Expert Systems With Applications 205 (2022) 117510

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Providing recommendations for communities of learners in MOOCs ecosystems

Rodrigo Campos [a,c,*], Rodrigo Pereira dos Santos [b], Jonice Oliveira [a]

[a] Institute of Computer Science, Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro 21941-916, Brazil
[b] Department of Applied Informatics, Federal University of the State of Rio de Janeiro (UNIRIO), Rio de Janeiro 22290-255, Brazil
[c] Systems Development Coordination, Federal Institute of Education, Science and Technology of Rio de Janeiro (IFRJ), Rio de Janeiro 20061-002, Brazil

ARTICLE INFO

Keywords:
Online education systems
Content-based recommendation
Topic modeling
Non-negative matrix factorization
Unsupervised machine learning

ABSTRACT

Massive Open Online Courses (MOOCs) have been widely disseminated due to the arrival of Web 2.0. However, the growth of MOOCs brings some difficulties for students in choosing suitable courses in these ecosystems. In recent years, some recommendation systems emerged to solve this problem but remain limited since they do not identify the student's prior knowledge broadly or the student's goals. To overcome this limitation, this work proposes the Fragmented Recommendation for MOOCs Ecosystems (FReME), a recommendation system to suggest parts of courses from multiple providers (i.e., Khan Academy, Udemy, and edX). FReME is based on the student profile and on the MOOCs ecosystems perspective to balance the ecological environment and strengthen interactions. Moreover, we differ from the current recommendation systems since our method identifies and reduces the students' knowledge gap optimizing the learning process. Experimental results conducted with a dataset integrating 3 MOOCs providers and 19 students demonstrated that the implemented techniques are more consistent than other approaches. Finally, it was verified through precision, utility, novelty, and confidence that our recommendations are 62,24% accurate, 68.89% useful, 72.81% reliable, and present new content in 99.12% of cases. These results validate that FReME supports students in reducing their knowledge gap.

1. Introduction

With the development of the Internet, open online education has been widely used for bringing new values based on ethics of participation, openness, and collaboration, which makes it a facilitator for the issues faced by traditional learning methods (Peters, 2008). One of the modalities of this movement is the Massive Open Online Courses (MOOCs) that are courses open to any student, anywhere, at almost any time, and with a very large number of students enrolled. MOOCs are run by MOOCs platforms, which can be proprietary (e.g., Udacity) or open-source (e.g., Open edX) (Kim, 2014).

In the context of e-learning in general, some authors point to the benefits of exploring distance education platforms based on the software ecosystem perspective, which is defined as a set of businesses operating as a unit that interacts with a shared market for software and services, with relationships among them underpinned by a common market or technological platform (Jansen et al., 2009). It becomes appropriate due to the demand for inter-organizational reuse (Bosch, 2009), either from learning objects repositories or even from educational services, such as forums and chats. Therefore, some work points out characteristics, roles, or definitions of software ecosystems to be applied specifically in the context of MOOCs, i.e., to analyze MOOCs in a broader perspective so-called MOOCs ecosystems (Campos et al., 2018).

Due to the COVID-19 pandemic, the number of courses and users in MOOCs has been growing worldwide. In the year of 2020, the top-3 MOOCs providers (i.e., Coursera, edX, and FutureLearn) had 2800 new courses with more registered users in April 2020 than in the whole year of 2019 (Shah, 2020a, 2020b). The pandemic caused a spike in unemployment and accelerated the demand for professionals trained in the jobs of the future, composed of new technologies, sectors, and markets in more interconnected global economic systems (World Economic Forum, 2020). There was an increase in demand for online learning, with employed people seeking out more personal development courses and unemployed people with a larger emphasis on digital learning skills (World Economic Forum, 2020). In this scenario, besides offering these most sought courses, MOOCs play a relevant role in democratizing learning without geographic boundaries and with several free courses provided by universities around the world (Shah, 2020b).

* Corresponding author at: Institute of Computer Science, Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro 21941-916, Brazil.
E-mail addresses: rodrigo.campos@ufrj.br (R. Campos), rps@uniriotec.br (R.P.d. Santos), jonice@dcc.ufrj.br (J. Oliveira).

https://doi.org/10.1016/j.eswa.2022.117510
Received 19 August 2021; Received in revised form 2 May 2022; Accepted 3 May 2022
Available online 20 May 2022
0957-4174/

However, with the growth of MOOCs platforms, problems arose for students, as the difficulty in finding the most suitable course that matches their expectations or needs. There are countless possibilities of courses distributed at different providers which can be a positive aspect. However, in many cases, it causes information overload with students without the proper guidance on choosing the most appropriate content. Previous research has suggested that more than half of dropouts in MOOCs may occur when the enrolled course was not the one that students were seeking or when their expectations were not attended by the course (Zhang et al., 2021).

Therefore, in this work, we propose a solution to this problem based on an investigation of software ecosystem strategies in such a scenario considering the following main research question (RQ): *How to identify and reduce knowledge gaps in the MOOCs ecosystems?*

Considering that there is a gap between the knowledge already obtained by the student and the new knowledge required to reach the desired expertise, this work aims at supporting in reducing specifically such users' knowledge gaps, i.e., the knowledge that the student does not yet have about a particular subject of interest. To do so, we combine recommendation systems with users' data from MOOCs providers and propose the Fragmented Recommendation for MOOCs Ecosystems (FReME) to investigate how to assist them in achieving their goals based on the recommendation of parts of courses (i.e., courses' modules),[1] considering the perspective of MOOCs ecosystems. These recommendations apply for full courses if providers do not organize content into modules, for example.

Several recommendation systems have been proposed to support MOOCs adopting different approaches, such as Collaborative Filtering (CF) or content-based recommendation. However, these efforts may fail to identify the most adequate knowledge broadly, since they focus on specific MOOCs providers, or recommend complete courses, or videos from MOOCs platforms. Thus, it is necessary to propose solutions that consider student's profiles using multiple platforms and fragmented knowledge aiming at the reduction of their knowledge gap. The learning item fragmentation is a target of future work in recent studies applied to recommendation in MOOCs, remaining a challenge in this field. Jing and Tang (2017) ratify the need to improve recommendations in MOOCs not only recommending full courses, but also "different kinds of content such as video and knowledge" since it can improve the personalization level of recommendations in MOOCs. Barros et al. (2018) emphasize plans to include "the recommendation of a set of topics of a course for a student, not just a full course, in order to provide more accurate recommendations".

This article is organized, as follows: Section 2 describes the main concepts of our work. Section 3 reviews the related work. Section 4 introduces the FReME conceptual model, details the content-based and the topic labeling methods, and presents a example of recommendation. Section 5 presents our experimental results, and this article is concluded in Section 6.

2. Background

In this section, we describe the main concepts addressed in this paper: recommendation system and MOOCs ecosystems.

2.1. Recommendation system

A recommendation system aims to provide useful and effective suggestions of items for a user typically based on the user's preferences and constraints. This type of system has been used in a variety of scenarios, such as to recommend music, products, movies, or even educational resources. The latter emerged with a need to support students in choosing a better learning path, given a large number of possibilities (Ricci et al., 2015).

Recommendation systems can apply several approaches, including CF, content-based, and hybrid that combine different approaches. CF considers multiple users' preferences parsed from their profile, whereas the content-based considers similar patterns related to specific user's experiences. Two main entities are defined in most of the recommendation systems: (a) user – the entity to which the recommendation is provided; and (b) item – the product that is effectively recommended (Aggarwal, 2016).

CF considers similarities of items' evaluation from users to recommend new items. Therefore, when a user u evaluates items similar to another user u', the chances of the next evaluation of u to a new item to be similar to the evaluations of u' increase. There are some techniques widely used in CF, such as the Matrix Factorization (MF) (Do et al., 2010), which decomposes an array by applying linear algebra, creating a matrix of historical data that aims to model item features and user preferences (Li et al., 2021). However, the sparsity of this rating matrix can cause some problems, including on the recommendation performance. Several types of extra information can be added to alleviate this problem and construct an accurate latent factor of the user or item, such as reviews (Li et al., 2021).

In some scenarios, the ratings of other users (such as on the example with u') are not known. An alternative might be the content-based approach, which is based on the items' descriptive attributes (the so-called content). The content-based approach uses the descriptor of an item already evaluated by the user to recommend another item i with similar descriptors based on the similarity level. The content-based can also be advantageous for scenarios with new items since they have little or no classification. However, a possible problem is the limitation of recommended items if that user has never classified other items with the same set of words as i, thus i is rarely recommended, the so-called cold-start. A hybrid approach, or even other less-used approaches such as the knowledge-based, can be applied to solve the cold-start problem (Aggarwal, 2016).

Several secondary techniques can support recommendation systems approaches in many aspects, such as clustering techniques, machine learning techniques, or classification algorithms. For instance, the content-based approach can be supported by a convolutional neural network (CNN), i.e., a deep learning algorithm, to make personalized learning resources or news recommendations (Shu et al., 2018). Deep learning can also be applied to support CF approaches (Shen et al., 2019; Yi et al., 2019). When applied in recommendation systems, deep learning can be grouped into: (a) the deep learning prior estimate model (DLPE), which applies methods such as CNN; and (b) the single channel recommendation model (SCR) that learns key patterns from user's historical behavior to predict new recommendations (Yi et al., 2019).

2.2. MOOCs ecosystems

MOOCs can be divided into two categories: xMOOCs and cMOOCs. In the former, there is an instructional proposal and a teacher still plays the central role in the dissemination of knowledge (Bond & Leibowitz, 2013). Teachers also create and record classes' content, define exercises, and guides the students in defining the paths to be followed. Despite the debate between students is encouraged, it is directed by a tutor (Biagiotti, 2016). The latter is based on the theory of connectivism. The contents can be curated and sent to students through a daily

[1] In MOOCs, each course program has a set of courses, and each course has a set of modules, varying according to each provider. When recommending parts of the MOOCs courses, we consider "part of the course" as the smallest grouping of a course structure in a MOOC provider. It did not consider videos or books, for example, since a video does not group elements. A video in a MOOC can be interpreted as a learning object (Gope & Jain, 2017).

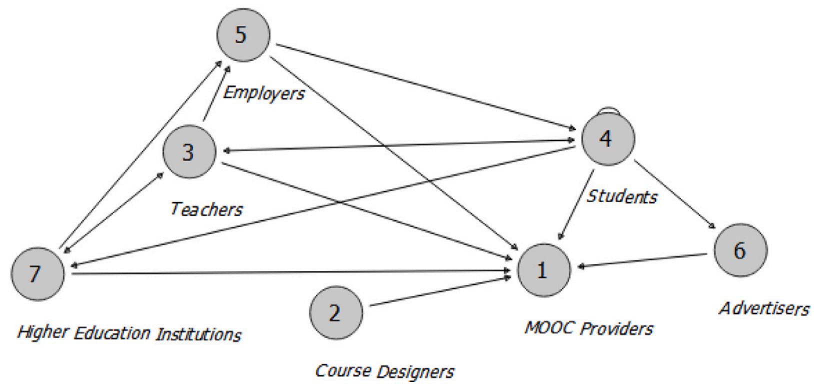

Fig. 1. Interactions in the network of actors within the MOOCs ecosystems.
Source: Adapted from Campos et al. (2018, 2020d).

newsletter (aggregation concept). The connection between students and such contents can be made and documented through resources, i.e., blogging or social bookmarking (remixing concept). Participants can create their internal connections with such contents (repurposing concept), besides being able to share their new connections with other participants (feeding forward concept) (Yeager et al., 2013). In cMOOCs students are co-authors of the course content and the teacher is on the same hierarchical level as the students (Biagiotti, 2016). Besides the difference in the construction of the content, the "massive" aspect is also presented differently. In xMOOC, there is scalability in the offer of courses to students, whereas in cMOOC the courses are geared towards specific online communities and can be easily reused and remixed to fit different interests (Biagiotti, 2016; Yeager et al., 2013).

In this context, MOOCs ecosystems can be analyzed from the interactions of such actors that, besides teachers and students, also include higher education institutions, among others, so that it is possible to share knowledge between different entities (Campos et al., 2018). Fig. 1 shows a graph in which the nodes represent these actors and the edges represent these interactions. MOOCs ecosystems inherit features from the software ecosystem perspective that enable software products' suppliers and customers to collaboratively create technologies and components to add value for a platform or products (Jansen & Cusumano, 2012). Other characteristics are the cooperation between independent entities factor and the fact that each actor assumes a different role in this ecosystem. Higher education institutions, for example, plays the role of keystone, i.e., it is an actor that adds value to MOOCs ecosystems and that is primarily responsible for maintaining their longevity and growth (Campos et al., 2018).

The FReME specification is focused on MOOCs ecosystems which facilitates the inclusion of multiple MOOCs providers in the conceptual model of FReME, enabling that other developers extend it and contribute to the implementation of other providers. To do so, the only requirement is that the provider allows the extraction of course data by developers or researchers in JavaScript Object Notation (JSON) standard. Next, such data can be treated and added to the FReME database (see Section 4) so that FReME starts to consider it.

3. Related work

In our previous work, we conducted a Systematic Mapping Study (SMS) (Campos, 2019; Campos et al., 2021) to analyze studies that present recommendation systems applied to MOOCs. The most used approach in this context is CF, such as in the study of Symeonidis and Malakoudis (2019). However, this approach is only feasible in scenarios where it is possible to extract data from several users, which is not the case of the present work since it aggregates several providers including those that only allow data extraction from the user interested in receiving the recommendation (i.e., target user). Therefore, we apply the content-based recommendation. To support this recommendation, we apply topic modeling that creates a distribution of probabilities avoiding the problem of grouping many dimensions and associating documents with groups to discover the thematic structure (Greene et al., 2014).

In this section, we briefly review the studies that are more related to our proposal among the selected studies in the SMS, i.e., those that propose fragmented recommendations of courses (Zhao et al., 2016) or that also apply topic modeling in content-based recommendation (Apaza et al., 2014; Aryal et al., 2019; Bhatt et al., 2018). The comparisons between the previously mentioned studies and our work were limited, given the lack of access to the datasets or source code used by these studies. Despite it, we present in Table 1 a comparison between the recommendation characteristics presented by these studies and FReME.

Zhao et al. (2016) propose an algorithm to recommend the sequence of micro-learning units aggregated in a learning path for students using an ant colony algorithm. This fragmentation allows for flexible learning based on the transitions of the learning goal, knowledge level, or area.

Bhatt et al. (2018) propose a recommendation system called SeqSense to recommend videos from multiple providers, considering as input the transcript of the MOOC platform videos and as the approach a sequence-based recommendation that finds a sequential relationship among these videos. The topic modeling is used through the most commonly used model Latent Dirichlet Allocation (LDA), choosing the fixed number of topics $Z = 30$.

Differently, the course recommendation proposed by Apaza et al. (2014) considers user data and trains the model through grading information in college (as the "ratings"), forming the user profile. The items' documents are represented by the platform courses, applying LDA to find the distribution of topics and predicting which courses would have similar ratings to recommend. Although it does not work with multiple providers, it reports that "further domains can be included by performing feature extraction".

Aryal et al. (2019) present the MOOCRec platform which also makes course recommendations from multiple providers, with data (videos and transcripts) collected with web scrapers. To recommend, MOOCRec classifies the videos using deep learning to relate it to the student's learning style. Topic modeling using LDA extracts keywords/topics from transcripts, using them to support students in filtering courses.

Although the fragmented recommendation has already been applied to MOOCs by Zhao et al. (2016), there is still a demand for the application in scenarios with data from multiple providers. SeqSense (Bhatt et al., 2018) advances in this direction with a focus on videos but the fragmented recommendation of courses in multiple providers remains a challenge. Creating a new recommendation system with these requirements is important since it guarantees a broader data extraction and, consequently, it provides a greater chance of success in the item

Table 1
Comparison between FReME and related work.

		Apaza et al. (2014)	Zhao et al. (2016)	Bhatt et al. (2018)	Aryal et al. (2019)	FReME
Approach	Ant colony algorithm		X			
	Content-based recommendation with LDA	X		X	X	
	Content-based recommendation with NMF[a]					X
Output	Micro-learning path		X			
	Videos			X		
	Courses	X				
	Learning resources (courses)				X	
	Courses and its parts					X
Type	System	X		X		X
	Algorithm		X			
	Platform				X	
Provider	Not specified		X			
	Coursera	X				
	Multiple			X	X	X

[a]Non-negative matrix factorization.

recommendation. In this context, the system needs to integrate the student's historical data in such providers and to recognize similar or equal courses in different providers. Our work contributes not only to these aspects, but also in combining parts of courses from providers. Thus, it is possible to recommend modules at one provider using student history data from another provider, or make a recommendation combo with parts from one provider and others from another provider (in both cases, according to the content relevance to the target user). Different from recent work, our work focuses on the reduction of students' knowledge gaps, i.e., on decreasing the difference between a student's goal of learning (subject and level of expertise) and their current knowledge level. To do so, FReME does not recommend courses, but "parts of courses" instead, so it can achieve a better matching among the student's needs, the specific content to cover these needs, and the time to learn.

Comparing our proposal with other topic modeling-based approaches, we achieved better results as presented in the comparisons detailed in the following sections. The use of Non-negative Matrix Factorization (NMF) instead of LDA contributes to this success. LDA makes a distribution of groups for each term of a textual document and a distribution of groups for each document. Thus, it is possible to group the documents according to the probabilities associated with each group (Nolasco & Oliveira, 2016b). Differently, NMF is based on the MF technique, and it has as input a matrix of documents and terms, allowing the generation of two approximate matrices for the latent structure in data (Kuang et al., 2015). O'Callaghan et al. (2015) indicate that the advantage of using LDA is in providing descriptors for broad topics. However, LDA has no such advantage for niche or non-mainstream content, which are scenarios where NMF performer better since it has higher coherence and lower generality in topics in such scenarios.

Moreover, to collect the content, we created a method that automatically connects to the Application Programming Interface (API) of the selected MOOCs providers and imports the courses and their parts as documents, different than studies that use web scrapers or that do not describe how their datasets are collected.

4. The proposed system

From the MOOCs ecosystems definition, identifying the actors' roles and their relationships (Campos et al., 2018), it was possible to propose the FReME conceptual model, as shown in Fig. 2. Given the restriction of some providers regarding the extraction of private data from users, the system contains an authorization layer (3) that is activated as soon as the user (1) requests a recommendation from the system (2). Data (5) from that user (i.e., authorized data from MOOCs ecosystems) is accessed through the knowledge base (4) to extract the user's history that is transformed into a set of documents (6). These documents are input to the "user column" in the user–item matrix (7). The "item column" is composed of data from the Background Data (8) layer which has data from different MOOCs providers (9) that can be recommended to users in our system. These courses are imported into the "item column" as "item topics" after going through the topic modeling step (10). To better define item topics (11), our system contains a topic labeling step (12) that also helps in creating "user topics of interest" (13) which indicates the topics the user is more familiar or interested in Campos et al., 2020b, 2020c. More details about the topic labeling step are described in Section 4.1.

Next, the recommendation engine layer (14) uses as input the user–item matrix (7) to execute two procedures: the former is the Knowledge Gaps Identification (15) that identifies the user's knowledge gap from the desired skills. The latter is the Recommendation Algorithm (16) to calculate similarities between item's topics (11) and user's documents (6). The system ranks the results and finally shows resources (recommended modules/user's topics of interest) in the system (2) (Campos et al., 2020b, 2020c). We present more details about the recommendation engine in Section 4.2.

We implement the described conceptual model extracting data (documents) from Khan Academy, Udemy, and edX, and then store and integrate them in a single MongoDB schema. These providers were selected because of open data availability and the conceptual model is flexible in extension to other providers. The choice of a document-oriented database also allows the number of fields to be expanded or new features to be added, contributing to scalability, consistency, and availability of our system (Campos et al. (2020b), Campos et al. (2020c); Corbellini et al. 2017). For the extraction, we store the URL of each provider's API in a variable so that another variable makes an HTTP request passing this value as the parameter. Next, another variable loads the JSON content of this request. The data extraction starts in with the topic tree and traverses the children's fields, following the data organization particularities in each API.

We implemented the topic modeling with a modified NMF method (Campos et al., 2020a). We adopted NMF since, according to Aggarwal (2016), it provides a high level of interpretability in understanding the user–item interactions, despite having a better accuracy when compared with other methods. NMF interpretability is even greater in systems in which users can manifest a liking for an item, but in which there is no mechanism to specify a dislike (Aggarwal, 2016). However, different from the most used NMF, our work modified the method to contemplate the stability analysis approach (Greene et al., 2014) which automatically identifies the best number of topics for the model by calculating the best topic coherence (i.e., the relatedness between terms in a topic). In the present work, we also relate all user documents to the item model (see Section 4.2), preventing the users from receiving recommendations only from their target user topic, which could limit novelty in the recommended items.

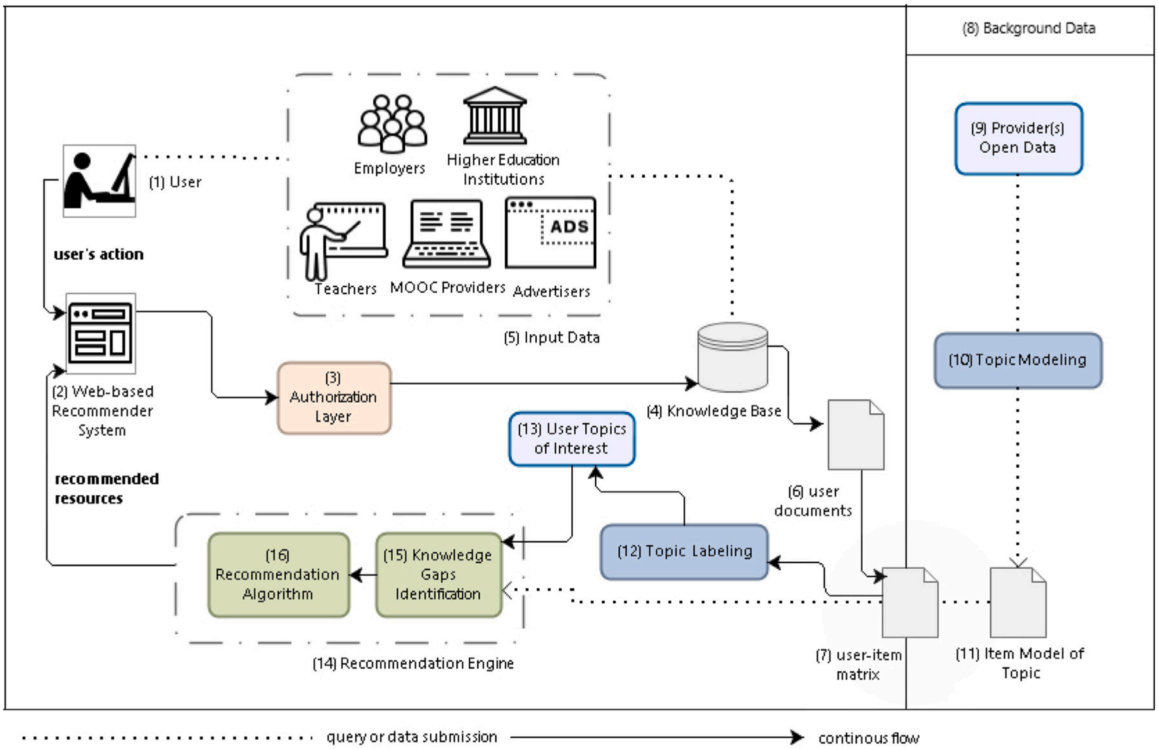

Fig. 2. The FReME proposed conceptual model.
Source: Adapted from Campos et al. (2020b, 2020c).

The modified NMF method includes a set of data treatments, starting by transforming the entry data from providers into a grouped list in UTF8 by clearing special characters. Next, it applies the Bag of Words Model, where each document is represented by a vector of unique terms with a "LemmaTokenizer" and a "TfidfVectorizer". Using this Vectorizer, the method creates a Document–Term Matrix by assigning a weight to each term. Therefore, it is possible to extract the vocabulary of this matrix (Campos et al., 2020a).

4.1. Topic labeling technique

The topics generated by topic modeling techniques can be complex to understand, given the lack of knowledge about the domains or specific terms, or even the difficulty of defining a theme for the topic between several words. This difficulty can impair the purpose of modeling, which is to extract central themes from documents. In this case, the topic labeling technique supports automatically selecting a word (label) to define the topic area or theme (Magatti et al., 2009).

In our work, the topic labeling (Fig. 2, see 12) follows some steps based on another method proposed by Nolasco and Oliveira (2016a). The first step is the "Selection of Candidates" which uses a sample of terms from the most significant documents associated with the topic we want to define a label. This relevance between topics and documents is possible since each topic has an associated probability with documents and their words. We select the top-D documents associated with the target topic θ, where D is freely estimated in the application. Next, we iterate the top-D documents to select the primitive labels adopting two different text extraction approaches: (a) the Keywords Selection (KS); and (b) the Text Selection (TS) with the Fast Keyword Extraction algorithm. We compare them to select the one that better represents our topics in Section 5.3.

The advantage of TS is the simplicity of extraction. Whereas KS extracts terms defined by the author to describe the whole and considering that not all collections have labels provided by the authors, in TS the terms are simply extracted from the text body. This factor makes TS the most used form of candidate selection, with many authors proposing sophisticated extraction possibilities such as the fast keyword extraction algorithm. Otherwise, when the labels are defined by the authors, KS has the advantage of well describing the area, since the keywords already defined may be more relevant to the documents (Nolasco & Oliveira, 2016b).

In TS with Fast Keyword Extraction algorithm, the system considers as primitive label any word between stop words and phrase delimiters (such as commas). Then, the system selects as "candidate label" only the primitive labels that are in the top-T terms of θ, where T is the number of terms freely estimated.

In KS, the documents' keywords are considered primitive labels. One of the changes we made to the originally selected topic labeling method is to consider all keywords as candidates when the approach is KS, regardless of whether or not they are contained in top-T. This decision was made since we do not consider n-grams in the modeling but many keywords are n-grams, so it would be interesting to consider these full keywords as a possible label.

Next, we apply the Term Frequency (TF) in the "Ranking" step to assign points to each "candidate label" according to its relevance (term frequency of occurrences) to the topic. Usually, Inverse Document Frequency (IDF) is applied to supply the high scores that TF applies for stop words and non-relevant terms (e.g., common verbs) (Nolasco, 2016). In the case of FReME, IDF is not applied since we delete the stop words in the previous step, and considering that only the candidates in the list of the most relevant terms W are selected, which removes the set of non-relevant terms.

Lastly, the "Selection of Labels" step considers the labels and their relevance from the previous step to select the first label of the list, i.e., the term that most represents the topic being considered a label. There is another difference between KS and TS here: in KS the top-1 term is selected as a label, whereas in TS we also select the top-3 labels. Algorithm 1 represents how these steps were implemented in FReME.

Algorithm 1: MOOCs ecosystems automatic topic labeling. Source: Campos et al. (2020b)

```
   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
1  topTerms = getTop10T(model, 10, k, vec);
2  for topic_i = 1 to k do
3      top_D = getTop30D(snippets, W, topic_i, 30);
4      top_T = topTerms[topic_i];
5      for d = 1 to top_D do
6          dt_label = dt_label + getPrimitiveLabels(d, approach);
7      end
8      if approach is TS then
9          for primitive = 1 to dt_label do
10             if primitive in top_T then
11                 list = list + primitive;
12             end
13         end
14     else if approach is KS then
15         list = dt_label;
16     end
17     candidates = applyTFtoRank(list);
18     top-1 = getTopLabel(candidates, 1);
19     top-3 = getTopLabel(candidates, 3);
20 end
```

As shown in Algorithm 1, in each item topic (line 2) the system selects the top-10 documents *D* (line 3), and the top-10 terms *T* (line 4) associated. Then, the primitive labels are extracted (line 6) for each document (line 5). Primate labels are extracted by "getPrimitiveLabels(d, approach)" where the *approach* can be KS or TS. Then, the system checks whether this primitive label is contained in the list of top-10 terms *top_T* if the *approach* is TS (lines 8 and 9). If so, this label is added to the *list* (line 10). We consider all keywords as candidates if the *approach* is KS (lines 11 and 12). These candidate labels are ranked according to TF in line 13. Finally, top-1 and top-3 are extracted from the ranked list of labels, respectively in lines 14 and 15.

4.2. Knowledge gap identification and recommendation method

Once item topics (Fig. 2, see 11) are generated using NMF with their respective labels and after collecting user documents (Fig. 2, see 6), the recommendation engine (Fig. 2, see 14) consists in cross-referencing these elements in the documents–terms matrix. As shown in Fig. 3, to detect this relationship *rel_topics*, we vectorize and transform each user profile document into an item of the generated items model (Fig. 2, see 11) using nmf.transform so that each document is associated with one or more topics. Thus, it is possible to verify which topic has a higher association probability *tmax* for each of such documents *ud*, represented by the notation *rel_topics = [ud, tmax]*.

Since such *tmax* already have their proper labels automatically generated by our solution (see Section 4.1), the system extracts such labels to generate the "user topics of interest", i.e., it is generated by the labels from all the most associated topics that have been identified. As shown in Fig. 4, such labels are transformed into a wordcloud using matplotlib and wordcloud Python libraries. Word weights are defined by the number of times they appear in the set.

Next, in order to find the relevant modules to the target user, the system concatenates the user documents into a single search string that represents the user knowledge, enrolled or obtained. The system calculates the Euclidean distance between the search string and the item model containing item documents so that it is possible to sort the results from the least distant values onwards and, finally, select a set of the closest documents *s* to the user profile, as shown in Fig. 5.

The result with the closest documents is represented by the document identifier. To show them back to the user, it is necessary to make them more informative, i.e., to collect some specific information in the document-oriented database. This key information could be represented only by the module title. However, the same titles could exist in different providers. Thus, a text clipping is created containing: module title, module URL, provider identifier, linked exercise or video URLs (exclusively for Khan Academy), and course URL. Fig. 5 shows the recommendation result, i.e., a set of documents *s* that can be recommended to the target user.

Although *s* represents relevant documents to the target user, it is still necessary to consider the student knowledge gap. In this context, FReME verifies which documents are new to the user, i.e., the documents in the list that have not been previously studied by the user. This verification is possible by comparing *s* with the user profile documents, contemplating different types of items according with providers. In case the user profile contains modules or courses, the system compares if the document in *s* is the same as the one in the user profile. In the case of video, if the student watched a video from a module, the system understands that the student is already enrolled or has already completed that module, so the document is not considered as a knowledge gap and it is not recommended, i.e., it is removed from *s*. As shown in Fig. 5, FReME selects the top-6 documents of *s* to show it back to the student.

4.3. Example of recommendation

This section presents a recommendation example to better visualize how the recommendation process works and provide a preliminary evaluation of the proposed steps. It simulates a target user (Fig. 2, see 1), Tom, interested in receiving recommendations on his knowledge areas, such as software development, business, and photography, in order to check which content would better fill his knowledge gap. Upon accessing our recommendation system, a start screen (Fig. 2, see 2) asked Tom to select and authorize (Fig. 2, see 3) which providers the system would extract information from. Among the options, Tom has selected and authorized extraction from Khan Academy, Udemy, and edX. After authorization, the information retrieved at this stage is Tom's curriculum information, such as course history from providers. This information is grouped in the Knowledge Base (Fig. 2, see 4). All these documents together represent the user profile in the MOOCs ecosystems.

To populate the item layer of the user–item matrix (Fig. 2, see 7), the system uses the item model of topics based on multiple provider modules extracted from Background Data (Fig. 2, see 8). Next, the Topic Labeling step (Fig. 2, see 12) generates a label for each topic. With the matrix filled in, the information is sent to the recommendation engine (Fig. 2, see 14), which verifies the link between each of these documents with the previously generated item layer topics. Given k=14, where k represents the number of the best topics clustering (this result is presented in Section 5.2), it is possible to verify the weight of the association between user profile and topics. Therefore, it is necessary to vectorize and transform these documents as though they were documents of the item model (using nmf.transform) so that each document is associated with one or more topics. For the target student profile evaluated in this section, the *rel_topics*, relationship between each user document *ud* and its most associated topic *tmax* can be represented through the notation *rel_topics = [ud, tmax]* as follows: [0,6; 1,12; 2,10; 3,12; 4,7; 5,10; 6,0; 7,10; 8,10; 9,10; 10,10].

The strongest relationship between topic and documents is at [2; 10], where document 2 ("Creating webpages") is closest to topic 10 (represented by labels "app", "code", and "image"). It is possible to check all the topics that appear most closely associated and extract

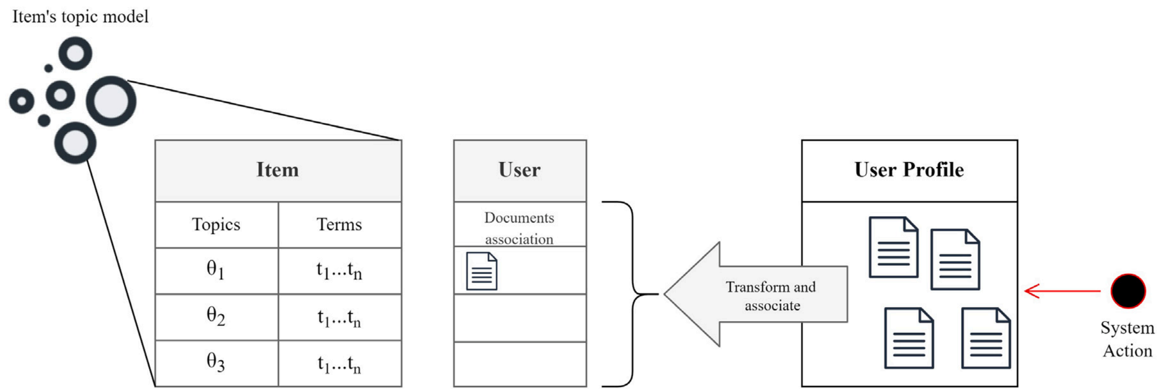

Fig. 3. Association methods between item topics and user documents.

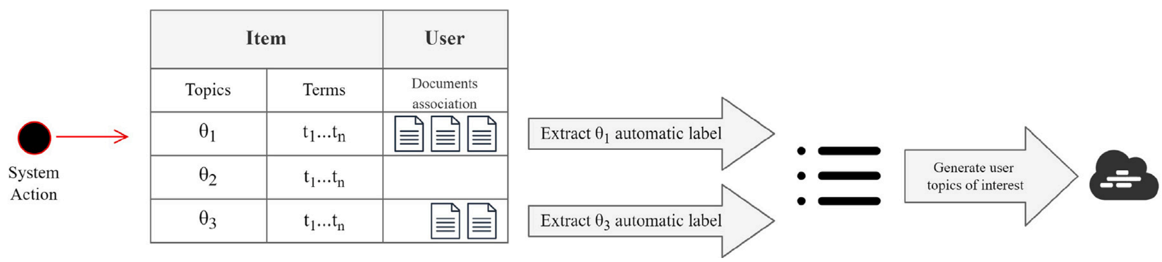

Fig. 4. User topics of interest generation.

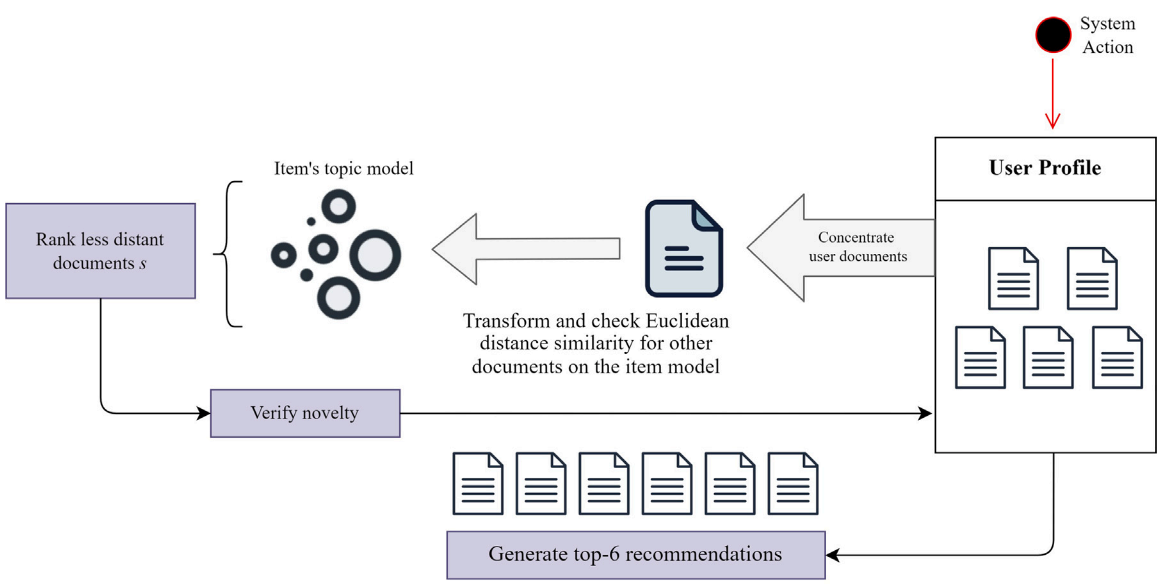

Fig. 5. Recommending parts of courses based on the selection of the knowledge gap.

the User Topics of Interest (Fig. 2, see 13). Our User Topics of Interest extracts the TS (top-3) labels from each *tmax* of the user's *rel_topics* since the TS (top-3) approach is the one that has the best results, according to the evaluation conducted in Section 5.3. Therefore, each topic has 3 labels that indicate the content of their respective best-associated documents. From these labels, it is possible to build a wordcloud. Considering the word weights and the fact that the user has a large amount of *tmax*=10 in *rel_topics*, the labels of topic 10 become more prominent in the word cloud. This example is illustrated in Fig. 6.

From this processed user data, we apply the content-based method described in Section 4.2 to verify which item documents are closest to the identified user profile. Therefore, the recommended items for Tom are all from Udemy in the following order: "Organization and Productivity Apps" (https://www.udemy.com/course/macmojave/), "Business Logic and Process Automation" (https://www.udemy.com/course/salesforce-platform-app-builder/), "We Want To Give You An Introduction To This Course Before Diving In" (https://www.udemy.com/course/ethereum-developer/), "Making Decisions" (https://www.udemy.com/course/starting-out-with-visual-c-sharp-coding-exercises-for-beginners/), "Pre Flight Planning Flight Day Management" (https://www.udemy.com/course/drones-aerial-videography-and-photography/), and "Information Displays" (https://www.udemy.com/course/flutter-mobile-development/).

It is also worth mentioning that applying the topic modeling technique (Fig. 2, see 12) directly impacts the similarity between user

Fig. 6. User topics of interest generated from target user data.

documents and item layer documents. Such impact is clear if we model item layer documents with a value of k not equal to 14. For instance, when generating a model with k=13, and if we apply our recommendation to the same target user, the recommended documents change, as follows in order: “Video answers to Questions”, “System Software”, “Excel Essentials Level VBA Programming”, “Model based Systems Engineering Foundations”, “Combinatorial Mathematics”, and “Editing Audio in Audacity”. The 4th and 5th documents are from edX, whereas the others are from Udemy.

Provider integration also directly impacts the obtained results. For example, it is possible to simulate what recommendations would be if only two of the providers (Khan Academy and edX) were used resulting in all modules from Udemy as follow in order: “We Want To Give You An Introduction To This Course Before Diving In”, “Deep Learning with R”, “Business Logic and Process Automation”, “Node Glossary”, “ESXi Host and Virtual Machine Performance Analysis and Tuning”, and “Troubleshooting Server Problems”.

Only one item is common between the effective recommendation using data from three providers and the simulated recommendation with two providers. There is also a change in the order this item appears. Whereas in the three providers’ recommendations there are two other items in front of the ranking, in the partial recommendation this common item is at the top of the list, not identifying the first two in any top-6 position. It is also possible to check what the recommendations would appear using only modules enrolled from Khan Academy. The results follow in order (2nd, 3rd, and 5th from Khan Academy and the others from Udemy): “We Want To Give You An Introduction To This Course Before Diving In”, “Module Connecting algebra and geometry through coordinates”, “Human anatomy and physiology”, “Purchasing Processing”, “Module Linear equations”, and “Troubleshooting Server Problems”.

5. Evaluation and results

This section presents an evaluation method based on both qualitative and quantitative analyses that is composed of a set of three experiments conducted to verify our work and validate its implemented techniques, i.e., the topic modeling, the topic labeling, and the recommendation method itself based on real data from multiple providers. Therefore, the first experiment compares the effectiveness of our improved topic modeling technique with a widely used topic modeling technique - LDA. The labeling technique representativeness is verified in our second experiment, through a comparison between the topic labels generated with our approaches and the labels from MOOCs providers. The last experiment focuses on evaluating the recommendation engine through users’ feedback and recommendation system metrics.

Table 2
Overview of the dataset used in the three experiments.

Provider	Modules (Documents)		Distinct terms	Distinct areas	Distinct courses
Khan Academy	1,705	≈1.6%	23,888	14	258
Udemy	101,177	≈94.94%	257,250	113	9,657
edX	3,692	≈3.46%	28,455	476[a]	1,627
Total	106,574	100%	–	–	–

[a]A course at this provider may be in one or more areas. In this case, areas are represented by grouping strings. The "Distinct Areas” column checks only identical groupings or areas.

5.1. Dataset

The dataset used in the three experiments contains the entire implemented provider’s data, resulting in 106,574 modules (documents), as better described in Table 2. The fields used for the recommendation are module contents information, module title and description, videos/exercises/articles information, and the course short description (in the case of edX). Some data treatments are performed as well as for item topic modeling, such as non-noun removal, stop words removal, punctuation removal. Almost 95% of the dataset comes from Udemy. Despite being a higher portion compared with other providers, this dataset is a representation of the real world. This implemented integrated dataset fills the background data, whereas the user profile is loaded when authorized by the own user.

5.2. First controlled experiment: topic modeling effectiveness

The stability analysis automatically defines the best number of topics given a dataset. Our evaluation verifies if this approach can be used in NMF and applied to MOOCs. Therefore, we check the topic coherence values for the same range of topics both in the modified NMF and in LDA to identify the best topic number. We verify topic coherence using the TC-W2V metric (O’Callaghan et al., 2015), which builds a word embedding model based on the model since this metric is appropriate for use with a large reference corpus. This metric uses the Word2Vec model, which in our scenario is possible through the Gensim implementation that estimates word representations given a vector space (O’Callaghan et al., 2015).

5.2.1. Data preparation

When applying the modified NMF (Campos et al., 2020a) for the item layer, we use the total of 278,079 distinct documents’ terms, which result in a matrix created with the library Scikit-learn of a size of 106,574 x 278,079. The TfidfVectorizer exclude stop words, considers TF and IDF, and associate each item with each associated document attributing a weight.

Next, as part of the stability analysis, the modified NMF requires to inform the range of topics, i.e., a minimum $kmin$ and maximum $kmax$ number of topics so the system can test which have a better topic coherence. Greene et al. (2014) reinforce that this pre-defined range is a strategy adopted in the most common approach to stability analysis. The definition of the range values is empirical. Some experiments, such as the one conducted by Greene et al. (2014), are based on the “ground truth” labels provided by the metadata of the datasets. As the datasets used in our experiment do not provide a “ground truth”, the range was an approximation with a higher margin of the macro-categories existing in the providers (i.e., Udemy are 13, Khan Academy are 12, edX are 6). Therefore, in this experiment, we defined $kmin$ = 5 and $kmax$ = 30. When implementing the modified NMF, the matrices W (document–topic) and H (topic–term) are initialized, and the coherence is calculated for each topic number k in this range (kmin to kmax), where the value of k with the greatest coherence is selected to model the collection. The result for the adopted collection was the best number of topics k = 30.

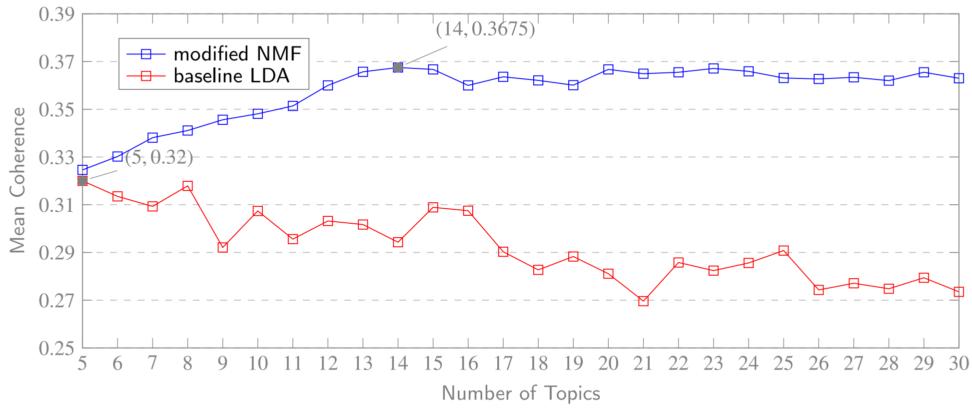

Fig. 7. Topic coherence in each k-value for the item layer using modified NMF and LDA.
Source: Adapted from Campos et al. (2020b).

When visualizing the top terms of the generated topics through H, it is possible to understand and infer about the subjects and themes of each topic. At this point, human analysis may point to some modeling flaws, i.e., there are opportunities to optimize the topic modeling to obtain the top terms closer to the reality. The modified NMF (Campos et al., 2020a) contains some adaptations to address these issues, including increase the stop words list and remove courses' titles and descriptions. After these adaptations, the vocabulary changes to 31,525.

However, there is another need to adjust the topic modeling to the used dataset by applying grammar class filtering according to the purpose of the topics. In our case, the most important words for the topics' definition are the nouns, since they name the knowledge areas and their descriptions more specifically than the verbs or adjectives (which may appear in common in any area). Therefore, the removal of all non-noun words is executed by tokenizing the words in the corpus using the Natural Language Toolkit (NLTK) word tokenization, tagging tokens grammatically, and so removing all tokens that are not in the desired grammar class. NLTK has a specific way of part of speech (POS) tagging. For this method, we leave only the words NN (Noun, singular, or mass), NNP (Proper noun, singular), NNS (Noun, plural), and NNPS (Proper noun, plural).

Then all subsequent steps were reprocessed until it achieves new values. The results point to new vocabulary with 25,850 terms (reduction of 90.7% from the originally selected vocabulary) which reflects on the dimensions of matrix A, which becomes 106,574 x 25,850. The best coherence identified in the topic range was 0.3675 for the set of 14 topics.

5.2.2. Baseline comparison and results

In this first experiment, we compared the topic coherence obtained using our model with the topic coherence using LDA, from the topic descriptors terms. Based on this comparison, it is possible to capture the topics' semantic interpretability, which is one of the most suitable criteria in the topic modeling evaluation (O'Callaghan et al., 2015).

Therefore, we apply our dataset in the baseline LDA model with the same range (5 to 30), the same stop words, the Lemmatization process, and the only noun criteria. Therefore, we use a Word2Vec model built from the corpus as the input, so we can have a similar scenario with the modified NMF's TC-W2V. Fig. 7 shows that after generating topic coherence for each k, the best coherence when using LDA was less than the topic coherence obtained when using our modeling.

The results presented in Fig. 7 show some approximation between the coherence value of the modified NMF and LDA. For instance, when $k = 5$, LDA has its best value of 0.32, whereas our modified NMF has its lowest value of 0.3246 which is bigger than LDA. The modified NMF model obtains higher results in all k options. The mean (SD) is also higher in the modified NMF with 0.3574 (0.011) while LDA achieves 0.2925 (0.014). Moreover, the modified NMF achieves a median of 0.3628 and a range of 0.3246–0.3675 while the baseline LDA reaches 0.2905 of median and a range of 0.2696–0.32.

Through the comparisons, it is possible to conclude that this experiment validated our modified NMF in the context of MOOCs ecosystems since it improved the topic modeling when compared with the LDA. This experiment also allowed us to verify the consistency since we applied the entire MOOCs providers' available database which includes several data scenarios and course areas. Therefore, the modified NMF is relevant to integrate the recommendation system as it best represents its multiple provider documents.

5.3. Second controlled experiment: labeling technique representativeness

The second controlled experiment aims to evaluate the topic labeling step through comparisons between our automatic labels (detailed in Section 5.3.1) and providers' labels, using the cosine distance (see Section 5.3.2). Therefore, the most appropriate automatic labeling approach is the one that gets the closest proximity to the provider's manual label.

5.3.1. Generation of labels

The automatic label generation follows the instructions of Section 4.1, and the Algorithm 1 is executed two times with different inputs: (a) the first considering *approach* = TS; and (b) the other *approach* = KS. In both D = 30 and T = 10. Considering that our documents do not contain keywords in their structure, we select the knowledge area from the course where the module is allocated as the keyword in the "Selection of Candidates" step. We generate labels for each of the 14 topics modeled in the previous section. To track how this data is handled in practice, it is possible to follow the process for the first topic, as described next.

Approach = TS. Starting with the "Selection of Candidates", we selected top-30 documents associated with the topic. In the case of the first topic, some document name repetitions occurred (e.g., courses with same name in different modules or providers), such as in "Classes Objects". This list of documents also indicated that titles have been treated before generating models, e.g., special characters were removed. Next, to select the "candidate labels" we selected the top-10 terms in the topic, resulting in the following terms in the case of topic 1: ["class", "method", "object", "java", "array", "inheritance", "variable", "constructor", "code", "loop"]. It is noticed that the terms are all nouns and with less variations (e.g., all "classes" became "class") due to treatments performed.

For the “primitive labels” selection using TS, the Fast Keyword Extraction algorithm iterated each one of the top-30 documents extracting the words. The primitive labels went through some removals such as the exclusion of non-nouns. Therefore, to select which of these “primitive labels” are “candidate labels”, we selected all primitive labels that are in the top-10 terms of the topic. In the case of topic 1, all terms in the top-10 terms were selected. The “Ranking” step applied TF so the ranked candidates and their frequency are as follow: [“class” 754, “method” 315, “object” 241, “constructor” 105, “java” 104, “variable” 103, “inheritance” 90, “code” 85, “loop” 17, “array” 13]. Finally, in the “Selection of Labels” we had TS (top-1) = “class” and TS (top-3) = [“class”, “method”, “object”]. In other topics, there were cases of substitution, where one label was a substring of another label in the top-3 selection.

Approach = KS. The results of the first procedures in the “Selection of Candidates” step (i.e., select the top-30 documents and the top-10 terms associated with the topic) in KS were the same as in TS. The differential of the KS starts in the “primitive labels” step where the keywords in each one of the top-30 documents is selected. For the case of topic 1, the primitive labels are all “programming language” except for the 10th and 14th labels which are both “web development”, indicating a large concentration of course modules inserted in the “programming languages” area.

Differently from TS, in KS all the “primitive labels” are also “candidate labels”. The TF is also applied in the “Ranking” step that retrieved the result as follows: [“web development” 2, “programming languages” 42]. The “Selection of Labels” step resulted in selecting KS = “programming languages” for topic 1.

5.3.2. Comparisons and results

When comparing to provider labels, since each topic contains modules from multiple courses and providers, there is no common provider label for all modules of a topic. Therefore, our goal was to find strings in the modules that could define as close as possible to a label.

We considered labels differently in each of the providers selected for this experiment. At Khan Academy, the label is from the area where the courses and modules are inserted. In edX courses, the label is characterized by course subjects, which can be even more than one. In Udemy, the label could be selected by course category or course subcategory. We select the subcategory since it is less generic.

Thus, we used these strategies to select provider labels from the top-30 documents of each topic, making it possible to create a comparison. The comparison is made by calculating the distance of the provider labels to our automatic labels based on TS (top-1), TS (top-3), and KS. One way to calculate these distances is by using cosine distance which resulted in a closer proximity of the implemented TS (top-3) approach to provider labels as shown in Table 3.

Topics 0 and 12 are the only exceptions, both with KS as their closest approach what is justified by the fact that these are topics that have respectively 93% and 96% of their top-30 documents in a single area. As KS selects by keywords and considering the TF of the candidate labels, we get a KS label that approximates almost all of the provider labels we are comparing. The same is not true in the other topics, which have more mixed areas and, consequently, harm an approximation of such provider labels with the automatic label KS.

Given these results, we can conclude this experiment by stating that our topic labeling technique can automatically calculate labels for the modeled topics, with better results in our TS (top-3) approach. This automatic identification helps in describing item layer topics and in identifying the “user topics of interest”. Therefore, obtaining relevant strings to represent topics, verified in the present evaluation, contributes to reduce the students’ knowledge gap by identifying their interests.

Table 3
Distance between strings in each topic labeling approach.
Source: Campos et al. (2020b)

θ	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)

5.4. Third controlled experiment: user perspective

The recommendation evaluations in the recommendation systems are derived from the evaluation techniques of the Information Retrieval field. The first aspect to be observed in evaluations involving users is that in many cases the system depends on user feedback, i.e., a response to the recommendations that the system provided. This feedback, also called annotations, can be captured explicitly or implicitly. In this work, we collect explicit feedback from each participant (represented as a target user in the recommendation system) through a developed web system.

5.4.1. Evaluation metrics

We adopt some of the metrics specified by Ricci et al. (2015) to evaluate the recommended modules, as described next.

Mean Average Precision (MAP). *MAP* considers the ranked list of items and the concept of precision P, which in recommendation systems is applied as the fraction of top-N recommended items that are relevant to the user. When collecting the feedback, the user can inform relevant items m. For instance, given 6 recommended items, if a user judges m=4 (indicated in a vector with 0 for not relevant and 1 for relevant, as in: [0,1,0,1,1,1]), the precision P is the equivalent of 4/6, so P = 0.66. Precision can also be calculated by giving a cutoff k, the so-called $P@k$. In this case, only the subset of recommendations from rank 1 to K is considered. As such, $P(k{=}5)$ would result in 3/5, i.e., $P(k{=}5)$ = 0.6.

$P@k$ and m are used to calculate the average precision AP. If in a new example we assume that from these m=4 relevant items, not all were recommended, the relevance vector would be [0,1,0,1,1,0], i.e., only 3 recommended items would be relevant. For the first calculation step, it is necessary to identify the $P@k$ of each recommended item, where the value of k is equal to the position of the item. Therefore, the first relevant value in position 2 has a P=1/2, as the cutoff identifies only 2 items being 1 relevant. The same is followed for other items until reaching the vector [0, **1/2**, 1/3, **2/4**, **3/5**, 3/6]. Next, it is needed to sum all the P’s of those relevant items (bolded), resulting in [(1/2) + (2/4) + (3/5)] = 1.6. Finally, the result is multiplied by $1/m$, i.e., 1/4, so AP = 0.4. *AP* is higher as the recommendation has more hits and the higher these hits are. *MAP* is the mean of all participants’ *AP*.

Utility. The recommendation utility may be more relevant than the accuracy in some scenarios, since it indicates the value of the recommendation to the user or the system. In our context, the utility was verified from the perspective of users (i.e., students). One of the ways to capture utility is through user ratings, which can even include negative values (e.g., in case a recommendation might offend users).

Novelty. Novelty evaluates if the recommended items are new to the users, i.e., when they do not yet know. In our case, this information could be confirmed in the user's profile by verifying the existence of a bond in the recommended module. However, it would not be enough since the student may have registered in this module with another account or in a provider not implemented in this evaluation. Therefore, the direct question to the user is an option to bring correct information about the novelty.

To calculate the novelty, we consider the ratio between the new modules that were recommended and the total recommended modules, as follows in Eq. (1).

$$Novelty = NN/NT \quad (1)$$

where *NN* is the number of recommended new modules and *NT* is the total number of recommended modules.

Confidence. The recommendation system confidence indicates if the system provides reliable recommendations. To do so, it calculates the ratio of all positive evaluations to the total number of valid evaluated modules as shown in Eq. (2).

$$Confidence = NP/NT \quad (2)$$

where *NP* is the number of positive marks given to recommended items and *NT* is the total number of recommended modules.

This confidence percentage can influence the decision of other users regarding the items recommended. Users prefer systems that have high confidence. For situations where confidence is low, the user needs to read about the item to make sure that it is really interesting to obtain it. Therefore, this metric can represent a "tiebreaker" criterion in the case of two recommendation systems with similar accuracy. The one that obtains the greatest confidence is the most preferred.

5.4.2. Hypothesis and variables

The experiment was based on the Goal-Question-Metric (GQM) paradigm (Basili, 1992), where we **analyze** the recommendation method created **with the purpose of** evaluating the quality of parts of courses recommendations **with respect to** user perception and satisfaction and the properties MAP, Utility, Novelty, and Confidence **from the point of view of** MOOCs students **in the context of** the modules offered by the selected platforms: Udemy, edX, and Khan Academy (Campos et al., 2020b).

From this definition, we conducted the experiments to validate whether the proposed recommendation method achieves efficacy of less than 50% in the properties of MAP, Utility, Novelty, or Confidence. More specifically, from all the recommended items and collected feedback, it was possible to test the third experiment hypothesis:

- Null hypothesis *(H0)*: The proposed recommendation method achieved efficacy of less than 50% in the properties of MAP, Utility, Novelty, or Confidence. Therefore, *H0* = *(μMAP_OurApproach < 50%) OR (μUtility_OurApproach < 50%) OR (μNovelty_OurApproach < 50%) OR (μConfidence_OurApproach < 50%)*, where *μMAP_OurApproach* = MAP, *μUtility_OurApproach* = Utility, *μNovelty_OurApproach* = Novelty, *μConfidence_OurApproach* = Confidence, which refer to our recommendation and were collected through user feedback.
- Alternative hypothesis *(H1)*: The proposed recommendation method achieved efficacy greater than 50% in the properties of MAP, Utility, Novelty, and Confidence; therefore, the null hypothesis is rejected. Therefore, *H1 = (μMAP_OurApproach ≥ 50%) AND (μUtility_OurApproach ≥ 50%) AND (μNovelty_OurApproach ≥ 50%) AND (μConfidence_OurApproach ≥ 50%)*.

In this experiment, the independent variables are the user information that influences the recommendation, i.e., login, titles and descriptions of the studied modules, titles and descriptors of the content modules studied, titles and descriptions of the videos, exercises, or articles studied. Dependent variables are the recommended modules, the recommended modules position, and the variables collected by explicit feedback indicating if users are satisfied with the recommendations.

5.4.3. Instruments and preparation

A web system was developed to collect data from participants with more control and extension possibilities. We used a virtualenv (virtual environment for Python project) with Python 3.6 and the web technologies Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, Bootstrap, and Django. The system was deployed using Heroku as the application hosting and Amazon Web Services (AWS) to store the media associated with modules. The technology choices were made for convenience, not being restricted to those.

The system contains an authentication layer where each participant can log in with their unique credentials. Next, an informed consent acceptance screen shows information about the study, the evaluation informed consent, the stakeholders, and the data privacy so that users can indicate if they want to proceed. We collect personal information from users, besides the usage profile in MOOCs and other learning platforms, through a questionnaire to characterize participants. In the last steps, each participant must answer some specific questions about the 6 modules recommended by our system, so that it is possible to collect feedback from the recommendation item itself.

We invited participants from different motivations and personal profiles, who were enrolled in MOOCs platforms or online courses in another learning environment. First, we collected their recommendation inputs with their authorization to access provider data. Next, we processed the offline recommendation, resulting in the top-6 recommended modules or courses for each participant. Finally, we imported their results into the web system and they had a deadline to access the system with their credentials, evaluate, and fill in the other required information. It is worth mentioning that participants could decline participation or leave the system at any time.

5.4.4. Execution and analysis of results

From the 27 participants invited to the study, 19 completed all the required steps and each participant was considered as a specific subset. Through the participant characterization questionnaire, it was possible to collect some relevant aspects. When distributing them in age groups, the responses indicate that ≈5.3% is under 20 years old, ≈78.9% between 20 and 30 years old, and ≈15.8% is over 30 years old. When distributing them according to the Brazilian geographic region, ≈52.6% are from the Southeast region, ≈10.5% from the Northeast, ≈10.5% from the Midwest, ≈5.3% from the South, and ≈5.3% are from the North. ≈15.8% live outside Brazil. ≈79% identify themselves as male, whereas ≈21% as female.

Their professional profile is distributed as follow: ≈31.6% work in the "Exact and Earth Sciences" area, ≈15.8% with "Humanities", ≈15.8% with "Linguistics, Letters, and Arts", ≈10.5% with "Biological Sciences", ≈10.5% with "Engineering", ≈5.3% in the "Health Sciences" area, and ≈10.5% have "no vocational training". The assignment of participants' areas of activity contributed to the recommendations being evaluated in the most diverse scenarios, as the three selected providers include courses that cover the areas of all participants.

We consider that the module recommendation should be useful for anyone interested. Even those who have no learning experience on MOOCs platforms. The questionnaire allowed to choose some options with a scale from 1 to 5 to verify the participant's knowledge level in MOOCs usage, where 1 represents "Very Low" (≈15.8% of participants), 2 is "Low" (≈26.2% of participants), 3 is "Average" (≈5.3% of participants), 4 is "High" (≈15.8% of participants), and 5 is "Very High" (≈5.3% of participants). Those who had no experience in the platform could opt for option 0 indicated as "no usage knowledge" (6 participants or ≈31.6% indicated this level). Among the 6 participants who answered with option 0- "No usage knowledge", 5 answered to know other online learning platforms, among them (mentioned in an open text field): YouTube, Duolingo, and Virtual Learning Environments. Other participants also described using other platforms besides MOOCs.

Participants also estimated the hours per week that they use other learning platforms. ≈68.4% indicated between 1 and 5, ≈15.8% between 5 and 10, ≈10.5% use 30 or more, while 1 participant (≈5.3%) indicated that do not use other learning platforms.

Participants also answered some specific questions regarding the use of MOOCs. About the platform that they have the preference for when accessing MOOCs, most of them (≈52.6%) indicated that they have no preference for a platform, which shows a positive aspect regarding multiple providers' recommendations. Other participants also indicated the preference for Khan Academy (≈5.3%), edX (≈15.8%), Udemy (≈15.8%), or other (≈10.5%).

Given the answers extracted from the 19 subsets, it was possible to verify the results for each defined metric:

MAP result. The evaluation of these subsets was performed given our recommendation of N=6 modules to 19 different users who judged m modules as relevant, where $m >= 0$ and $m <= 6$.

Users had to answer the question "Would you find it relevant to learn this content?" in all 6 modules, enabling the participant to answer "Yes" or "No". The good scenario is represented in the case that the *AP* result is as close as 1, i.e., all 6 recommended modules judged as "Yes" (relevant) by the user, or that as many of the relevant modules appear at the top of the recommended vector. The results of the evaluation with the *AP* values for each participant are shown in Table 4. Given the respective *AP@6* of each user, it is possible to calculate *MAP(@k)*, i.e., MAP given the cutoff k, which reached 62,24% in our case.

Utility result. Participants were asked to answer: "How useful would this content be for you?". They answered on a Likert scale where 5 means totally useful, 4 means possibly useful (both positive), 1 means totally useless and 2 means partially useless (both negative). Therefore, utility is the ratio between the number of modules rated with grades 4 and 5 and the number of modules rated with grades 1, 2, 4, and 5.

Given N and the number of participants, the system totaled 114 recommended modules or courses. 62 answers (54.38%) indicated options 4 or 5. The total number of items that were evaluated with options 1, 2, 4, or 5 was 90 (78.94%). Therefore, the system utility has reached 68.89%.

Novelty result. Participants were asked to answer: "Have you taken this course before?". Answers are restricted to "Yes" (when the participant has previously studied that content) or "No" (if they have never studied the content before).

Given NT = 114, and that all but one of the answers indicated that the recommended module was new, our experiment achieved a novelty of 99.12%.

Confidence result. We consider as positive evaluations that obtained "Yes" as an answer to the question "Would you find it relevant to learn this content?". To calculate confidence, we apply Eq. (2). Given a total of 83 positively evaluated modules, the results point to a total of 72.81% confidence.

Third experiment results. The conducted experiment involving students with answers collected via web system and processed through metrics showed that FReME was more than 50% effective in all the evaluated properties. This result confirms the alternative hypothesis. Fig. 8 shows a compilation of the results obtained in this experiment and described in this section.

This evaluation with real participants from different areas and with different motivations regarding MOOCs also provided instant feedback. Participants P8 and P17 reported two instabilities in the web system which were then corrected. Participant P15 exposed difficulty in understanding the items to be evaluated because it was in the English

Table 4
Calculation of the average precision of each subset.

User Id	m	Recommendation vector	P@K's	AP@6
P1	6	[1,1,1,1,1,1]	[1/1, 2/2, 3/3, 4/4, 5/5, 6/6]	1
P2	4	[0,1,1,1,1,0]	[0, 1/2, 2/3, 3/4, 4/5, 0]	0.4527
P3	2	[0,0,1,0,1,0]	[0, 0, 1/3, 0, 2/5, 0]	0.1222
P4	4	[1,1,0,1,1,0]	[1/1, 2/2, 0, 3/4, 4/5, 0]	0.5916
P5	6	[1,1,1,1,1,1]	[1/1, 2/2, 3/3, 4/4, 5/5, 6/6]	1
P6	5	[1,1,0,1,1,1]	[1/1, 2/2, 0, 3/4, 4/5, 5/6]	0.7305
P7	5	[1,1,1,1,1,0]	[1/1, 2/2, 3/3, 4/4, 5/5, 0]	0.8333
P8	5	[1,0,1,1,1,1]	[1/1, 0, 2/3, 3/4, 4/5, 5/6]	0.675
P9	5	[1,1,0,1,1,1]	[1/1, 2/2, 0, 3/4, 4/5, 5/6]	0.7305
P10	4	[1,1,0,1,0,1]	[1/1, 2/2, 0, 3/4, 0, 4/6]	0.5694
P11	5	[1,0,1,1,1,1]	[1/1, 0, 2/3, 3/4, 4/5, 5/6]	0.675
P12	1	[0,0,0,0,0,1]	[0, 0, 0, 0, 0, 1/6]	0.0277
P13	5	[1,0,1,1,1,1]	[1/1, 0, 2/3, 3/4, 4/5, 5/6]	0.675
P14	5	[1,1,1,1,0,1]	[1/1, 2/2, 3/3, 4/4, 0, 5/6]	0.8055
P15	6	[1,1,1,1,1,1]	[1/1, 2/2, 3/3, 4/4, 5/5, 6/6]	1
P16	3	[0,0,1,1,1,0]	[0, 0, 1/3, 2/4, 3/5, 0]	0.2388
P17	4	[1,1,1,1,0,0]	[1/1, 2/2, 3/3, 4/4, 5/5, 0]	0.8333
P18	4	[1,1,1,0,1,0]	[1/1, 2/2, 3/3, 0, 4/5, 0]	0.6333
P19	4	[1,1,1,0,0,1]	[1/1, 2/2, 3/3, 0, 0, 4/6]	0.6111

language. This requirement can easily be circumvented since we use NLTK and spacy which make it possible to choose from language-specific models (e.g., Portuguese, Chinese, French, among others) when implementing FReME. In this case, it would be necessary to implement providers that offer courses in the desired language (e.g., Veduca, Miríada X etc.).

We punctually consulted some of the participants where the recommendation system obtained *AP@6* < 0.5 to collect more information about the users' profiles that could justify the low result or propose feedback for future versions. Participant P3 is from the journalism area. However, P3 never took courses focused on this area in MOOCs. When considering the input data from Khan Academy, FReME extracted courses that are no longer of interest to that user. In this case, the user's interaction with FReME could be an interesting resource to improve recommendations. Participant P12 reported that it was not possible to understand the criteria for evaluating the first relevant question. Despite having marked maximum values on the utility-scale for most of the recommended items, the result of precision was incompatible.

Participant P16 reported that some of the recommendations do not indicate a module that this participant would enroll. In this case, FReME found similarities between P16's enrolled modules about programming area, recommending some mobile programming modules (e.g., "Creating Android apps Basics"), and these modules were precisely those that received "false" relevance. We did not get feedback from participant P2. However, it was possible to analyze the same. In most of the recommendations for the photography area, the participant rated positively, whereas for the "wedding photography" area (module "Event Wedding Photography") the participant rated negatively. User interaction in these cases would be important to filter which subarea of modules interests them (e.g., the possibility of excluding "mobile programming" and "wedding photography" in the filter).

5.4.5. Threats to validity

The internal validity depends on the number of participants who perform it. We prioritize having participants from different areas. More participants would contribute to increasing the study's internal validity. Nevertheless, the participants of this study performed the evaluations independently and most were not part of the same social group, which did not allow contact between those who have already participated and those who would still participate, also contributing to internal validity. The study's external validity can be verified by the present evaluation. Participants from different geographic regions with different profiles on the MOOCs platforms were selected, which ensures that the study is not restricted to a specific group of participants. The chosen participants

MEAN AVERAGE PRECISION (MAP)	UTILITY
N=6 modules	Answers indicated options 4 or 5: 62 (54.38%)
19 different users who judged m modules as relevant, where m> = 0 and m <= 6	Total number of items who were evaluated with options 1, 2, 4, or 5: 90 (78.94%)
MAP = 62,24%	**Utility = 68.89%**

NOVELTY	CONFIDENCE
Recommended modules: 114	Recommended modules: 114
Number of recommended new modules: 113	Relevant recommended modules: 83
Novelty = 99.12%	**Confidence = 72.81%**

Fig. 8. Compilation of the results obtained from the third experiment.

have no knowledge about FReME itself, thus ensuring the construct validity of the study. Moreover, they did not know the themes of this work. Regarding the study conclusion validity, no threats were identified that could hinder the ability to complete the present study.

5.5. Discussion

Besides showing the applicability to the MOOCs scenario and the FReME semantic interpretability, the greater topic coherence of NMF when compared to LDA in all of the possibilities also ratifies the suitability of NMF to niche or non-mainstream content (O'Callaghan et al., 2015). The value of the TC-W2V with NMF (0.3675) also remained satisfactory if we compare it with the range of experiments conducted in other scenarios that apply this metric, such as the TC-W2V of online news articles from the BBC (range of 0.2 – 0.3), or from The Guardian (range of 0.2 – 0.34) (O'Callaghan et al., 2015).

The results also reinforced the relevance of the FReME preprocessing steps, which allowed the exclusion of 252,229 terms without relevance to the model, contributing to greater consistency of the topics and, consequently, to a better recommendation. The automation of these steps, from data extraction with APIs that allow reading the JSON data, through preprocessing, including the topic labeling statistical method, the choice of document-oriented databases, and the stability analysis, guarantee the scalability of our proposal. This automation contrasts with recent solutions that addressed fixed topic values in MOOCs recommendation (Apaza et al., 2014; Bhatt et al., 2018) or that reported the need for a method of automatically identifying the best value of topics (Apaza et al., 2014).

Regarding the topic labeling, among the TS approach, TS (top-3) obtained better results, as there are more chances to be close to the compared provider labels. TS (top-1) can have labels that differ from all labels compared since we are using cosine distance between two sets of strings, but without considering the similarity between words, for example. Thus, similar words do not contribute to one string being close to another. A reason for TS (top-3) has achieved better results than KS is that provider keywords are not document keywords (from modules) so the word set in KS is more generic and less descriptive. The results obtained in TS (top-3) were close to the providers' labels, indicating that our proposal was able to automate this labeling process, which also helps in reducing the students' knowledge gap by identifying their interests.

Considering that the FReME interface is not addressed in this work, it is not practical to collect aspects of the users' perception of usability through the developed web system. However, Vieira et al. (2007) demonstrate that usability in computer systems can be verified through an example of use. Following this perspective, we point that the example of using FReME presented in Section 4.3 is part of this validation process. Therefore, it exemplifies the system's usability with a simulation of real data. The step-by-step exemplification illustrates the flow of system utilization. Thus, this property is not included in the hypothesis check.

When analyzing the user's perspective on the third experiment, results indicated MAP = 62.24%, Utility = 68.89%, Novelty = 99.12%, and Confidence = 72.81%, i.e., efficacy greater than the level established in the alternative hypothesis. A relevant aspect to be considered in the FReME recommendation is the fragmented recommendation of courses in multiple providers. Distinguishing the data structure at each provider and even the divergent extraction techniques required in this context make this task more complex. The recommendation example indicated that the use of multiple providers guarantees a broader extraction possibility and then a greater chance of success in the item recommendation, as well as a greater customization level in terms of the recommendation. Therefore, the treatment of multiple providers in the recommendation process is also a great opportunity for researchers and for the industry in the MOOCs domain. FReME facilitates the integration of the students' historical data in these providers and adopts techniques to recognize similar courses from different providers. The results also confirm the benefits of content flexibility when recommending parts of courses, delivering packages of personalized modules to users according to their knowledge gap.

6. Conclusion

This work investigated the problem faced by students in MOOCs in choosing the best course due to the increase in the number of courses on these platforms. It focused on answering the main research question RQ: *How to identify and reduce knowledge gaps in the MOOCs ecosystems?* Overall, we proposed a new content-based recommendation system applied to the scenario of MOOCs - FReME.

6.1. Contributions

Our work contributes to a further exploration of MOOCs ecosystems, which derives from the software ecosystem perspective, resulting in providers and users becoming contributors to the learning processes and enabling us to integrate and extend data from multiple MOOCs providers in the recommendation. This extensibility, along with the chosen methods that contribute to scalability, allows other developers to contribute to the conceptual model, adding new providers or extending the recommendation to any other ecosystem actor.

Moreover, our work contributes to detecting similarity between these multiple MOOCs ecosystems, automatically defining documents labels, recommending parts of courses (i.e., courses' modules) in multiple providers, and creating a new method to extract API data from these providers. It is also worth mentioning that this fragmented recommendation in multiple providers contributes to delivering packages of personalized modules to users according to their knowledge gap and to the customization level in terms of recommendation, increasing the possibilities of the recommended modules.

Based on experimental results involving real-world data in three experiments (and when returning to our RQ) we can indicate that our modified NMF with the topic labeling presents coherent results and that the recommendations are accurate, useful, reliable, and provide new content for students. These results state that our proposal contributes to assist students in the identification and reduction of knowledge gaps. This support involves the acquisition of new and relevant knowledge, which impacts the improvement of students' experiences in the platforms and it can then reduce dropouts rates in this learning modality. Also, we argue that our work contributes to the process of structuring and boosting the social and economic dynamics of a city, supporting the knowledge demands identification.

6.2. Limitations

Meanwhile, there are some limitations to this work. When executing the third experiment, the 50% in the hypotheses and the top-6 recommendations are empirical choices based on experiments made on other works described in domain literature. The verification of an ideal number of items to be recommended for each user could have been added. Moreover, collecting feedback only from the resulted modules from FReME is also a limitation. Modules from other MOOCs recommendation systems could have been added to avoid bias in the evaluation. Another limitation is that courses already enrolled are not recommended, even that the student was evaded (which does not necessarily mean disinterest in the course). For instance, the cause of dropout may be disinterest in the platform or disinterest in the teacher's methodology. In this case, FReME could recommend a module on the same subject from another provider, which was not an objective in this work.

6.3. Future work

Directions for future work include an evaluation that contemplates comparing our automatic labels with manual labels defined by humans. Furthermore, it can be conducted a broader identification of the knowledge gap, creating parameters to identify the learning degree and deeply investigate a given topic by a user. It is also possible to define problems a user can already master through big data approaches for cognitive science, e.g., modeling knowledge states. These strategies can be necessary since the fact of a user complete a module does not necessarily indicate that knowledge has been acquired.

Although some MOOC providers list the courses' prerequisites, Goopio and Cheung (2021) - when investigating the MOOC dropout phenomenon - highlight that one of the reasons for the increase in MOOCs dropout rates is the absence of prerequisite requirements, as occurs in some traditional classroom courses that require students to complete an admission process. This openness with no limitation requiring prerequisite knowledge is still one of the challenges faced in the MOOCs scenario. Therefore, a prerequisite structure for the modules - including building a pedagogical order among the top recommended modules in order to "create" a new course (with a beginning, middle, and end) on subjects of interest to the user - can be pointed out as a next step in our work. There is also the possibility of reinforcing other aspects of recommendation that were not explored in this work, such as the FReME robustness, i.e., its ability to operate under stressful conditions, typically in the presence of fake information (e.g., through a concerted Sybil attack). Finally, there is an opportunity to add datasets from other providers on the experiments, including providers in other languages, by adapting the implementations made to include the language of the courses.

CRediT authorship contribution statement

Rodrigo Campos: Methodology, Software, Investigation, Resources, Writing – original draft, Visualization. **Rodrigo Pereira dos Santos:** Validation, Writing – review & editing, Supervision. **Jonice Oliveira:** Conceptualization, Validation, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank IFRJ, Brazil for supporting this research. We also thank CAPES, Brazil, CNPq, Brazil, and FAPERJ, Brazil (Proc. 211.583/2019) for their financial support, and Oracle Cloud credits and related resources provided by the Oracle for Research program (award number CPQ-2160239).

References

Aggarwal, C. C. (2016). *Recommender systems*. Cham, Switzerland: Springer International Publishing, http://dx.doi.org/10.1002/9781119054252, arXiv:1202.1112.

Apaza, R. G., Cervantes, E. V., Quispe, L. C., & Luna, J. O. (2014). Online courses recommendation based on LDA. In J. A. Lossio-Ventura, & H. A. Salas (Eds.), *CEUR Workshop Proceedings*: *vol. 1318*, *Proceedings of the 1st symposium on information management and big data - SIMBig 2014, Cusco, Peru* (pp. 42–48). CEUR-WS.org.

Aryal, S., Porawagama, A. S., Hasith, M. G. S., Thoradeniya, S. C., Kodagoda, N., & Suriyawansa, K. (2019). MoocRec: Learning styles-oriented MOOC recommender and search engine. In *2019 IEEE global engineering education conference* EDUCON, (pp. 1167–1172). Dubai, United Arab Emirates: IEEE, http://dx.doi.org/10.1109/EDUCON.2019.8725079.

Barros, H., Magalhães, J., Marinho, T., Silva, M., Miranda, M., & Costa, E. (2018). Linked educational online courses to provide personalized learning. In F. Castro, S. Miranda-Jiménez, & M. González-Mendoza (Eds.), *Lecture notes in computer science*: *vol. 10633*, *Advances in computational intelligence* (pp. 198–207). Cham, Switzerland: Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-02840-4_16, http://link.springer.com/10.1007/978-3-030-02840-4_16.

Basili, V. R. (1992). *Software modeling and measurement: The goal/question/metric paradigm*: *Technical Report*, University of Maryland.

Bhatt, C., Cooper, M., & Zhao, J. (2018). SeqSense: Video recommendation using topic sequence mining. In K. Schoeffmann, T. H. Chalidabhongse, C. W. Ngo, S. Aramvith, N. E. O'Connor, Y.-S. Ho, M. Gabbouj, & A. Elgammal (Eds.), *MultiMedia modeling* (pp. 252–263). Cham, Switzerland: Springer International Publishing.

Biagiotti, B. A. (2016). *Assessment of the quality of information from massive courses: A case study of TELELAB* (Master's Thesis in Engineering and Knowledge Management), (p. 140). Federal University of Santa Catarina.

Bond, P., & Leibowitz, F. (2013). MOOCS and serials. *Serials Review*, *39*(4), 258–260. http://dx.doi.org/10.1016/j.serrev.2013.10.007, http://linkinghub.elsevier.com/retrieve/pii/S0098791313001342.

Bosch, J. (2009). From software product lines to software ecosystems. In *SPLC '09*, *Proceedings of the 13th international software product line conference* (pp. 111–119). San Francisco, California, USA: Carnegie Mellon University, http://dx.doi.org/10.5555/1753235.1753251.

Campos, R. (2019). *Recommendation system for knowledge acquisition in MOOCs ecosystems* (Master's thesis PPGI/UFRJ), Federal University of Rio de Janeiro Brazil.

Campos, R., Santos, R. P., & Oliveira, J. (2018). Web-based recommendation system architecture for knowledge reuse in MOOCs ecosystems. In *IEEE 19th International conference on information reuse and integration* (pp. 193–200). Salt Lake City, Utah, USA: IEEE, http://dx.doi.org/10.1109/IRI.2018.00036.

Campos, R., Santos, R. P., & Oliveira, J. (2020a). A recommendation system based on knowledge gap identification in moocs ecosystems. In *SBSI'20, XVI Brazilian symposium on information systems* (pp. 1–8). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3411564.3411572.

Campos, R., Santos, R. P., & Oliveira, J. (2020b). Recommendation system for knowledge acquisition in MOOCs ecosystems. In *Extended proceedings of the XVI Brazilian symposium on information systems* (pp. 93–108). Porto Alegre, RS, Brazil: SBC, http://dx.doi.org/10.5753/sbsi.2020.13132.

Campos, R., Santos, R. P., & Oliveira, J. (2020c). Supporting students through a recommendation system for knowledge acquisition in MOOCs ecosystems. In *Proceedings of the workshops of the Brazilian congress on informatics in education* (pp. 62–71). Porto Alegre, RS, Brazil: SBC, http://dx.doi.org/10.5753/cbie.wcbie.2020.62.

Campos, R., Santos, R. P., & Oliveira, J. (2020d). A recommendation system enhanced by topic modeling for knowledge reuse in MOOCs ecosystems. In L. Rubin, Stuart H. and Bouzar-Benlabiod (Ed.), *Reuse in intelligent systems* (pp. 116–142). Boca Raton, Florida, EUA: CRC Press.

Campos, R., Santos, R. P., & Oliveira, J. (2021). Techniques, challenges and future directions of recommendation systems for MOOCs. *IEEE Transactions on Learning Technologies*, submitted for publication.

Corbellini, A., Mateos, C., Zunino, A., Godoy, D., & Schiaffino, S. (2017). Persisting big-data: The NoSQL landscape. *Information Systems*, *63*, 1–23.

Do, M.-P. T., Nguyen, D. V., & Nguyen, L. (2010). Model-based approach for collaborative filtering. In *Proceedings of the 6th international conference on information technology for education* (August 2010), (pp. 217–225). URL https://goo.gl/BHu7ge.

Goopio, J., & Cheung, C. (2021). The MOOC dropout phenomenon and retention strategies. *Journal of Teaching in Travel & Tourism*, *21*(2), 177–197. http://dx.doi.org/10.1080/15313220.2020.1809050.

Gope, J., & Jain, S. K. (2017). A learning styles based recommender system prototype for edx courses. In *2017 International conference on smart technologies for smart nation* (pp. 414–419). Bangalore, India: IEEE, http://dx.doi.org/10.1109/SmartTechCon.2017.8358407, URL https://ieeexplore.ieee.org/document/8358407/.

Greene, D., O'Callaghan, D., & Cunningham, P. (2014). How many topics? Stability analysis for topic models. In T. Calders, F. Esposito, E. Hüllermeier, & R. Meo (Eds.), *Machine learning and knowledge discovery in databases* (pp. 498–513). Berlin, Heidelberg: Springer Berlin Heidelberg.

Jansen, S., & Cusumano, M. (2012). Defining software ecosystems: A survey of software platforms and business network governance. In *Proceedings of the international workshop on software ecosystems 2012* (pp. 41–58). Cambridge, MA, USA: http://dx.doi.org/10.4337/9781781955628.00008.

Jansen, S., Finkelstein, A., & Brinkkemper, S. (2009). A sense of community: A research agenda for software ecosystems. In *2009 31st International conference on software engineering - companion volume* (pp. 187–190). http://dx.doi.org/10.1109/ICSE-COMPANION.2009.5070978.

Jing, X., & Tang, J. (2017). Guess you like: Course recommendation in MOOCs. In *Proceedings of the international conference on web intelligence* (pp. 783–789). New York, NY, USA: ACM, http://dx.doi.org/10.1145/3106426.3106478, https://dl.acm.org/doi/10.1145/3106426.3106478.

Kim, P. (2014). *Massive Open Online Courses: The MOOC Revolution*. Routledge.

Kuang, D., Choo, J., & Park, H. (2015). Nonnegative matrix factorization for interactive topic modeling and document clustering. In *Partitional clustering algorithms* (pp. 215–243). Cham: Springer, http://dx.doi.org/10.1007/978-3-319-09259-1_7.

Li, D., Liu, H., Zhang, Z., Lin, K., Fang, S., Li, Z., & Xiong, N. N. (2021). CARM: Confidence-aware recommender model via review representation learning and historical rating behavior in the online platforms. *Neurocomputing*, *455*, 283–296. http://dx.doi.org/10.1016/j.neucom.2021.03.122, URL https://www.sciencedirect.com/science/article/pii/S0925231221005142.

Magatti, D., Calegari, S., Ciucci, D., & Stella, F. (2009). Automatic labeling of topics. In *2009 Ninth international conference on intelligent systems design and applications* (pp. 1227–1232). Pisa, Italy: IEEE, http://dx.doi.org/10.1109/ISDA.2009.165.

Nolasco, D. (2016). *Automatic research areas identification in S&T* (Master's thesis PPGI/UFRJ), (p. 195). Federal University of Rio de Janeiro, Brazil, In Portuguese.

Nolasco, D., & Oliveira, J. (2016a). Detecting knowledge innovation through automatic topic labeling on scholar data. In *2016 49th Hawaii international conference on system sciences* (pp. 358–367). Koloa, HI, USA: IEEE, http://dx.doi.org/10.1109/HICSS.2016.51, URL http://ieeexplore.ieee.org/document/7427226/.

Nolasco, D., & Oliveira, J. (2016b). Topic modeling and label creation: Identifying themes in semi and unstructured data. In *Brazilian symposium on database - topics in data and information management* (31ed), (pp. 87–112). Eduardo Ogasawara, Vaninha Vieira. (Org.): SBC.

O'Callaghan, D., Greene, D., Carthy, J., & Cunningham, P. (2015). An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, *42*(13), 5645–5657. http://dx.doi.org/10.1016/j.eswa.2015.02.055.

Peters, M. A. (2008). The history and emergent paradigm of open education. In *Open education and education for openness* (pp. 1–15). Leiden, The Netherlands: Brill | Sense, http://dx.doi.org/10.1163/9789087906818_002.

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2015). *Recommender systems handbook*. Springer.

Shah, D. (2020a). By the numbers: MOOCs during the pandemic. https://www.classcentral.com/report/mooc-stats-pandemic/, (Accessed 26 December 2020).

Shah, D. (2020b). By the numbers: MOOCs in 2020. https://www.classcentral.com/report/mooc-stats-2020/ (Accessed 26 December 2020).

Shen, X., Yi, B., Liu, H., Zhang, W., Zhang, Z., Liu, S., & Xiong, N. (2019). Deep variational matrix factorization with knowledge embedding for recommendation system. *IEEE Transactions on Knowledge and Data Engineering*, *33*(5), 1. http://dx.doi.org/10.1109/TKDE.2019.2952849, URL https://ieeexplore.ieee.org/document/8896070/.

Shu, J., Shen, X., Liu, H., Yi, B., & Zhang, Z. (2018). A content-based recommendation algorithm for learning resources. *Multimedia Systems*, *24*(2), 163–173. http://dx.doi.org/10.1007/s00530-017-0539-8, URL http://link.springer.com/10.1007/s00530-017-0539-8.

Symeonidis, P., & Malakoudis, D. (2019). Multi-modal matrix factorization with side information for recommending massive open online courses. *Expert Systems with Applications*, *118*(15), 261–271. http://dx.doi.org/10.1016/j.eswa.2018.09.053, https://linkinghub.elsevier.com/retrieve/pii/S0957417418306316.

Vieira, V., Tedesco, P., Salgado, A. C., & Brézillon, P. (2007). Investigating the specifics of contextual elements management: The cemantika approach. In B. Kokinov, D. C. Richardson, T. R. Roth-Berghofer, & L. Vieu (Eds.), *Modeling and using context* (pp. 493–506). Berlin, Heidelberg: Springer Berlin Heidelberg.

World Economic Forum (2020). The future of jobs report 2020. https://www.weforum.org/reports/the-future-of-jobs-report-2020. (Accessed 17 January 2021).

Yeager, C., Hurley-Dasgupta, B., & A. Bliss, C. (2013). CMOOCS and global learning: An authentic alternative. *Journal of Asynchronous Learning Networks*, *17*(2), 133.

Yi, B., Shen, X., Liu, H., Zhang, Z., Zhang, W., Liu, S., & Xiong, N. (2019). Deep matrix factorization with implicit feedback embedding for recommendation system. *IEEE Transactions on Industrial Informatics*, *15*(8), 4591–4601. http://dx.doi.org/10.1109/TII.2019.2893714, URL https://ieeexplore.ieee.org/document/8616805/.

Zhang, J., Gao, M., & Zhang, J. (2021). The learning behaviours of dropouts in MOOCs: A collective attention network perspective. *Computers and Education*, *167*(19), Article 104189. http://dx.doi.org/10.1016/j.compedu.2021.104189.

Zhao, Q., Zhang, Y., & Chen, J. (2016). An improved ant colony optimization algorithm for recommendation of micro-learning path. In *2016 IEEE International conference on computer and information technology* (pp. 190–196). Nadi, Fiji: IEEE, http://dx.doi.org/10.1109/CIT.2016.47.