A System to Recommend Open Educational Resources during an Online Course

Hiba Hajri, Yolaine Bourda and Fabrice Popineau
LRI, CentraleSupelec, bat 650 (PCRI), F-91405, France

Keywords: Technology Enhanced Learning, Online Learning Environment, Personalization, Learner Profile, OER, Recommender System, MOOC.

Abstract: Recently, personalization in Technology Enhanced Learning (TEL) has been researched extensively. With the spreading of online learning environments (OLE) as MOOCs and LMSs, a large number of learners with different characteristics and backgrounds can follow online courses. To support personalization, recommender systems can be used to provide each learner with learning objects helping him to reach his learning objectives. These recommendations are more specific to compute than usual recommendations (like consumer products). Furthermore, if they are included in a course, they depend not only on the learner's profile but also on the content of the course, because they need to fit with the course format at any point. At the same time, there is a growing number of open educational resources (OER) available to usefully enrich the content of online courses. To facilitate their reuse some of these OERs are described with metadata schemas following Linked Open Data principles (LOD). In this paper, we introduce a MOOC-based OER recommender system (MORS) that can be plugged in an OLE to provide recommendations of OERs to learners based on their profiles, the course profile and a process for calculating recommendations based on OERs metadata. This paper presents our approach that has been implemented in a MOOC platform: Open edX. However the proposed approach could be implemented in any OLE by using the same process to calculate recommendations, as long as the learner and the course profiles can be extracted.

1 INTRODUCTION

Personalization in the field of technology enhanced learning (TEL) is a topic that received a lot of concern by researchers. But, with the spreading of online learning environments (OLE) as MOOCs and LMSs, the issue of personalization becomes more acute. In fact the same course can be followed by a large number of learners, with different educational levels, learning styles, preferences, etc. This makes the provision for an efficient one-size-fits-all learning content more difficult.

To support personalization in OLE, recommender systems can be used in order to offer to each learner learning objects that match with his needs and his learning objectives. But these recommendations are more specific to compute than usual recommendations (like consumer products) since they depend both on the learner profile and the course profile. More precisely, learning objects to be recommended to each learner have to be suitable to his profile while remaining coherent with the course.

At the same time, the amount of Open Educational Resources (OER) available on the web is permanently growing. These OERs are considered as an efficient way for providing education for all and can usefully enrich the content of online courses. But the quality and the availability of OERs descriptions (metadata) are crucial to their reuse. In this context, linked Open Data principles are more and more applied to describe OERs in order to facilitate their discovery and their reuse.

In this paper, we introduce a MOOC-based OER recommender system (MORS) that can be plugged in an OLE to support learners. The proposed system provides recommendations of OERs to learners based on their profiles, the course profile and a process for calculating recommendations based on OERs metadata. This paper presents our approach that has been implemented for Massive Open Online Courses (MOOC) platforms. We choose MOOCs because (1) their large number of learners with varied profiles make them good candidates for personalization, and (2) because targetting an open platform like Open edX will allow us to widely disseminate our system and to make it reusable. However the proposed approach

Hajri, H., Bourda, Y. and Popineau, F.
A System to Recommend Open Educational Resources during an Online Course.
DOI: 10.5220/0006697000990109
In *Proceedings of the 10th International Conference on Computer Supported Education (CSEDU 2018)*, pages 99-109
ISBN: 978-989-758-291-2

could be implemented in any OLE by using the same process to calculate recommendations, as long as the learner and the course profiles can be extracted.

This paper begins with related work. In section 3 we introduce the recommendation scenarios proposed by our solution. Section 4 draws the architecture of the proposed system. Section 5 presents how we implemented our solution. The evaluation plan is proposed in section 6. Section 7 concludes the paper and presents future directions.

2 RELATED WORK

Even though Personalization in TEL is a research topic with a long history, studies on MOOCs personalization have started since 2013 (Sunar et al., 2015). In this context, different personalization approaches have been adopted. One of the most popular techniques relies on a recommender system.

Some approaches are dedicated to MOOCs in a specific subject. For example, the approach proposed by (Maran et al., 2015) is specific to health MOOCs in the area of Motivational Interviewing. It recommends to the learner the MOOC resources related to concepts they only need to know, by analyzing learners contexts. However, this approach has been dedicated to MOOCs only about a specific topic and can't be considered as a generic solution.

Some approaches are not dedicated to a specific subject but they generate their recommendations based on internal resources. For example, (Agrawal et al., 2015) targets learners who post a question in a MOOC discussion which reflects a confusion and recommends educational videos related to the confusion subject. (Bansal, 2013) recommends additional learning activities to learners who show a lack of knowledge in a particular subject.

But when internal resources fail to meet the expectations of the learner, it becomes interesting to also recommend external resources. This is the case in (Alario-Hoyos et al., 2014). In fact, it recommends to the learner a set of MOOCs which mostly match his learning objectives. Another approach (Paquette et al., 2015) offers a scenario of activities to each group of learners according to the gap between their actual competencies and the target ones. This scenario can perform a number of recommendations of either internal or external educational resources, captured from the web. However, on the one hand, even if external resources are recommended, these approaches don't consider MOOC specificities and the recommended resources may be out of line with the MOOC. Since these external resources will complement the MOOC initial learning path, it is important to select resources that are in sync with this path. These selected resources have to fit the MOOC by respecting its specific characteristics. On the other hand, the recommender system has to adapt his results dynamically according to the course's point, the evolution of the knowledge acquired by the learner during the MOOC and the versatility of OER repositories. Dynamic computation is needed because the set of available OERs at the MOOC time is constantly changing.

In our work, we propose a generic solution providing recommendations of OERs in a MOOC platform when a lack of knowledge is detected for a learner. These recommendations are computed dynamically based on different learner characteristics and also on MOOC specificities.

3 RECOMMENDATION SCENARIOS

In this section, we describe how our system personalizes a MOOC for a learner. More precisely, we introduce some realistic scenarios of recommendation offered by MORS: where and when exactly the recommendation process is triggered for a learner during the MOOC.

Let's consider the MOOC as a set of sections (Figure 1). Each section offers pedagogical resources as video, text, quiz, etc. We consider also that studying the MOOC requires some prerequisites with a certain performance level defined by the MOOC's creator who is the teacher. Each MOOC's section provides some learning objectives with a certain performance level defined by the teacher.

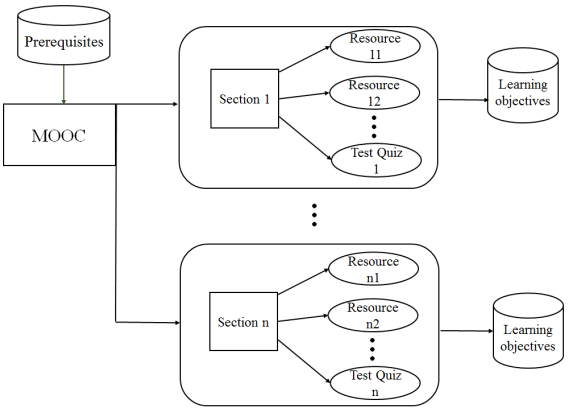

Figure 1: The MOOC organization.

In our solution, we decide to recommend OERs to the learner at two different kind of stages of the

MOOC: before starting the MOOC and after each MOOC's section.

Before Starting the MOOC. Once a learner is enrolled in a MOOC and decides to start its first section, MORS verifies if the learner has the prerequisites of the MOOC with the appropriate performance degrees. If a lack of knowledge is detected in at least one of the MOOC's prerequisites, the recommendation process is triggered and a set of OERs dealing with the appropriate prerequisites are recommended to the learner.

At the end of Each MOOC's Section. As stated earlier, each MOOC's section has at least one learning objective. This learning objective can also be a prerequisite for the following section. So it is important to ensure that the learner has assimilated the section's content. That is why at the end of each section, a quiz is presented to the learner where each question aims to assess his assimilation level of at least one of the section's learning objective. Now if the learner gets bad results in the quiz, MORS triggers the recommendation process in order to recommend to him a set of OERs dealing with the learning objectives where he failed.

4 SYSTEM ARCHITECTURE

In this section we describe the architecture of our system for recommending OERs in a MOOC (MORS). MORS is composed of four major modules (Figure 2): MOOC modelling module, learner modelling module, PreSearch module and results refinement module.

At first, the MOOC modelling module generates the MOOC profile. When a learner logins to the MOOC for the first time, the learner modelling module generates the learner's profile which will be updated during the MOOC. When the system identifies gaps in student mastery of a certain topic, the PreSearch module requests the external OERs descriptions repositories in order to collect an initial set of OERs descriptions dealing with the appropriate topic. Once the system has this initial set, the refinement module applies selection and ranking depending on different criteria. These criteria have been selected to ensure adaptation to the learner profile and coherence with the MOOC.

4.1 MOOC Modeling Module

This module aims at generating the MOOC profile. The MOOC profile contains information collected from the teacher when he creates the MOOC. These information relate to the domain and the knowledge elements of the MOOC. We consider two types of knowledge elements: learning objectives of each MOOC week and prerequisites of the MOOC.

Notations. In this paper, we denote the number of MOOC weeks by n_{week}, the set of MOOC knowledge elements as KE, the set of MOOC prerequisites as P, the set of the learning objectives provided by the kth week as LO_k, where $1 \leq k \leq n_{week}$ and the set of the learning objectives provided by the entire MOOC as LO. Then $LO = \cup_{k=1}^{n_{week}} LO_k$ and $KE = LO \cup P$.

The teacher defines the MOOC knowledge elements together with their performance degrees. In this work, we use the performance degrees as introduced in (Imran et al., 2016) (1: beginner, 2: intermediate and 3: expert) to which we add (0: no performance).

Definition 1. (Performance Degree by Teacher) *Given a knowledge element ke from KE, the performance degree of ke, set by the teacher, is defined by the function LP_T.*

$$LP_T : \begin{cases} KE \longrightarrow \{1,2,3\} \\ ke \longmapsto lp \in \{1,2,3\} \end{cases}$$

The prerequisites and the learning objectives of the MOOC associated to their performance degrees are modelled respectively by the vectors $V_{P,MOOC}$ and $V_{LO,MOOC}$. We represent in (Figure 3) the evolution of knowledge elements during the MOOC. As input, we represent in $V_{P,MOOC}$ the required performance degrees defined by the teacher for each prerequisite. Then we represent the evolution of $V_{LO,MOOC}$ during the MOOC. At first, $V_{LO,MOOC}$ is initialized to zero. Thus, at the end of each week, $V_{LO,MOOC}$ is updated with new values. These values correspond to performance degrees expected to be acquired in learning objectives provided by the week.

The MOOC domain is deduced from the MOOC description and represented by a variable called D_{MOOC}.

4.2 Learner Modelling Module

This module is responsible for generating and updating the learner profile during the MOOC. The learner profile contains his knowledge elements, his learning style and other registration information. Concerning

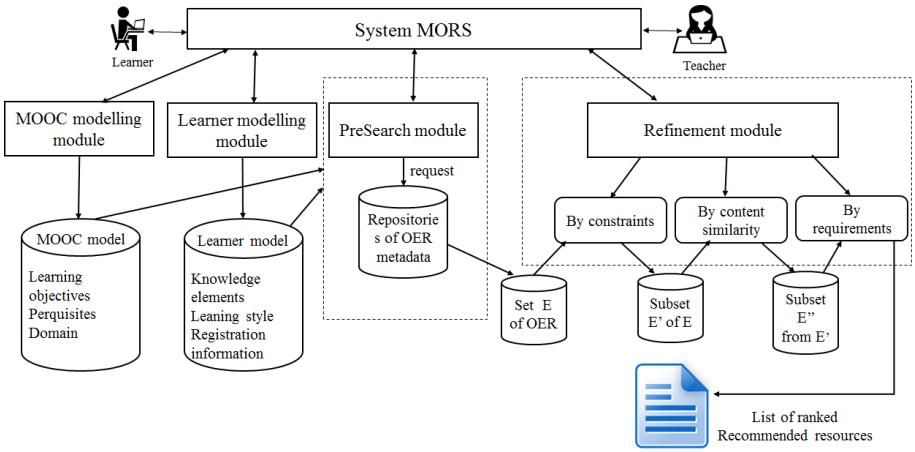

Figure 2: The architecture of MORS.

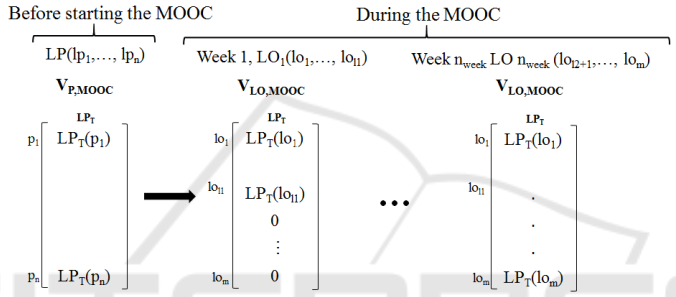

Figure 3: The evolution of the knowledge elements during the course.

knowledge elements, we consider only the prerequisites and learning objectives of the MOOC. So the learner has a performance degree for each MOOC prerequisite and each MOOC learning objective.

Definition 2. (Performance Degree of Learner) *Given a knowledge element ke from KE, the learner performance degree in ke is defined by the function* LP_L.

$$LP_L : \begin{cases} KE \longrightarrow \{0,1,2,3\} \\ ke \longmapsto lp \in \{0,1,2,3\} \end{cases}$$

The learner performance degrees on MOOC prerequisites are determined by browsing the MOOC platform database looking for knowledge elements the learner has acquired previously on the same MOOC platform. If no significant evidence is collected this way – because the learner is a new user of the platform for example, then we ask the learner a few questions about the prerequisites in order to evaluate his performance degrees. Each MOOC week ends with a quiz. The learner's results on those quiz are used to weekly compute the learner performance degrees on MOOC learning objectives. The prerequisites and the learning objectives of the MOOC associated to the learner performance degrees are modeled respectively by the vectors $V_{P,Learner}$ and $V_{LO,Learner}$. A quite similar modeling process is proposed in (Sahebi et al., 2016). In (Figure 4) we represent the evolution of the knowledge elements in the learner profile during the MOOC. Before starting the MOOC, the learner performance degrees in MOOC prerequisites are stored in $V_{P,Learner}$. During the MOOC and after each week, $V_{LO,Learner}$ is updated with the new learner performance degrees acquired with the learning objectives provided in this week.

The learning style refers to the way a learner receives and processes information (Felder et al., 1988a). In the literature, many profiles are defined to analyze learners learning styles like Kolb (Kolb, 2005) and Felder and Silverman (Felder et al., 1988b). In our work, we use the frequently used, Index of Learning Style (ILS) questionnaire (Soloman and Felder, 1999). It was developed by Felder and Soloman to identify learning styles based on Felder and Silverman Learning style Model (Felder et al., 1988b). The FSLSM classifies learning styles along to four dimensions which are active/reflective, sensing/intuition, visual/verbal and sequential/global. In our work we also use the patterns introduced by (Fasihuddin et al., 2014) to identify the type of learning

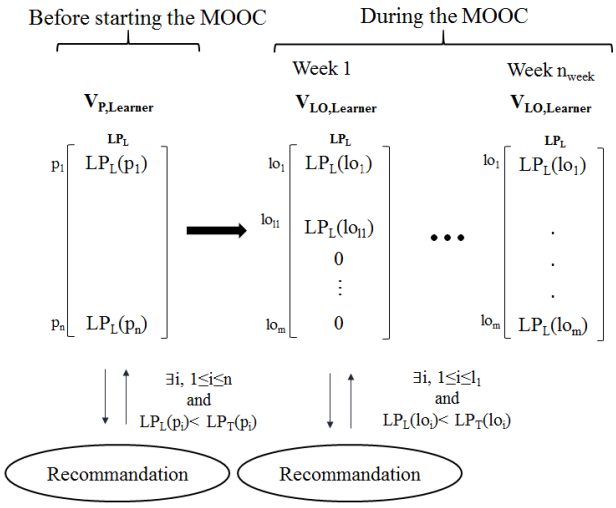

Figure 4: The evolution of the knowledge elements in the profile of the learner during the course.

resources to be provided to the learner based on his answers to the ILS questionnaire. For example, sensing learners prefer to get more examples and exercises (Fasihuddin et al., 2014). We model the learning style by using the term $S_{learner}$.

4.3 PreSearch Module

As indicated in (Figure 4), the recommendation process is triggered for a learner L at two different kind of steps of the MOOC.

Before Starting the MOOC. Let $LP_L(p)$ the performance degree of a learner L in $p \in P$, the recommendation process is triggered if:

$$LP_L(p) = V_{p,L}(p) < LP_T(p) = V_{p,MOOC}(p)$$

During the MOOC. At the end of the kth week of the MOOC, let a learner L who acquires $LP_L(lo)$ in $lo \in LO_k$, the recommendation process is triggered when:

$$LP_L(lo) = V_{LO,L}(lo) < LP_T(lo) = V_{LO,MOOC}(lo)$$

Where $V_{LO,MOOC}$ and $V_{LO,L}$ considered are those updated after the kth week.

The aim of this module is to select a set of candidate OERs dealing with the knowledge element for which the recommendation process has been triggered. In order to find these resources, the system performs a keyword search in metadata stored in external accessible repositories of OERs metadata (Hajri et al., 2015). The metadata used in this search is "the description of the resource" which introduces the subject and the global idea of the resource. The search is conducted using two keywords: the knowledge element and the domain of the MOOC. The combination between the knowledge element and the MOOC's domain comes from the fact that the same knowledge element may belong to many domains. For example the notion of "recursion" is used in a variety of disciplines as "computer science", "language", etc. But, we do not use just the exact form entered by the teacher to express the knowledge element and the MOOC's domain. In fact, as represented in (Figure 5), we introduce a module of synonyms detection[1] based on DBpedia[2] structured data that has been extracted from Wikipedia. Some possible synonyms of the knowledge element and the MOOC domain are inferred using this module. And thus we select descriptions including the knowledge element and the MOOC domain or at least one of their synonyms.

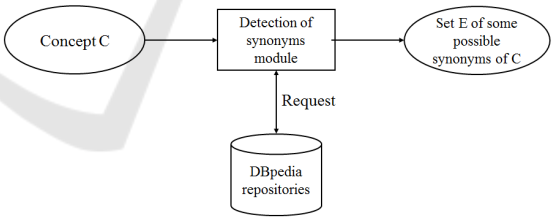

Figure 5: Detection of synonyms module.

In a formal way, let OER be the set of Open educational resources, $R \in OER$, $ke \in KE$, $Meta_R$ is the set of the metadata of R where $\text{Meta}_R = \{Descript_R, Lang_R, Prereq_R, etc\}$, $Sy_{D_{mooc}}$ the set of the synonyms of the domain of the MOOC, D_{mooc} and Sy_{ke} is the set of the synonyms of the ke generated by our module of synonyms detection.

(R is dealing with ke) If and only if
$((ke \in \text{Descript}_R)$ OR $(\exists i : Sy_{ke}[i] \in \text{Descript}_R))$

[1] https://davidallenfox.wordpress.com/2013/09/05/generating-synonyms/

[2] http://wiki.dbpedia.org/

$$\text{AND } ((D_{mooc} \in \text{Descript}_R) \text{ OR } (\exists j : Sy_{D_{mooc}}[j] \in \text{Descript}_R)).$$

To select these resources, our Pre Search module requests external repositories providing SPARQL endpoints. In order to manage the diversity of metadata schemas employed by these repositories, we use classes and properties as defined in the learning Object ontology of mapping (LOOM) introduced in (Hajri et al., 2015).

At the end of this PreSearch step, we have as result a set E of n candidate resources where $E = \{R_1, ..., R_n\}$.

4.4 Refinement Module

This refinement module aims to improve the results provided by the PreSearch module by taking into account more characteristics of the learner and the MOOC. This improvement consists in selecting resources that respect some characteristics of the MOOC and the learner and sorting them from the most adapted to the less adapted to the learner and the MOOC profiles. To this end, we define different criteria that reflect resources adaptation with the learner and the MOOC profiles.

A set of selection and sorting operations will be conducted based on these criteria. These operations are performed in three steps.

- Selection by constraints.
- Selection by semantic similarity.
- Sorting by requirements.

4.4.1 Selection by Constraints

The first step is selecting a subset of resources E' from the set E generated at the previous step. The subset E' contains resources respecting some constraints. These constraints are mandatory criteria that must be respected by recommended resources. In other words, without respecting these constraints, it will be difficult for the learner to assimilate the recommended resource or it will impact the MOOC follow up. In this context we define three constraints. The first one is: "The language of the resource must be known by the learner"**(C1)**. The second constraint is: "Resources must not require prerequisites not assimilated by the learner"**(C2)**. The third constraint is: "The resource has to bring a performance level which is greater than or equal to the the level defined in the MOOC" We consider that the recommendation process is initiated after the week m ($m \leq n_{week}$).

C1 Violation. Let $R \in E$, R doesn't respect the constraint C1 if:

$$(Lang_R \notin L)$$

Where L is the set of the languages known by the learner and $Lang_R$ the metadata presenting the language of the resource.

C2 Violation. Let $R \in E$, R doesn't respect the constraint C2 if:

$$\exists ke \in Prereq_R : (LP_L(ke) = 0) \vee (\exists i, LP_L(Syn_{ke}[i]) = 0)$$

Each resource $R \in E$ is represented by a vector $V_{KE,R}$ arranging the performance degrees acquired in each ke of the MOOC after following this resource.

Definition 3. (Resource Performance Degree) *The performance degree acquired by the learner with regard to a specific knowledge element ke, from KE, after following the resource R is defined by the function LP_R.*

$$LP_R : \begin{cases} KE \longrightarrow \{0,1,2,3\} \\ ke \longmapsto lp \in \{0,1,2,3\} \end{cases}$$

C3 Violation. As shown in (Figure 4), the recommendation process is triggered when $LP_T(ke) > LP_L(ke)$. So, Let $R \in E$, R doesn't respect the constraint C3:

$$LP_R(ke) < LP_T(ke).$$

As the OERs metadata don't specify the performance degrees of OREs, we need to estimate them from learners who have already used the resources. For a new OER we use two performance degrees 0: a resource doesn't deal with the knowledge element and 1: the resource deals with the knowledge element. Once the OER has been studied by a learner, its performance degree is computed using the learner's quiz results.

Definition 4. (A Knowledge Element Derived from a Resource)

(ke is provided by R) If and only if ($V_{KE,R}(ke) \neq 0$) If and only if

$$\{ke \in \text{Descript}(R) \vee \exists i : Syn_{ke}[i] \in \text{Descript}(R)\}.$$

4.4.2 Selection by Semantic Similarity

Once resources respecting the defined constraints have been identified, we select resources which are close to the initial query (the query defined in the PreSearch module). For that purpose, we calculate the similarity between selected resources and the initial query terms (IQT). The IQT are the terms used in

the initial query: the knowledge element, the domain of the MOOC and their synonyms generated by our module of synonyms detection. We denote IQT as a set.

$$IQT = \{T_1, ..., T_{nt}\}$$

Where T_{nt} is one of the initial query terms and nt is the total number of terms used in the initial query.

We start with using Term Frequency Inverse Document Frequency (TF-IDF) (Chowdhury, 2010) to identify the importance of IQT inside the selected resources descriptions. Each selected resource R is represented by a vector V_R.

$$V_R = (V_{R_{T_1}}, ..., V_{R_{T_{nt}}})$$

Where $V_{R_{T_1}}$ is the TF-IDF value of the term T_1 in the description of the resource R.
The IQT is also represented by a vector V_{IQT}.

$$V_{IQT} = (V_{IQT_{T_1}}, ..., V_{IQT_{T_{nt}}})$$

Where $V_{IQT_{T_1}}$ is the TF-IDF value of the term T_1 in the descriptions of all selected resources.

Then a cosine measure is employed to compute the similarity between each resource vector V_R and the initial query vector V_{IQT}. As result each selected resource R is characterized by one measure which is his cosine measure, CosSim(R). The resource R with higher value of CosSim(R), is the closest to the initial query.

At the end of this step, the result is the set E' of resources ordered by their semantic similarity with the IQT. We select the subset E'' of the first n ones to be the input of the next step. We take a limited number of resources because the objective of our recommendations is to help the learner to improve his knowledge level in a certain knowledge element by following at least one resource. For this reason we don't recommend a large number of OERs to him. The number n is defined arbitrarily and in our case study, n has been fixed to 5 but the teacher can change its value. We define the relation of preference ($\geq$).

Definition 5. (Preference Relation $\geq$) *Given two resources $R_1 \in OER$ and $R_2 \in OER$:*

$$R_1 \geq R_2 \text{ means } R_1 \text{ is at least as good as } R_2.$$

In this first step $R_1 \geq R_2 \Leftrightarrow \text{CosSim}(R_2) \geq \text{CosSim}(R_1)$.

4.4.3 Sorting by Requirements

The final step involves sorting resources based on requirements. The requirements are another set of criteria that we define to reflect coherence with the MOOC and learner characteristics. However these requirements are not mandatory criteria like the constraints of the first step. In other words, recommended resources may not comply with all the requirements but they are presented to the learner in an order depending on how much they satisfy the requirements. Let Req the set of requirements and n_{req} the total number of requirements.

Definition 6. (Score Function) *For each $req_i \in Req$, where $1 \leq i \leq n_{req}$, we define the score function U_i:*

$$U_i : \begin{cases} E'' \longrightarrow [0,1] \\ R_j \longmapsto a_i^j \end{cases}$$

U_i assigns a score a_i^j,between 0 and 1, to each candidate resource R_j depending on how much it satisfies the req_i. The scores a_i^j are calculated differently depending on the requirements req_i type.
For each requirement, the resource score represents its requirement satisfaction percent. Then we consider each requirement as a fuzzy set and the score of each resource as its membership degree to this set. We represent each resource R_j by a vector S_{R_j} whose components are the values of its score for each requirement.

$$S_{R_j} = (a_1^j, a_2^j, ..., a_2^{n_{req}})$$

The ideal resource id has a vector S_{id} whose components are equal to 1. This means that the resource meets all the requirements at 100%.
A weight value $p_i \in 1, 2, 3$ is assigned to each requirement in order to characterize its importance (1: less important, 2: important and 3: very important).
Initially all requirements have the same importance (p_i=3) but we give the teacher the possibility to change weights values of requirements defined to reflect coherence with the MOOC. As an example, we start with defining two requirements: "Recommended resources should respect the learning style of the learner" (Req_1) and "Recommended resources should have a 'typical Learning Time' similar or bellow the mean effort needed to assimilate a MOOC resource, as defined by the teacher." (Req_2).
For Req_1, we define the corresponding score function U_1 as below:

$$U_1(R) = \begin{cases} 1 & \text{if } \text{Typ}_R \in RT_L \\ 0 & \text{else.} \end{cases}$$

where Typ_R is the type of the resource $R \in E''$ and RT_L is the set of resources types corresponding to the learner L learning style.

Concerning Req_2, we define the corresponding score function U_2 as below:

$$U_2(R) = \begin{cases} 1 & \text{if } LT_R \leq ME_{W/R} \\ (\varepsilon + ME_{W/R} - LT_R)/\varepsilon & \text{elseif } LT_R \\ & \in [ME_{W/R}, ME_{W/R} + \varepsilon] \\ 0 & \text{else } LT_R \geq ME_{W/R} + \varepsilon \end{cases}$$

where $R \in E''$, $ME_{W/R}$ (Mean Effort Week) corresponds to the quotient of the week effort defined by the teacher and the number of the week resources and LT_R corresponds to the value of the metadata 'typical Learning Time' for R. The value ε is defined arbitrarily and has been fixed to ME_W in our case study.

To rank candidate resources, we use the Chebyshev distance to compute the distance between the ideal resource and each resource to recommend. The smaller the distance is the better the resource is. This distance is defined as below:

$$\mathrm{DCH}_{R_j,id} = max_{i \in n_{req}} \lambda_i |V_{R_j}[i] - V_{id}[i]|$$

where λ_i is defined as below:

$$\lambda_i = p_i / (\mathrm{Sup}_{R_j \in E''}(V_{R_j}[i]) - \mathrm{Inf}_{R_j \in E^*}(V_{R_j}[i]))$$

where E^* is a subset from E'' of candidate resources that not have a maximal satisfaction degree for any requirement.
In conclusion,

$$R_1 \geq R_2 \Leftrightarrow \mathrm{DCH}_{R_2,id} \geq \mathrm{DCH}_{R_1,id}$$

5 IMPLEMENTATION

In order to implement our solution, we choose edX as the MOOC platform. We opt for edX because:

- It is an open platform which is widely used.
- Its architecture is modular thanks to XBlocks (Kolukuluri, 2014).
- By chosing this platform, we can offer our solution to the vast community of OpenEdX users and hope to replicate experiments about personalization, then gather more data about its efficiency.

The XBlock is a component architecture developed in 2013 by edX, which allows developers to create independent course components (xBlocks). These components can be combined together to make an online course (Kolukuluri, 2014). The advantage of XBlocks is that they are deployable. The code that you write can be deployed in any instance of the edX Platform or other XBlock runtime application[3]. We found also that there is a recent focus on using these XBlocks to add personalization in MOOCs, for example the work (Li and Mitros, 2015) where a recommender XBblock was created in order to recommend resources for remediation in a MOOC. Once developed, each XBlock can be installed and added by the MOOC's creator, in the appropriate unit of the appropriate section of his MOOC[4]. In fact, Open edX organized the courses in a hierarchy of sections, subsections and units, where the unit is the smallest component in the MOOC.

For these reasons, we use XBlocks to implement our solution in edX. Three XBlocks have been implemented.

An XBlock to Calculate the Learner and the MOOC Profiles. This XBlock is meant to be added at the beginning of the MOOC's first section. It is responsible for collecting information about the learner and the MOOC by filling in forms. It is also the place where we ask questions to the learner in order to detect his learning style (see (Figure 6)).

An XBlock to Compute Recommendation at the Beginning of the MOOC. A second XBlock is developed to be added at the beginning of the first section, after the first XBlock. It is responsible for assessing the knowledge level of the learner in the MOOC's prerequisites by asking him some questions (an example for the prerequisite "structure data" (Figure 7)). Then if he doesn't answer the questions correctly, a set of OERs links are recommended to him. These links are ranked by descending order by satisfaction of the criteria defined at the previous section (Figure 8).

An XBlock to Compute Recommendation after Each MOOC's Section. A third XBlock is developed to be added at the end of each section. This XBlock computes recommendations of OERs to the learner based on his answers to the quiz presented at the end of the section. These OERs links are presented to the learner sorted based on the criteria we defined in the previous section.

6 EVALUATION PLAN

In order to evaluate our solution we launched three experiments. The purpose of the evaluation is to check whether the recommended resources are adequate with the criteria defined previously. In fact, the scalability and the versatility of OERs repositories means that OERs and their descriptions change dynamically and then compromise confronting the proposed resources with all the available OERs. Therefore, during the evaluation process, we focus on checking the

[3]https://open.edx.org/about-open-edx

[4]http://edx.readthedocs.org/projects/open-edx-building-and-running-a-course/en/latest/

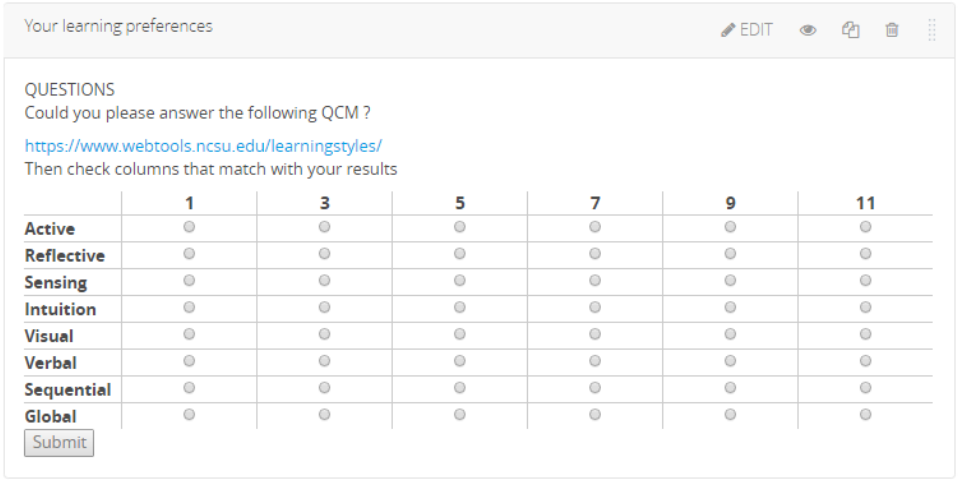

Figure 6: Interface to collect information about the learning preferences of the learner.

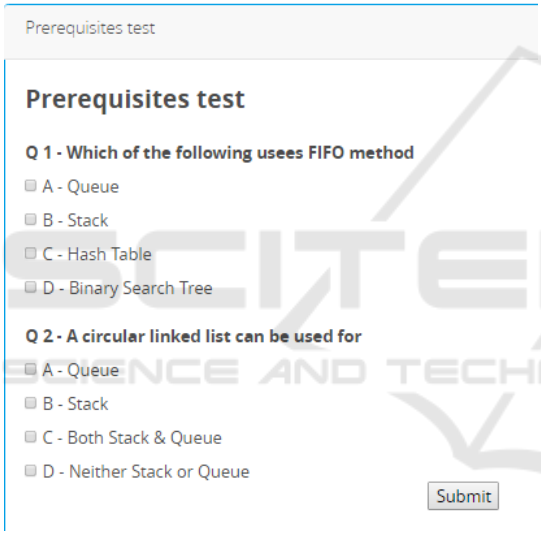

Figure 7: Interface for testing the assimilation of the prerequisites (example ”Data structure”).

adequacy of the proposed resources with the criteria defined to suit the learner profile and the coherence with the course.

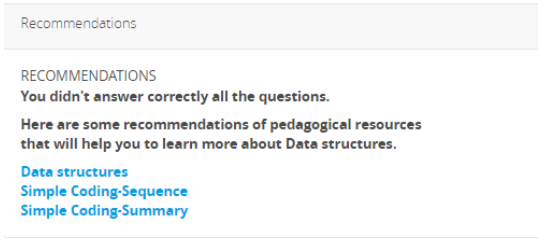

Figure 8: Interface for recommended resources.

6.1 Experiment 1

The objective of the first experiment is to show how much the resources selected by the initial query are relevant and deal with the appropriate knowledge element. In this experiment several experts (professors from CentraleSupelec) are invited to evaluate a set of resources selected using our initial query. We present several sets of resources to the experts. Each set of resources is collected for a specific knowledge element. Experts' evaluation consists on selecting relevant resources from each set. The aim of our system is to recommend a limited number of resources suitable with the profiles of the learner and the course in order to facilitate the exploitation of the recommended resources by the learner. Therefore, we are not interested in assessing whether our system recommends all the ”good” resources. Our objective is to assess whether the retrieved resources are ”good” resources. For this reason, we define a performance measure adapted to our objective which is Precision_{ad}. Precision_{ad} represents the percentage of relevant resources from the retrieved ones.

$$\text{Precision}_{ad} = \frac{|\{\text{Relevant resources}\}|}{|\{\text{Retrieved resources}\}|}$$

This rate is computed firstly to evaluate the set of resources resulting from the initial query and secondly to evaluate the subset of resources resulting from the filtering based on cosine similarity. The objective is to show that this filtering improves the adequacy of the recommended resources.

6.2 Experiment 2

This second experiment aims at evaluating the adequacy of recommended resources from the learner's point of view. The MOOC can be followed by anyone from all over the world and with any profile. So in order to collect a large number of users profiles without having to wait until they subscribe and follow a MOOC from the beginning to the end, we use the website Foule Factory[5]. It is a site which offers the possibility to ask the crowd to do some tasks as answering to questions or finding data, etc. In the first step we start with asking Foule Factoy users to answer a few questions in order to build their profiles. The first part of these questions is about their learning styles and their educational level. For the learning style, the user is invited to answer (ILS) questionnaire (Soloman and Felder, 1999). For the educational level we ask him about the highest level degree he obtained. In the second part, the user is invited to answer some questions taken from the different quiz proposed in the MOOC in order to evaluate his knowledge level in its learning objectives. More precisely, for each learning objective, we ask him three questions. Then, for each learning objective, if one of the questions is not answered correctly, we recommend to the user a set of resources supposed to help him to improve his knowledge level in this knowledge element. The learner is invited to study at least one the recommended resources. Then we ask him these questions: (1) which resource did he study, (2) why did he choose this resource and not the other ones and (3) did the resources allow him to learn something new. The user is also invited to answer for a second time to the questions taken from the MOOC quiz to re-assess his knowledge level.

Based on these results we are going to evaluate whether the resources offered by our solution improve the knowledge level of learner in a certain knowledge element. We will also evaluate how much the learner chose resources respecting his learning style over other resources.

6.3 Experiment 3

This third experiment is about evaluating the order in which the recommended resources appear to the learner. In order to ensure this, on the one hand, the experts are invited to allocate a score (in $\{1,2,3,4,5\}$) for each relevant resource depending on some criteria. These criteria are: (1) how much the resource provides the level of knowledge as required, (2) how much the learning duration needed to assimilate the resource respects the effort duration mentioned in the MOOC, (3) how much the resource deals with the appropriate knowledge element and (4) how much the granularity of the resources is in line with the MOOC lessons. An average score is then calculated for each resource. On the other hand, Foule Factory users are invited also to give a score (in $\{1,2,3,4,5\}$) for each relevant resource based on two criteria: (1) how much this resource is easy to follow and (2) how much this resource is pleasant to follow based on his learning preferences. An average score is calculated for each resource. In this experiment we use the measure DCG to evaluate the order of the recommended resources in the list proposed to learner:

$$\mathrm{DCG} = \sum(2B(i) - 1/\log_2(1+i))$$

Where $B(i)$ is the product of the scores given by the teacher and the learner for the resource i of the list.

The objective of this experiment is to compare the order in which the resources recommended by our system are presented and the order deduced from the scores attributed by Foule Factory users and experts.

This experiment will also allow as to evaluate, on the one hand, whether the recommended resources are suitable to the learning style and the educational level of the learner. On the other hand it will allow us to evaluate whether they are suitable to the granularity, the effort duration and the knowledge level fixed for MOOC resources. In other words, we will evaluate how much the resources recommended to a learner are adapted both to his characteristics and to the MOOC's characteristics, as defined in their profiles.

7 CONCLUSIONS

This paper introduces a generic solution to provide recommendations of OERs in an online course. The MORS system described in this paper is especially devoted to MOOCs. By integrating MORS in a MOOC platform, MOOC and learner profiles are computed by extracting relevant information obtained from the platform. These profiles are also updated dynamically during the course. When our system detects a lack of knowledge in a certain topic, the PreSearch module queries OERs endpoints to select OERs dealing with the concerned topic. Then the refinement module applies selection and ranking depending on different criteria in order to offer to the learner a set of ordered OERs that match with the MOOC and the learner specificities as represented in their profiles.

Our solution uses the OERS Metadata stored in accessible repositories. So, the more these metadata

[5]https://www.foulefactory.com/

are available and filled in, the better the quality of recommended resources is. Nothing stands the use of our system in other types of OLEs as long as the learner and the MOOC profiles data can be extracted.
Our objective in the intermediate future is to carry out the experiments as defined previously in this paper in order to assess the adequacy of the recommended resources with the learner and the MOOC profiles : how much the resource we recommend to the learner is adapted to his characteristics while respecting the characteristics of the MOOC.
The results of these experiments will also allow us to improve our system.
Then in the distant future we manage to assess our solution in a more comprehensive way. More precisely our objective is to integrate our recommender system in an existing MOOC and to assess how it will be working under real conditions. In this evaluation we will be interested with the interactions between our recommender system and learners with different profiles and how much the recommendations we offer to learners support them during the MOOC. Another future work is to use other characteristics of the learner when computing recommendations to him. One of these characteristics is the context of the learner that can represent an important criteria to be taken into consideration in the refinement module. In fact depending on the nationality of the learner for example several information can change as the teaching styles or the educational levels.

REFERENCES

Agrawal, A., Venkatraman, J., Leonard, S., and Paepcke, A. (2015). Youedu: addressing confusion in mooc discussion forums by recommending instructional video clips.

Alario-Hoyos, C., Leony, D., Estévez-Ayres, I., Pérez-Sanagustín, M., Gutiérrez-Rojas, I., and Kloos, C. D. (2014). Adaptive planner for facilitating the management of tasks in moocs. In *V Congreso Internacional sobre Calidad y Accesibilidad de la Formación Virtual, CAFVIR*, pages 517–522.

Bansal, N. (2013). Adaptive recommendation system for mooc. *Indian Institute of Technology*, pages 1–40.

Chowdhury, G. G. (2010). *Introduction to modern information retrieval*. Facet publishing.

Fasihuddin, H. A., Skinner, G. D., and Athauda, R. I. (2014). Personalizing open learning environments through the adaptation to learning styles. ICITA.

Felder, R. M., Silverman, L. K., et al. (1988a). Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681.

Felder, R. M., Silverman, L. K., et al. (1988b). Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681.

Hajri, H., Bourda, Y., and Popineau, F. (2015). Querying repositories of oer descriptions: The challenge of educational metadata schemas diversity. In *Design for Teaching and Learning in a Networked World*, pages 582–586. Springer.

Imran, H., Belghis-Zadeh, M., Chang, T.-W., Graf, S., et al. (2016). Plors: a personalized learning object recommender system. *Vietnam Journal of Computer Science*, 3(1):3–13.

Kolb, A. Y. (2005). The kolb learning style inventory-version 3.1 2005 technical specifications. *Boston, MA: Hay Resource Direct*, 200:72.

Kolukuluri, S. (2014). *XBlock-Courseware Component Architecture*. PhD thesis, Indian Institute of Technology, Bombay Mumbai.

Li, S.-W. D. and Mitros, P. (2015). Learnersourced recommendations for remediation. In *Advanced Learning Technologies (ICALT), 2015 IEEE 15th International Conference on*, pages 411–412. IEEE.

Maran, V., de Oliveira, J. P. M., Pietrobon, R., and Augustin, I. (2015). Ontology network definition for motivational interviewing learning driven by semantic context-awareness. In *Computer-Based Medical Systems (CBMS), 2015 IEEE 28th International Symposium on*, pages 264–269. IEEE.

Paquette, G., Mariño, O., Rogozan, D., and Léonard, M. (2015). Competency-based personalization for massive online learning. *Smart Learning Environments*, 2(1):4.

Sahebi, S., Lin, Y.-R., and Brusilovsky, P. (2016). Tensor factorization for student modeling and performance prediction in unstructured domain. In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 502–506. IEDMS.

Soloman, B. A. and Felder, R. M. (1999). Index of learning styles questionnaire. *Retrieved March*, 26:2003.

Sunar, A. S., Abdullah, N. A., White, S., and Davis, H. C. (2015). Personalisation of moocs: The state of the art.