pagerank: The math behind the search engine. Math. Intelligencer pp. 6–10 (2006)