Recommendation System for Knowledge Acquisition in MOOCs Ecosystems

Rodrigo Campos[1] (author), Rodrigo Pereira dos Santos[2] (co-advisor), Jonice Oliveira[1] (advisor)

[1]PPGI – Universidade Federal do Rio de Janeiro (UFRJ)
Av. Athos da Silveira Ramos, 274, Bloco E, Ilha do Fundão
Cidade Universitária – Rio de Janeiro, RJ, Brasil

[2]PPGI – Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Avenida Pasteur, 458 – Urca – Rio de Janeiro, RJ, Brasil

rodrigo.campos@ufrj.br, rps@uniriotec.br, jonice@dcc.ufrj.br

Abstract. *In recent years, students face difficulties in choosing the best content from the online distance learning of MOOCs (Massive Open Online Courses). The emerged recommendations systems to solve this problem do not identify the student's prior knowledge broadly. From this problem, the main contribution of this work is the identification and reduction of the students' knowledge gap in MOOCs. As such, in this Master's thesis, we model and analyze the MOOCs ecosystems and propose a solution for recommending parts of courses. Based on a set of three experiments, we verify that our recommendations are accurate, useful and reliable. We also present new content to fill the knowledge gap of users as the main contribution of this work to the state of the art.*

1. Introduction

Massive Open Online Courses (known as MOOCs) are courses that integrate the concept of open online education with specific characteristics, e.g. having no limit on the number of participants and students having control over the class time. MOOCs are stored and delivered by providers. In this context, learning is done through a web platform. Some platforms have gained visibility in recent years, such as Coursera, Udacity, and edX.

These facilities attracted universities that began to invest in courses over such platforms. In the last survey about MOOCs [Shah, 2019], there were more than 11,000 courses with more than 100 million users enrolled in any course. This number caught the attention of the software engineering and information systems fields. Shanyun et al. (2015) raised the need to understand MOOCs within an ecological context, composed of a learning environment and a target community. Therefore, it would be possible to understand the interactions between these elements and such interactions allows the creation of an ecosystem.

However, a current challenge is the identification of MOOCs particularities, largely due to the sudden growth of MOOCs' offering in the existing platforms. As such, the references used to build these ecosystems came from the Virtual Learning Environments (VLE) and not necessarily reflect the processes of this context. In addition, that growth can also create other problems, e.g. lack of interaction among students, students without feedback, and/or increase of dropout rate. In the Master's thesis, we decided to propose solutions to a problem regarding the support to users to achieve their

own specific goals and reduce their knowledge gaps, i.e., acquire new knowledge according to his/her interests. Users want to acquire some knowledge, but there is no guidance on what courses are the most appropriate.

In this context, the main research question (RQ) in this work is:

RQ1: How to identify and reduce knowledge gaps in the MOOCs ecosystems?

Moreover, our work investigates the following alternative research questions:

RQ2: What are the existing works about recommender systems applied to MOOCs?

RQ3: What are the main challenges in recommender systems applied to MOOCs?

RQ4: What are the main actors in the MOOCs ecosystems, and how they relate to each other?

In this context, we investigate how to combine recommendation systems and MOOCs providers' platforms to help users based on a combination of course modules[1]. This work also investigates MOOCs ecosystems' characteristics, exploring how this perspective can support the providers' basic processes. We argue that this work is relevant to Information Systems and Software Engineering since it covers studies on recommendation systems, knowledge reuse management, and software ecosystems (SECO). In this work, we improve the recommendation process and optimize knowledge management, ensuring benefits for students within these ecosystems.

2. Methodology

The research methodology adopted in this work follows three phases: A) literature investigation enhanced with a specific Systematic Mapping Study (SMS) in the topic; B) specification and implementation of a content-based recommendation system for MOOCs ecosystems; and C) evaluation of our proposal based on applying a quali-quantitative method.

In order to follow the phase B of our research methodology and allow us to answer RQ1, some important steps were identified in this work:

a) Step #1: contribute to a better understanding of MOOCs ecosystems, modeling MOOCs domain based on a SECO approach. Thereby, it can bring benefits and solutions related to knowledge reuse in MOOCs;

b) Step #2: define the conceptual model of the proposed recommendation system as well as the techniques used to support the processes of data extraction, data storage, provider data union, modeling, and labeling topics, and recommendation itself;

c) Step #3: after specifying the recommendation system, the goal is the creation, i.e., implementation of a system to recommend course modules to users. At this step, the implementation process is described in detail from the specification of techniques, environment, code, and system use.

1 These recommendations apply for full courses in the case of providers that do not partition learning

3. Related Work

A SMS was executed [Campos, 2019] to map other works that addressed recommendation systems in the MOOCs scenario. Moreover, it was investigated which tools or recommendation systems techniques are available in MOOCs context. One of the techniques identified in the SMS was the topic modeling (applied in our solution). This technique is presented in the literature applied to collaborative filtering (CF), content-based, and topic-specific recommendations.

After analyzing our SMS results, we observed that the solutions proposed by Apaza et al. (2014) and Bhatt et al. (2018) - both using a content-based recommendation to MOOCs - are more related to our proposal than other works. However, given the limitation of no access to the source code or datasets from those works, it was not feasible to compare them with our work in a greater level of detail, e.g. simulating how our dataset would behave in these related solutions to verify whether recommendations would get better results regarding the users' interests. The metrics used to evaluate these studies are also not related to our metrics, except 'novelty' in Bhatt et al. (2018) solution. Even though there is no comparison at this level of metrics and results, we present a comparative analysis of characteristics between the works in Table 1 (we referred to our proposal as "RS"- the acronym for "Recommendation System").

We can be highlight that Jing and Tang (2017), Song et al. (2017) and Li et al. (2018) apply CF approach because these works consider recommendations in scenarios where there is (and it is possible to extract) interaction between users. So, such solutions are not applicable in scenarios without available interactions. Jing and Tang (2017) indicate an open research opportunity to recommend not only complete courses, but also the low level of contents. Apaza et al. (2014) indicate a need to apply changes in the topic modeling technique (e.g. automating the calculation of topics in the model), enabling the scalability of the model that is the input of the recommendation process.

Our solution differs from the others identified in the literature by treating student's profiles from multiple MOOCs platforms aiming at the reduction of their knowledge gap. This is possible because we create a new method for extracting API data from multiple MOOCs providers in JSON format. It also differs in the recommendation process and the work construction since our solution considers parts of course recommendations, delivering packages of personalized modules according to users' knowledge gap. Moreover, we model and analyze the target context as MOOCs ecosystems based on the software ecosystems (SECO) approach, in order to balance the ecological environment and strengthen interactions in the conceptual model of the proposed recommendation system.

Based on the literature investigation and considering our SMS [Campos, 2019], we answered RQ2, i.e., *"What are the existing works in recommender systems applied to MOOCs?*", and RQ3, i.e., "*What are the main challenges in recommender systems applied to MOOCs?*". Finally, RQ4 ("*What are the main actors in the MOOCs ecosystems, and how they are related?*") was answered by modeling MOOCs Ecosystems and mainly: a) verifying the particularities of this ecosystem; b) defining the interactions between their actors; c) identifying actors' roles; d) identifying benefits of the SECO approach; and e) by making use of such approach to define and implement the proposed recommendation system.

Table 1. Comparison between topic modeling related work and the web-based recommendation system (RS) proposed in our research. Source: [Campos, 2019]

		[Song et al., 2017]	[Jing and Tang, 2017]	[Li et al., 2018]	[Bhatt et al., 2018]	[Apaza et al., 2014]	[Wang et al., 2015]	RS
Recommendation Approach	CF	X	X	X				
	Content-Based with LDA[2]				X	X		
	Topic Specific						X	
	Content-Based with NMF							X
Output	Threads	X						
	Open Education Resources (OER)						X	
	Videos				X			
	Courses		X			X		
	Tags and Courses			X				
	Courses and its parts (module, relevant content)							X
Type	System				X	X	X	X
	Network			X				
	Algorithm Framework		X					
	Learning Assistance	X						
Provider	XuetangX		X	X				
	Coursera	X				X		
	Multiple				X		X	X

4. Proposed Recommendation System Conceptual Model

As a basis for the creation of the conceptual model of our proposed solution, it was identified the key stakeholders involved in MOOCs in order to define the actors and their respective roles in the ecosystem. Thereafter, we define a conceptual model shown in Figure 1. In addition to the development of sustainable MOOCs, other benefits of a SECO approach applied to this domain can be pointed as follows [Barbosa et al., 2013]:

a) It supports software evolution and innovation in organizations, increasing the attractiveness (new players) and promoting the platform success;

b) It helps to analyze and understand software architecture to decide which platform can be used - the largest benefit exploited in our work;

c) It supports knowledge sharing through multiple and independent entities, strengthening cooperation;

[2] Latent Dirichlet Allocation

d) It helps business identification tasks, product architecture design, or risk identification.

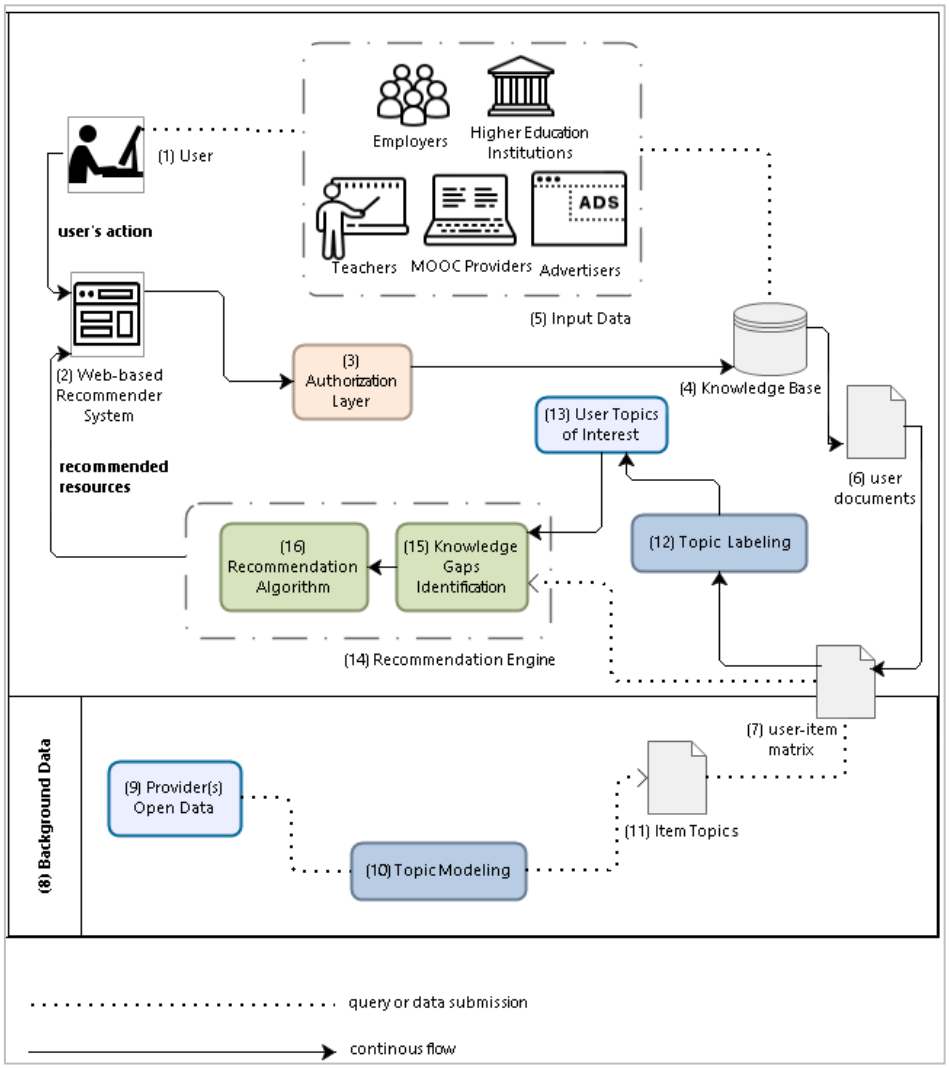

Figure 1. Proposed conceptual model. Source: [Campos, 2019]

Directed arrows in Figure 1 determine the recommendation process flow. Dashed lines indicate queries or data submissions. The process is divided into layers and steps:

a) A user (1) accesses the Web-Based Recommendation System (2). Then, to make a recommendation, the system requests access to the user's data through the authentication layer (3). Next, it is possible to access the user's Knowledge Base (4) that holds information from the ecosystem's Input Data (5). Input data (5) contain authorized information (e.g., user's competencies), curriculum and other data from MOOCs providers. Such information represents possible interactions that such users have in the ecosystem. Therefore, this structure (5) contemplates the actors that interact with students;

b) Next, the system selects the user's history and create a set of documents (6). This data serves as inputs to the user-item matrix (7). The "user" of the matrix (7) is composed of documents from the user's Knowledge Base (4);

c) To create the “item” of the matrix (7), it is used data from the Background Data (8) layer (i.e., system's information before starting the recommendation process). This layer has data from different MOOCs providers (9), then allowing a broader recommendation. With the providers’ data selected, the topic modeling method (10) creates the item topics (11). Using these topics, it is possible to complete the “item” of the user-item matrix (7);

d) This matrix (7) is input to the techniques responsible for labeling topics (12). In our approach, the labels help to define item topics (11) and also to create the user's topics of interest (13);

e) Once the topics and layers are created, the recommendation engine layer (14) applies two procedures using the user-item matrix. First, the Knowledge Gaps Identification (15) collects the user’s existing/desired skills and identifies the current knowledge gap. Then, the Recommendation Algorithm (16) is applied in order to find similarities among the item’s topics (11) and user’s documents (6). Finally, it ranks results and sends resources (recommendations/user’s topics of interest) back to the system (2).

5. Proposed Recommendation System Implementation

One of the stages of our work is to map the availability and openness of users’ data in each MOOC provider. An initial search was performed in the technical literature. We extracted data from three providers selected by convenience based on extraction format standardization (JSON through the API), open data availability, documentation richness, and the language of the courses (English). As such, the selected providers were Khan Academy, Udemy, and edX. The implementation of our solution can be applied to other MOOCs providers, as long as it supports data availability and data format compatibility. This possibility for cooperation between independent entities/providers is due to the use of the SECO approach. As such, other developers can contribute to (and extend) the conceptual model of our recommendation system.

With the necessary data (documents) properly extracted from each provider, it is necessary to store the set in an integrated way. We use a document-oriented database to store these documents since the conceptual model of our solution is based on the fact that the recommendation system is extended by other developers. To do so, criteria such as scalability, consistency and availability must be considered. Document-oriented databases are recommended for these cases, allowing the growth in the number of fields or even new features to be added [Corbellini et al., 2017]. MongoDB (a free Document-oriented database) was chosen for data storage. The main reason is that it integrates data in JSON format, finding the BSON – binary that takes up less space and is faster.

Once the document extraction algorithm is implemented according to each provider’s specificity, another algorithm models the topics using the Non-negative Matrix Factorization (NMF) technique. Algorithm 1 shows the steps of this implementation. The procedures executed to find the value of the ideal number of topics represented by the variable *k* (lines 6 to 10 of Algorithm 1) are based on the stability analysis approach for automatic calculation of *k*, proposed by Greene et al. (2014, apud Nolasco, 2016). As such, tests to find *k* are based on applying the modeling method to different values of *k* (given a minimum and maximum *k,* called respectively *kmin* and *kmax* at lines7 of Algorithm 1) until a *k* that reproduces a topic coherence value higher than the others.

Topic coherence verifies how semantically the terms of a topic relates to each other. Therefore, there is no need for human interference with the method for selecting the number of topics that best fit the model.

Algorithm 1: Automatic NMF topic modeling integrated with providers. Based on [Greene et al., 2014]

Input: JSON data dt from providers where each row represents a module and list of stopwords sw
Output: W (document-topic matrix) and H (topic-term matrix)

```
list = TransformDataIntoUTF8List(dt)
tokenizer = LemmaTokenizer( )
Vectorizer = TfidfVectorizer(sw, tokenizer)
A = CreateDocumentTermMatrix
vocabulary = Vectorizer(A)
kmin, kmax = SetValues( ) #integer is required
for k = kmin to kmax do
    CalculateCoherence(k)
    coherences = [k]
best_k = GetBestK(coherences)
W = GenerateDocumentTopicMatrix(A, best_k)
H = GenerateTopicTermMatrix(A, best_k)
```

The topic labeling method uses the same extracted base for the construction of item-layer topic models to be implemented. Algorithm 2 demonstrates the steps followed by the recommendation system.

Algorithm 2*:* MOOCs Ecosystems Automatic topic labeling

Input: Quantity of generated topics *k*, document-topic matrix *W*, description (*snippets*) of each document, the generated *model* of topics, the vectorized terms *vec*, and the *approach* selected
Output: *top-1 label and top-3 label*

```
topTerms = getTop10T(model, 10, k, vec)
for topic_i = 1 to k do
    top_D = getTop10D(snippets, W, topic_i, 30)
    top_T = topTerms[topic_i]
    for d = 1 to top_D do
        dt_label = dt_label + getPrimitiveLabels(d, approach)
    if approach is TS do
        for primitive = 1 to dt_label do
            if primitive in top_T do
                list = list + primitive
    else if approach is KS do
        list = dt_label
    candidates = applyTFtoRank(list)
    top-1 = getTopLabel(candidates, 1)
    top-3 = getTopLabel(candidates, 3)
```

Another step of the recommendation process involves the implementation of our content-based method by cross-referencing the item topics (Figure 1, see 11) and the user's documents (Figure 1, see 6) through NMF in the documents-terms matrix. Considering that the user profile represents what data the student has already enrolled, it is possible to identify the "user topics of interest" (Figure 1, see 13). This identification is based on the topics of the item layer mostly related to the user.

The user documents are concatenated into a single search string. This string represents all the knowledge that the user has already obtained or is enrolled. To recommend other documents related to the user profile, we apply the Euclidean distance between the search string and the other documents in the item model. As a result, we have the item document identifier and its distance between this document and the user string. To identify the closest documents to the user profile, we sort the results ordering by the smaller distances.

We merge some information to represent documents: module title, module URL, provider identifier, linked exercise or video URLs, and course URL. A set of documents *s* related to the topic is extracted. However, it is necessary to identify in this set what is the students' knowledge gap, i.e., which modules linked to this topic the student has not had contact yet (verify 'novelty'). The system verifies what documents of the user layer are in the set of extracted documents *s*. If the user layer contains modules or courses, the comparison is made by analyzing if the documents are equal. If the user layer contains videos, the video identifiers present in each module of *s* are extracted. Next, if a user's watched video is in *s*, this document is discarded, i.e., the student has already enrolled in this module and it does not appear as a knowledge gap. In this case, the documents that are out of the watched videos are selected for recommendation. We consider the top-6 of the list and show them back to the user.

6. Evaluation and Results

This section presents a set of three experiments to evaluate our proposal and verify if the expected goal has been achieved. First, an experiment with real data collected from multiple MOOCs providers is performed to verify the effectiveness of the topic modeling technique in our recommendation system if compared to LDA (another technique widely used in modeling of topics [Nolasco, 2016]). The second experiment focused on evaluating the topic labeling method, mainly the representativeness of labeling technique. Labels for topics are generated from our approach and then compared to labels from MOOCs providers. Finally, we conduct a quasi-experiment to collect feedback from users through a web system. Results were collected and analyzed using recommendation system metrics in order to evaluate the recommendation stage.

6.1. First Controlled Experiment: topic modeling effectiveness

The protocol for the evaluation of topic modeling in our method involves the entire provider's dataset. This integrated data (a total of 106,574 modules) extracted from MongoDB were used to fill the item layer and the user layer with modules enrolled by a target user. Our method proposes that the automatic definition of the number of topics in NMF can be applied also to the domain of the MOOCs. The data input occurs through the API topic tree. However, only some specific fields are extracted.

The results of our model are compared to the LDA baseline approach through topic coherence, which verifies how semantically the terms of a topic are related. For effective comparison with LDA, we use the same database that is used with NMF. The stop words and the only noun criteria are also the same and a Lemmatization process is also applied. We apply the TC-W2V metric [O'Callaghan et al., 2015] to verify coherence. The top 10 terms of each topic are grouped and inserted in the coherence method. The result is 0.32, i.e., less than the coherence of our method. Figure 2 represents the values of *k* and the coherence of the respective topic considering each *k* in both situations: with the modified NMF and with baseline LDA.

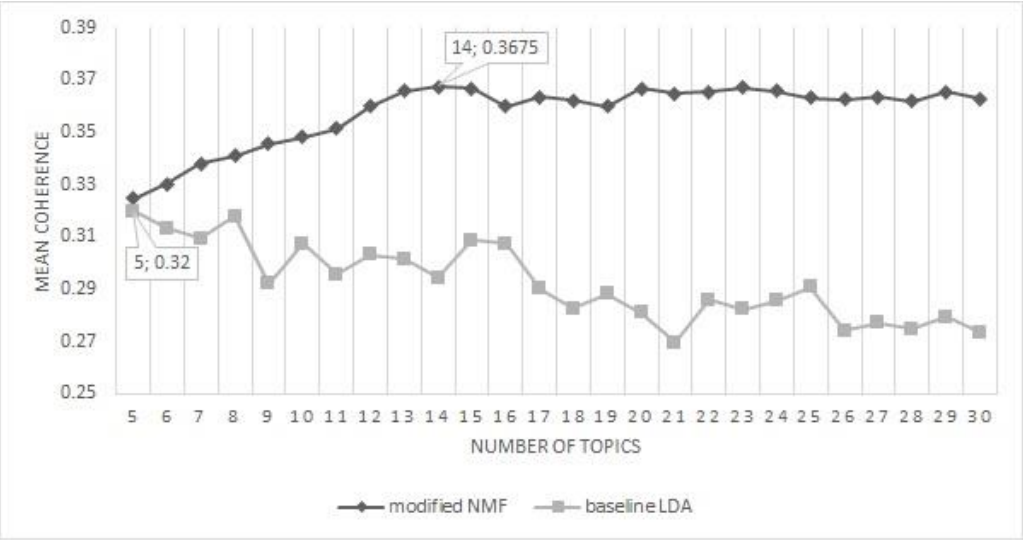

Figure 2. Topic coherence in each k-value for the item layer using modified NMF and LDA. Source: [Campos, 2019]

It can be observed that both techniques have a close coherence value when *k*=5: LDA with 0.32 (best value obtained in the model) and our modified NMF with 0.3246 (lowest value obtained in the model). Other approximations are observed, but always with higher coherence in the NMF model. Table 2 provides an overview of topic coherence values including the modified NMF model generated and the baseline LDA approach.

Table 2. Topic coherence comparison. Source: [Campos, 2019]

	Topic Coherence (best k)	**Mean (SD)**	**Median**	**Range**
1. Modified NMF	0.3675 (14)	0.3574 (0.011)	0.3628	0.3246 – 0.3675
2. Baseline LDA	0.32 (5)	0.2925 (0.014)	0.2905	0.2696 – 0.32

For the topic coherence measure evaluated in this study, the topic modeling with NMF produces better topic coherence in all verified possibilities. Table 2 suggests an improvement in NMF if compared to the baseline LDA. The application of the entire database of the selected MOOC providers allows consistency to be verified across the most diverse data scenarios. Therefore, it can be concluded that NMF is relevant to be applied in our recommendation process as it better represents the providers' multiple documents selected for this process.

6.2. Second Controlled Experiment: representativeness of the labeling technique

A second controlled experiment was conducted to verify the defined topic labeling method in comparison with the labels of providers' labels. The evaluation methodology is designed to select the dataset and use our approach to label generation (Text Selection or TS, its variations, and Keywords Selection or KS). Then, we select these labels and compare them with provider labels. This comparison is made by using the cosine distance of strings (labels). For this experiment, the automatic labeling technique that obtains the closest proximity to the provider's manual labels is the most appropriate and contains better results.

Labels for our system are first generated using Algorithm 2. It has two outputs: one consisting of top-1 labels for each topic and one consisting of top-3 labels for each topic. Considering that the module structures do not have direct keywords, we adopted the TS approach with text extraction following the fast keyword extract algorithm. However, as the courses in which these modules are allocated have keywords (knowledge areas of each course), it is interesting to check the KS approach by selecting keywords as candidate labels.

As a first step, it is necessary to select the top-30 documents associated with each topic. Each document is a module of different courses from multiple providers, so a course can have different modules with the same name. Some text treatments are applied to the displayed names (e.g. removal of special characters). Next, it is necessary to select the top-10 terms for each topic. For the selection of primitive labels in TS, the fast extraction algorithm provides a wide text of all 30 documents. The excerpt from the document goes through some removals, as well as those that are performed in topic modeling. Therefore, it considers the exclusion of non-nouns.

After selecting primitive labels, it is necessary to choose the candidate labels. In the case of TS, we consider all primitive labels that are in the top-10 term set of the topic. Next, Term Frequency (TF) ranks each term according to the word frequency of the candidates in the primitive label string. Finally, in the selection of top-3, we verified the case of one label being a substring of another (for substitution).

In the case of KS, we consider the same top-30 documents and the top-10 terms associated with each topic. The novelty of the KS approach starts in the selection of primitive labels. For each of the 30 documents, we select the area in which the course is inserted. While in TS there is a primitive check with the top-10 terms for selecting candidate labels, all primitives are candidates in KS. Then, we apply the TF to order the candidates and, finally, it is possible to select the terms.

From the collected results, one way to compare the approaches is to analyze the distance of the automatic labels with those already existing ones in the providers. Table 3 shows the cosine distance between the TS (top-1), TS (top-3), and KS strings relative to the provider strings.

The results point to a closer proximity of TS (top-3) to provider labels. Only topics 1 and 13 have KS as their closest approach. It can be stated that the topic labeling technique, specifically TS (top-3), can automatically calculate labels for the topics. This automatic identification is useful not only for describing item layer topics, but also for identifying the "user topics of interest" of the recommendation process.

6.3. Third Controlled Experiment: user perspective

The experiment presented in this section aimed to evaluate the quality and performance of our system's recommendations from explicit feedbacks. This feedback is collected from each participant (represented as a target user in the recommendation system) through a web system.

Table 3. Distance between strings in each approach. Source: [Campos, 2019]

Topic	TS (top-1) Cosine	TS (top-3) Cosine	KS Cosine	Best Approach
0	0.0000	0.5221	0.7957	KS
1	0.0000	0.7512	0.0578	TS (top-3)
2	0.0917	0.8214	0.2380	TS (top-3)
3	0.0000	0.8436	0.0430	TS (top-3)
4	0.0000	0.6642	0.0626	TS (top-3)
5	0.0000	0.6679	0.6180	TS (top-3)
6	0.0000	0.7605	0.3220	TS (top-3)
7	0.2917	0.8674	0.3713	TS (top-3)
8	0.1856	0.8572	0.1856	TS (top-3)
9	0.0000	0.6487	0.6356	TS (top-3)
10	0.0000	0.8211	0.2567	TS (top-3)
11	0.5071	0.8052	0.5738	TS (top-3)
12	0.0000	0.6221	0.6944	KS
13	0.0000	0.7754	0.6203	TS (top-3)

Regarding the evaluation of the recommended modules, we selected a set of metrics: Main Average Precision (MAP), Utility, Novel, and Confidence. These set are part of the metrics to evaluate recommendation systems based on their properties specified by Ricci et al. (2015). To better define the experiment, we used the Goal, Question, Metric (GQM) paradigm [Basili, 1992], as structured in Table 4.

Table 4: Goal planned for the third experiment. Source: [Campos, 2019]

Analyze	the recommendation method created
With the purpose of	evaluating the quality of module recommendations
With respect to	user perception and satisfaction and the properties MAP, Utility, Novelty, and Confidence
From the point of view of	MOOCs learners
In the context of	the modules offered by the selected platforms: Udemy, edX, and Khan Academy

Based on the objective, it was possible to define the hypotheses of the experiment:

a) **Null hypothesis (H0):** The proposed recommendation method achieved efficacy of less than 50% in the properties of MAP, Utility, Novel, or Confidence.

H0: (μMAP_OurApproach < 50%) OR (μUtility_OurApproach < 50%) OR (μNovelty_OurApproach < 50%) OR (μConfidence_OurApproach < 50%), where:

μMAP_OurApproach = MAP of the feedback collected for our recommendation system

μUtility_OurApproach = Utility of our recommendation system collected through user feedback

μNovelty_OurApproach = Novelty of our recommendation system collected through user feedback

μConfidence_OurApproach = Confidence of our recommendation system collected through user feedback

b) Alternative hypothesis (H1): The proposed recommendation method achieved efficacy greater than 50% in the properties of MAP, Utility, Novel, and Confidence.

H1: (μMAP_OurApproach ≥ 50%) AND (μUtility_OurApproach ≥ 50%) AND (μNovel_OurApproach ≥ 50%) AND (μConfidence_OurApproach ≥ 50%).

We developed a web system to follow the planned methodology. The system implemented the following steps:

a) **Authentication:** Each participant receives a login and password in advance. This access information is important because it is linked to the specific information that each user evaluates during the other steps;

b) **Informed Consent:** The information of the informed consent, the study, the stakeholders and the data privacy are fully disclosed. The participant must accept this information before advancing the study. The full informed consent is presented in the Master's thesis [Campos, 2019];

c) **Participant characterization questionnaire:** Each participant is asked to complete a questionnaire to collect personal information, usage profile in MOOCs, and other learning platforms;

d) **Evaluation of recommended modules:** In this step, the participant views 6 modules recommended by our system. For each module, the participant must answer some specific questions. At this point, feedback from the recommendation item is collected.

The experiment was conducted with participants with different personal profiles and motivations. We invited participants who are/were active in MOOCs platforms, or who had already enrolled in online courses in another learning environment. The first contact was done with the participants to collect recommendation inputs. The recommendation was made offline with the authorization of participants to access provider data. The result for each participant corresponded to the top-6 recommended modules or courses. For each module evaluated by each participant in the web system, a set of questions was defined, as well as each question corresponding to one or more of the metrics, as described next.

To calculate MAP, participants had to answer in the web system the following question: "*Would you find it relevant to learn this content?"*. The question was presented

in all the 6 modules, enabling the participant to respond Yes (to relevant content) or No (to no relevant content). After collecting the answers and applying the MAP formula, results indicated MAP = 62.24%. To calculate utility, participants were asked *"How useful would this content be for you?"*. Results indicated Utility = 68.89%. In the case of novelty, participants were asked *"Have you learned this course before*?". Answers are restricted to "Yes" (when the participant has previously studied the considered content) or "No" (if he/she has never studied the content before). Results showed Novelty = 99.12%. The confidence of the recommendation system is represented by how reliable it can be in its recommendations. In this case, the calculation is given by the ratio between all positive evaluations (those that obtained "Yes" as an answer to the question *"Would you find it relevant to learn this content?"*) and the total number of valid evaluated modules. In this evaluation scenario, the system obtained a confidence of 72.81%.

After applying the metrics based on the answers collected in the web system, it was possible to check that the proposed recommendation was more than 50% effective in all the verified properties. Thus, the alternative hypothesis was accepted. Based on the results and returning to RQ1, we can state that our proposal for MOOC ecosystems is an accurate, useful, and reliable tool that presents new content to fill the users' knowledge gap within this ecosystem. Moreover, the identification of the knowledge gap is possible in our proposal of applying the topics modeling and labeling, whose results were coherent, allowing extractions such as the user's topics of interest.

The results of our evaluation helped us to answer the main research question RQ1: *How to identify and reduce knowledge gaps in the MOOCs ecosystems?* It demonstrates that: a) one of the possibilities of the SECO approach is the integration and extension of data from several MOOCs providers used as a dataset in the evaluation and as an input in the proposed recommendation system; b) the detection of similarity between these multiple MOOCs ecosystems data is possible by applying the topic modeling techniques using the proposed modified NMF approach; c) the combination of topic labeling techniques adopted by this work can be applied in the MOOCs ecosystems scenario, obtaining similar results to the providers' ones, but automatically; and d) the proposed content-based recommendation system helps in the acquisition of new and relevant knowledge, improving student's experiences and reducing their knowledge gaps, by using student's interactions in MOOCs ecosystems and implemented structures/components.

7. Conclusion

With the increasing number of courses available in MOOCs ecosystems, it might be difficult for students to choose the best ones among all providers. The main contribution of this work is a new recommendation system applied to the scenario of MOOCs ecosystems. It is possible to highlight contributions such as reducing the knowledge gap and the content-based recommendation method itself. Moreover, the recommendation of part of courses from multiple providers contributes to the customization level in terms of the recommendation. Finally, we include the following contributions: the modeling of MOOCs ecosystems, the method for extracting data from multiple providers that can also be reused in other applications, and the secondary study (SMS) that was carried out.

The problem investigated by this work is aligned with the challenges "4.3.1. Information Ecosystem Development" and "4.3.2. Open and Collaborative Processes in Information Ecosystems" from the Grand Research Challenges in Information Systems

in Brazil [Araujo, 2016]. It can be observed that this work builds an information system for the open world of MOOCs, covering some aspects such as scalability, flexibility, and adaptation (Challenge 4.3.1). Scalability refers to the possibility of adding data from other providers, once we implemented techniques as a method that defines the ideal number of topics in topic modeling – and it is not necessary to change our conceptual model. Moreover, this integrated document base still goes through the topic modeling process, which considerably decreases the prediction time. It is a factor raised by Aggarwal (2016) as being crucial in recommendation systems since it represents the time that takes to the user receives the answer.

This work also addresses the aspect of flexibility in addressing MOOCs in the MOOCs ecosystems approach. It allows, for example, to extend recommendations to other ecosystems' actors since the recommendation interacts with the ecosystem rather than with a specific provider. Implementation features are also flexible, as no techniques related to a specific programming language or software have been adopted. As future work, this work will be made openly available so that the adaptation aspect can be contemplated, thus allowing developers to contribute with the recommendation system.

The recommendation system also includes ecosystem characteristics, e.g. provider data privacy, user data in the ecosystems, or similarity between courses from multiple providers. These considerations contribute to the challenge "new information systems in the open world". Based on our contribution, the learning process of providers (information systems) is openly and collaboratively integrated (Challenge 4.3.2). From a SECO approach, providers and users become contributors to the learning processes. The recommendation system proposed in this work has helped to support these interactions so that processes (such as finding suitable content for a student) are integrated.

The Master's thesis was reported in 4 papers and 2 book chapters directly and/or indirectly related to MOOCs ecosystems. We introduce our proposal, methodology, and objectives in a paper published at the **XIV Brazilian Symposium on Information Systems (SBSI 2018)** (Qualis B2) [Campos et al., 2018a]. Next, we publish a full paper about the modeling of MOOC Ecosystems and the conceptual model of our recommendation system in the **IEEE 19th International Conference on Information Reuse and Integration for Data Science** (Qualis B1) [Campos et al., 2018c]. The implementation details involving the inclusion of the topic modeling in our proposal was accepted as a chapter to compose the international book called **Reuse in Intelligent Systems** [Campos et al., 2019]. The first stage of our evaluation method regarding the effectiveness of the topic modeling method was accepted as a full paper at the **XVI Brazilian Symposium on Information Systems (SBSI 2020)** (Qualis B2) [Campos et al., 2020]. We can also mention two other works that contribute to the motivation and background of our proposal, respectively: a paper accepted to the **Workshop on Big Social Data and Urban Computing** (a 44th International Conference on Very Large Databases workshop) [Campos et al., 2018b] and a short course (and book chapter) at the **VI Rio de Janeiro Regional School of Information Systems** [Marinho et al., 2019].

A first future work would be to include the development of a full web interface to the recommendation system, where the user could interact directly with the system, triggering the recommendation, authorizing access to private data from the providers where he/she has an account. Finally, the system would display the recommended modules with the appropriate links to the pages in the providers, as it was implemented

in the web system. These steps so far are performed directly on the Python interpreter. One option would be to extend the already developed web system, making this system open and not requiring feedback or other personal data, as it was required for our evaluation.

Another future work is a broader identification of the knowledge gap, creating parameters to identify the degree of learning and deepening of a given topic by a user. This is necessary because users often complete a module, but it does not necessarily indicate that knowledge has been acquired. Thus, the knowledge gap would identify a student's degree of understanding concerning a specific topic.

Acknowledgment

We thank IFRJ for supporting this research. We also thank CAPES, CNPq, and FAPERJ (Brazil) for their financial support to the research group.

References

Aggarwal CC. 2016. Recommender systems. Cham: Springer International Publishing.

Apaza RG, Cervantes EV, Quispe LC. 2014. Online Courses Recommendation based on LDA. In: SIMBig., p 42–48.

Araujo R. 2016. Information Systems and the Open World Challenges. Boscarioli, C, R.M Araujo e R.S.P Maciel. I Gd. - Gd. Res. Challenges Inf. Syst. Brazil 2016-2026. Spec. Comm. Inf. Syst.: 42--51.

Barbosa OALP, Santos RP, Alves CF, Werner CML, Jansen S. 2013. A Systematic Mapping Study on Software Ecosystems from a Three-Dimensional Perspective. In: Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry., 1ed. Northampton/USA: Edward Elgar Publishing, p 59–81.

Basili VR. 1992. Software modeling and measurement: the Goal/Question/Metric paradigm.

Bhatt C, Cooper M, Zhao J. 2018. SeqSense: Video Recommendation Using Topic Sequence Mining. In: Springer, editor. Proceedings of the International Conference on Multimedia Modeling. Cham, Switzerland, p 252–263.

Campos R. 2019. Recommendation System for Knowledge Acquisition in MOOCs Ecosystems. Master's Thesis in Informatics. PPGI/UFRJ. Rio de Janeiro, Brazil.

Campos R, Santos RP, Oliveira J. 2020. A Recommendation System based on Knowledge Gap Identification in MOOCs Ecosystems. In: XVI Simpósio Brasileiro de Sistemas de Informação., p 8.

Campos R, Santos RP, Oliveira J. 2019. A Recommendation System Enhanced by Topic Modeling for Knowledge Reuse in MOOCs Ecosystems. In: Reuse in Intelligent Systems. Boca Raton, Florida, EUA: CRC Press (in press).

Campos R, Santos RP, Oliveira J. 2018a. Recommendation Systems for Knowledge Reuse Management in MOOCs Ecosystems. In: Anais do XIV Simpósio Brasileiro de Sistemas de Informação (SBSI), editor. XI WTDSI - XI Workshop de Teses e Dissertações em Sistemas de Informação. Caxias do Sul/RS, Brasil: Porto Alegre: SBC, p 46–48.

Campos R, Santos RP, Oliveira J. 2018b. Using Multilayer Social Networks in an Analysis of Higher Education for Professional Demand. In: I Workshop on Big Social Data and Urban Computing (BIDU), 2018, Rio de Janeiro. Proceedings of the Poster Track of the BiDU 2018. 44th International Conference on Very Large Data Bases (VLDB). Aachen: CEUR-WS, 2018, p 1–15.

Campos R, Santos RP, Oliveira J. 2018c. Web-Based Recommendation System Architecture for Knowledge Reuse in MOOCs Ecosystems. In: 2018 IEEE International Conference on Information Reuse and Integration (IRI). IEEE, p 193–200.

Corbellini A, Mateos C, Zunino A, Godoy D, Schiaffino S. 2017. Persisting big-data: The NoSQL landscape. Inf. Syst. 63: 1–23.

Greene D, O'Callaghan D, Cunningham P. 2014. How Many Topics? Stability Analysis for Topic Models. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Berlin: Springer, p 498--513.

Jing X, Tang J. 2017. Guess You Like: Course Recommendation in MOOCs. In: Proceedings of the International Conference on Web Intelligence. ACM, p 783–789.

Li C, Song Z, Tang J. 2018. User Tagging in MOOCs Through Network Embedding. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). IEEE, p 235–241.

Marinho LH, Campos R, Santos RP, Ferreira da Silva M, Oliveira J. 2019. Conceitos, Implementação e Dados Privados de Algoritmos de Recomendação. In: França TC de, Thomaselli Nogueira JL, Antunes JF, editors. Minicursos da ERSI-RJ 2019 - VI Escola Regional de Sistemas de Informação do Rio de Janeiro (ERSI). SBC, p 32.

Nolasco D. 2016. Identificação Automática de Áreas de Pesquisa em C&T. Master's Thesis in Informatics. PPGI/UFRJ. Rio de Janeiro, Brazil. 195 p.

O'Callaghan D, Greene D, Carthy J, Cunningham P. 2015. An analysis of the coherence of descriptors in topic modeling. Expert Syst. Appl. 42: 5645–5657.

Ricci F, Rokach L, Shapira B, Kantor PB. 2015. Recommender systems handbook. Springer.

Shah D. 2019. Year of MOOC-based Degrees: A Review of MOOC Stats and Trends in 2018.

Shanyun K, Qin S, Guolin Z. 2015. Research on the Construction of MOOC Learning Community Ecosystem Circle. In: 2015 International Conference of Educational Innovation through Technology (EITT). IEEE, p 199–203.

Song J, Zhang Y, Duan K, Hossain MS. 2017. TOLA: Topic-oriented learning assistance based on cyber-physical system and big data. Futur. Gener. Comput. Syst. 75: 200–205.

Wang J, Xiang J, Uchino K. 2015. Topic-Specific Recommendation for Open Education Resources. In: International Conference on Web-Based Learning., p 71–81.