2017 IEEE 17th International Conference on Advanced Learning Technologies

MORS:
A system for recommending OERs in a MOOC

***Abstract*—Personalization in the field of Technology Enhanced Learning (TEL) is a topic that received a lot of concern by researchers. At the same time, there is a growing amount of Open Educational Resources (OER) indexed according to the W3C standards. Relevant OERs can usefully complement the contents delivered to a learner during an online course. Computing the best OERs to offer to the learner at each point of his course is an aspect of personalization that we address in this paper. We designed our MORS system to solve this problem in the context of Massive Open Online Courses (MOOC). Our MORS system described in this paper, is based on a learner profile, on metadata describing the course and on a carefully crafted process to query the SparQL endpoints for OERs.**

***Index Terms*—Personalization, learning, learner profile, OER, recommendation, MOOC.**

I. Introduction

The rapid progress of MOOCs in the recent years is considered as a revolution in TEL. The principal goal of MOOCS is to take advantage of technologies, especially the web, in order to disrupt traditional modes of education and to allow free or low-cost access to high quality education [1] [2], by including many types of resources like videos, text, quiz, etc.

However, one of the major problems related to MOOCs is learners diversity. In fact, the same course can be followed by large numbers of learners all over the world with different ages, competencies, goals, etc. So providing an efficient one-size-fits-all learning content is neither easy nor obvious. In fact, MOOC creators have to offer the most suitable content, the one that will match everyone needs and expectations. One solution is to personalize the MOOC depending on learners profiles so that it becomes suitable to everyone.

Furthermore, more and more OERs are available on the Web. Indeed, producing quality OERs is costly in term of time and they should be reused in a context different from the one for which they were created. But their reuse is based on the ease with which their descriptions are available. This is why we give priority to OER having their metadata respecting the principles of Linked Open Data (LOD).

In this paper, we introduce the MOOC-based OER recommender system (MORS) which can be integrated in MOOCs platforms to provide personalization to learners. Our system recommends remedial OERs to the learner depending on his profile while remaining coherent with the MOOC.

The organization of this paper is as follows. The following section presents a review of previous works on MOOC personalization. Section 2 discusses the architecture of MORS. Section 3 concludes the paper and presents future directions.

II. Related work

Even though Personalization in TEL is a research topic with a long history, studies on MOOCs personalization have started since 2013 [3]. In this context, different personalization approaches have been adopted. One of the most popular techniques is recommender systems. For example, the approach proposed by [4] which 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, since the beginning, dedicated to MOOCs in a specific topic and can't be considered as a generic solution. There is also the approach introduced by [5], it targets learners who post a question in MOOC discussion which reflects a confusion and recommends educational videos related to the confusion subject. Also, [6] recommends additional learning activities to learners who shows a lack of knowledge in a particular subject. However [5] and [6] generate their recommendations from MOOCs internal resources. 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 [7]. In fact, it recommends to the learner a set of MOOCs which mostly match his learning objectives. Another approach [8] 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 only one characteristic is considered in the learner profile which might lead to a less specific personalization. On the other hand, even if external resources are recommended, these approaches don't consider MOOC specificities and the recommended resources may be incoherent with the MOOC. Since these external resources will complement the MOOC representing the initial path, it is important to select resources that are coherent with the content of this path. These selected resources have to integrate the MOOC while respecting its specific characteristics. 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 based on different learner characteristics and also MOOC specificities.

III. System architecture

The MORS system provides recommendations of OERs to MOOCs learners. The architecture of MORS is shown in (Fig.1). The four modules of the system are: MOOC modelling module, learner modelling module, PreSearch module and refinement module.

A. MOOC modelling module

This module aims at generating the MOOC model using the information entered by the teacher when he creates the MOOC. These information are the knowledge elements and the domain

2161-377X/17 $31.00 © 2017 IEEE
DOI 10.1109/ICALT.2017.89

IEEE computer society

Authorized licensed use limited to: Centrale Supelec. Downloaded on August 07,2021 at 09:44:43 UTC from IEEE Xplore. Restrictions apply.

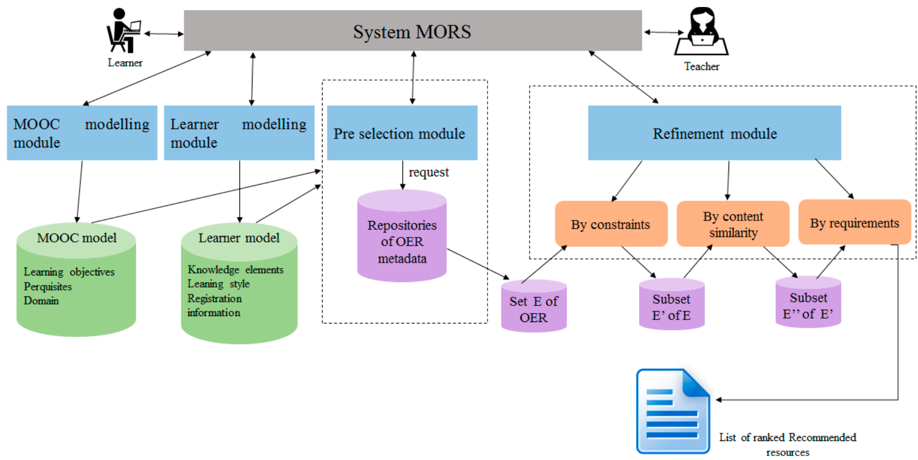

Fig. 1. The architecture of MORS

of the MOOC. For the knowledge elements, there are two types: learning objectives of each MOOC week and prerequisites of the MOOC. The teacher defines the learning objectives provided by each course's section and the prerequisites needed to follow the course. He defines also with each knowledge element a performance degree as introduced in [9] (1: beginner, 2: intermediate and 3: expert). For the prerequisites, it corresponds to the performance degree required in each prerequisite. For the learning objectives, it corresponds to the performance degree expected to be acquired in each learning objective after following the section providing this learning objective.

The MOOC domain corresponds to the subject of the course like Architecture, Computer Science, Physics, etc.

1) Learner modelling module: This module is responsible for generating and updating the learner model during the MOOC. The learner model contains his knowledge elements, his learning style and other registration information as the languages known by the learner. Concerning 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.

In his model, the learner has a performance degree in each knowledge element that will be updated during the MOOC. First for the prerequisites, learner performance degrees are recovered from the database storing all the knowledge elements acquired by the learner by using the MOOC platform. Otherwise, some questions dealing with the appropriate knowledge element are presented to the learner in order to deduce his learning performance. Then, for the learning objectives of each course's section, the performance degree acquired by the learner is deduced from his answers to the quiz introduced at the end of each section. Throughout the course, the learner model is updated with new performance degrees depending on the knowledge elements acquired by the learner and with what degree of performance.

Concerning learning style, it refers to the way a learner receives and processes information [10]. In the literature, many models are defined to analyse learners learning styles like Kolb [11] and Felder and Silverman [10]. In our work, we use the frequently used, Index of Learning Style (ILS) questionnaire [12]. It was developed by Felder and Soloman to identify learning styles based on Felder and Silverman Learning style Model [10]. The FSLSM classifies learning styles into four dimensions which are active/reflective, sensing/intuition, visual/verbal and sequential/global.

2) PreSearch module: The recommendation process is triggered for a learner in two steps of the MOOC:
(1) Before starting the MOOC: if the learner has in his model a performance degree, in one of the course's prerequisites, which is lower than the performance degree defined in the MOOC model for the same prerequisite.
(2) A the end of a MOOC's section: if the learner has in his model a performance degree in one of this section's learning objectives, which is lower than the performance degree defined in the MOOC model for the same learning objective at the same point of the course.

The PreSearch module selects a set of candidate OERs dealing with the knowledge element for which the recommendation process has been triggered. To find these resources, we exploit their metadata stored in external repositories [13]. In order to query OERs descriptions repositories, on the one hand, we use two metadata elements: "the description of the resource" and "the language of the resource". On the other hand, we introduce a module of synonyms detection[1] based on DBpedia[2] structured data that has been extracted from Wikipedia. And then the PreSearch module requests external OERs descriptions repositories having a SPARQL endpoints. In order to manage the diversity of metadata schemas employed by these repositories, we use classes and properties as defined in the ontology of mapping LOOM introduced in [13]. Selected OERs are those that include in their descriptions the knowledge element or one of his synonyms, the course's domain or one of his suynonyms and proposed in a language which is known by the learner.

The recommended OER must also bring a performance degree which is greater or equal to the performance degree fixed in the MOOC model. Therefore, each OER is modelled

[1] https://davidallenfox.wordpress.com/2013/09/05/generating-synonyms/
[2] http://wiki.dbpedia.org/

Authorized licensed use limited to: Centrale Supelec. Downloaded on August 07,2021 at 09:44:43 UTC from IEEE Xplore. Restrictions apply.

by the performance degrees expected to be acquired after following the resource, in each knowledge element of the MOOC. Now as the performance degrees of resources in OERs repositories are not defined in their metadata, we plan to collect the results from learners who have already used the resources. So initially we start with using two performance degrees 0: a resource doesn't deal with the knowledge and 1: the resource deals with the knowledge.

3) Refinement module: The refinement module consists of three steps.

Step 1: Selecting a subset containing the most relevant resources. This step consists on selecting a subset from the set of OERs generated by the PreSearch module. This selection is based firstly on a list of constraints. The resources selected in this step are OERs respecting some mandatory criteria. Without respecting these criteria, following the resource will be a difficult task for the learner or it will affect the follow up of the MOOC. Secondly, we select the resources which are closer to the initial query. To measure the distance between each resource and the initial query, we compute the cosine similarity between the two of them. We start by identifying the importance of the initial query terms in resources descriptions. By terms we mean the knowledge element, the domain of the MOOC and their synonyms generated by our module of synonyms detection. To compute this importance, we use Term Frequency Inverse Document Frequency (TF-IDF) [14]. Then a cosine measure is employed to compute the similarity between each resource vector and the initial query vector containing TF-IDF values. The resource R with higher value of cosine similarity is the closest to initial query. At the end of this step, we select the set of the closer resources. The number of the resources included in the set is defined in an arbitrary way.

Step 2: Sorting the resources. In this final step, the resources provided by the previous step are sorted based on some requirements. These requirements are criteria that we define to reflect some course's specificities and learner profile elements. Resources respecting these criteria are more coherent with the MOOC and more suitable to the learner profile. 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. For each defined requirement, we define a score function that associates to each resource a score value between 0 and 1 depending on how much the resource satisfies the requirement : for each prerequisite the score attributed to each resource corresponds to its requirement satisfaction percent. A weight is associated with each defined requirement reflecting his importance in the recommendation process. Initially we consider that all the requirements has the same importance but we offer to the teacher the possibility to change the weights of the requirements defined to reflect the specificities of the course. To order 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.

IV. Evaluation plan

We will start with checking the adequacy of the resources recommended by MORS system with the learner and the MOOC profiles. Therefore we manage to assess the recommended resources from two points of view: experts and learners. During the evaluation process, we will evaluate the relevance of the recommended resources before and after the refinement module. Then we will assess the final order in which the resources are presented to the learner.

V. Conclusion

This paper introduces the MORS system which provides recommendations of OERs to MOOCs learners. Our approach is based on two principles: (i) modelling the MOOC and the learner and (ii) a process to query OERs endpoints and filter out the results based on the learner profile and the MOOC specificities. The learner and the MOOC models are built by extracting the relevant information from the MOOC platform. These models are also updated dynamically during the MOOC. Albeit the query process we set up is well-founded, the efficiency of the system relies on the availability of OERs and the quality of their metadata. This efficiency needs to be assessed and this is a future and scheduled work. Nothing refrains from using the very same query process in other platforms like Moodle which are targeted at wider usages than merely MOOCs ; all kind of courses are subject to this recommendation as long as the courses metadata and learner profile can be extracted.

References

[1] T. R. Liyanagunawardena, A. A. Adams, and S. A. Williams, "Moocs: A systematic study of the published literature 2008-2012," *The International Review of Research in Open and Distributed Learning*, vol. 14, no. 3, pp. 202–227, 2013.

[2] N. M. Castillo, J. Lee, F. T. Zahra, and D. A. Wagner, "Moocs for development: Trends, challenges, and opportunities," *Information Technologies & International Development*, vol. 11, no. 2, pp. 35–42, 2015.

[3] A. S. Sunar, N. A. Abdullah, S. White, and H. C. Davis, "Personalisation of moocs: The state of the art," 2015.

[4] V. Maran, J. Palazzo Moreira de Oliveira, R. Pietrobon, and I. Augustin, "Ontology network definition for motivational interviewing learning driven by semantic context-awareness," in *Computer-Based Medical Systems (CBMS)*. IEEE, 2015, pp. 264–269.

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

[6] N. Bansal, "Adaptive recommendation system for mooc," Ph.D. dissertation, Indian Institute of Technology, Bombay, 2013.

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

[8] G. Paquette, O. Mariño, D. Rogozan, and M. Léonard, "Competency-based personalization for massive online learning," *Smart Learning Environments*, vol. 2, no. 1, pp. 1–19, 2015.

[9] H. Imran, M. Belghis-Zadeh, T.-W. Chang, S. Graf *et al.*, "Plors: a personalized learning object recommender system," *Vietnam Journal of Computer Science*, vol. 3, no. 1, pp. 3–13, 2016.

[10] R. M. Felder, L. K. Silverman *et al.*, "Learning and teaching styles in engineering education," *Engineering education*, vol. 78, no. 7, pp. 674–681, 1988.

[11] A. Y. Kolb, "The kolb learning style inventory-version 3.1 2005 technical specifications," *Boston, MA: Hay Resource Direct*, vol. 200, p. 72, 2005.

[12] B. A. Soloman and R. M. Felder, "Index of learning styles questionnaire," *NC State University. Available online at: http://www. engr. ncsu. edu/learningstyles/ilsweb. html (last visited on 14.05. 2010)*, vol. 70, 2005.

[13] H. Hajri, Y. Bourda, and F. Popineau, "Querying repositories of oer descriptions: The challenge of educational metadata schemas diversity," in *Design for Teaching and Learning in a Networked World*. Springer, 2015, pp. 582–586.

[14] G. Chowdhury, *Introduction to modern information retrieval*. Facet publishing, 2010.

Authorized licensed use limited to: Centrale Supelec. Downloaded on August 07,2021 at 09:44:43 UTC from IEEE Xplore. Restrictions apply.