A Recommendation System based on Knowledge Gap Identification in MOOCs Ecosystems

Rodrigo Campos
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
rodrigo.campos@ufrj.br

Rodrigo Pereira dos Santos
Federal University of the State of Rio de Janeiro
Rio de Janeiro, Brazil
rps@uniriotec.br

Jonice Oliveira
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
jonice@dcc.ufrj.br

ABSTRACT

The consolidation of recommendation systems in a big data era brings opportunities in different scenarios to customize methods that recommend data. In the scenarios of the Massive Open Online Courses (MOOCs) ecosystems, these recommenders mainly support students in choosing the best courses from the platforms. However, the expansion of course platforms and the scarcity of student data increases the difficulty in finding courses, or even part of courses, that fill a given knowledge gap. In this paper, we propose a recommendation system to support students in finding the best modules or courses in these ecosystems. First, topic modeling techniques were implemented with Non-negative Matrix Factorization (NMF) to find similarities between multiple MOOCs providers. Then, a content-based recommendation provides recommendations to a user interested in acquiring new knowledge, based on a history extraction on those platforms. We evaluate our approach through an experiment with real data collected in multiple MOOCs providers. In addition, by comparing the NMF approach with a baseline Latent Dirichlet Allocation (LDA) technique, we verify the model effectiveness and show that our system is useful to this context.

CCS CONCEPTS

• **Information systems** → *Information retrieval*; **Recommender systems**.

KEYWORDS

MOOCs ecosystems, recommendation system, topic modeling, non-negative matrix factorization

ACM Reference Format:

Rodrigo Campos, Rodrigo Pereira dos Santos, and Jonice Oliveira. 2020. A Recommendation System based on Knowledge Gap Identification in MOOCs Ecosystems. In *XVI Brazilian Symposium on Information Systems (SBSI'20), November 03–06, 2020, São Bernardo do Campo, Brazil.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3411564.3411572

SBSI'20, November 03–06, 2020, São Bernardo do Campo, Brazil
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8873-3/20/11…$15.00
https://doi.org/10.1145/3411564.3411572

1 INTRODUCTION

Online distance education (also called online distance learning) contemplates a combination of cost-effectiveness and flexibility to both students and teachers [3]. A specific type of course that is included in this movement is the Massive Open Online Courses (MOOCs) that emerged in 2008. It differs from traditional distance courses because of the possibility of an unlimited number of participants, besides being carried out exclusively through the internet, and giving students the freedom to choose the topics they want to study with time flexibility. MOOCs ecosystems are formed by MOOCs platforms, their students, higher education institutions, teachers, and other actors with different roles and interactions in the ecosystems, contributing to the software evolution, knowledge sharing, and other benefits [7].

However, with the growth of MOOCs ecosystems, there is a concentration of new courses emerging from different providers, mostly created and maintained by higher education institutions. This brings difficulties for students in the identification of what course is the best one to meet a particular demand for knowledge. Therefore, students have to identify which provider is best suited to their knowledge gap, considering their preferences.

One way to provide a solution to this problem is by using the recommendation technique. A recommendation system aims to automate the process of effectively identifying the most interesting content for a user, based on the opinions, choices, and data of a community of users [11]. These systems can apply collaborative filtering (CF) methods, content-based recommendation (CBR) methods, or a hybrid approach, merging the two previous types.

Based on the different possibilities of applying recommendation systems to support MOOCs, some authors suggest the use of topic modeling to find topics among data, mining and creating a distribution. The recommendations in these cases explore diverse information of students, such as interactions in forums [19], historical access behaviors [13], and watched videos [6].

Despite advances in this recommendation scenario, it is still necessary to invest in recommendations of courses or consider recommending part of these courses. Moreover, it is necessary to deal with data privacy issues, where only the general data of the courses are opened, but the student behavior information must be authorized to be accessed by each student (target user). Therefore, in this work, our goal is to investigate how to use only data of a target user in the recommendation process without limiting the content recommendation, i.e., to reduce the restriction of recommendations caused by the scarcity of items, consequently reducing the user-item matrix sparsity in MOOCs.

To achieve this goal, this paper proposes a recommendation system for MOOCs ecosystems that use multiple providers, which can decrease the scarcity of the item, and topic modeling algorithms in a CBR. The topic modeling technique has the role of clustering items to find the same or related items between the multiple databases, using a Non-negative Matrix Factorization (NMF). By iteratively training, we create a user profile and specific recommendations of parts of courses, such as modules that support the students within such an ecosystem.

2 BACKGROUND

The use of different techniques to support recommendation allowed users to receive the suggestion of an item in an increasingly efficient and personalized way. One of the possible techniques is called topic modeling, also applied in our work. Topic modeling is widely used in recommendation systems for data mining, getting the probability of topic distribution. As such, it is possible to use it to calculate the similarity between recommendation's users and items, and then provide recommendations [17].

2.1 Recommendation System Approaches

The works that propose topic modeling applied to recommendation systems, for the most part, present CF approaches or CBR approaches. Given item *i* and user *u*, the CF has as a principle the fact that if a user *u* evaluated items similar to another user *u'*, there is a great chance of the next evaluation of a user *u* for a new item to be similar to that of a user *u'*. Therefore, if other users evaluated two items in a similar way, there is a tendency for user *u* to evaluate them in the same way [10].

The CBR consider descriptive attributes of the items, the so-called content; hence, the content-based name. These methods are used when there is no user information, i.e., the ratings of other users are not known, as it is known when applying CF. Thus, from a descriptor of item *i*, it is possible to find other items already evaluated with similar descriptors and consider the level of similarity to recommend or not recommend the item *i* [2].

Although CBR is advantageous for new items since they have little or no classification, some problems are encountered, such as a limitation of recommended items. This can happen because CBR is not based on ratings of other users (only the target user). Thus, if that user has never classified other items with the same set of words as item *i*, the item *i* is rarely recommended [2]. This shortage of recommended content is called cold start, meaning that there is not enough rating for the recommendation process.

2.2 Topic Modeling Techniques

The most commonly used topic modeling processes currently work as follows: from a collection of documents D = $\{d_1, d_2, \ldots, d_{|c|}\}$ and their respective fixed vocabulary of words V = $\{w_1, w_2, \ldots, w_{|v|}\}$, they distribute all these words (represented by terms *w*) into groups (represented by topics ϑ) with their probability $p(w|\vartheta)$, which gets higher each time that term is more related to a topic. Then, through the probability distribution $p(\vartheta|d)$ technique, they verify the probability of each document being linked to each topic. Thus, an association between topics and documents is created [18].

The Latent Dirichlet Allocation (LDA) is described by Nolasco and Oliveira [15] as 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.

Another technique is the NMF, which has as input a matrix of documents and terms and allows the generation of two approximate matrices for the latent structure in data [14]. Considering a corpus of documents n and the unique terms m of that corpus, the matrix of documents-terms $A \in \mathbb{R}^{m \times n}$ is processed by NMF and generates a reduced rank-k approximation. It is represented by $A \approx WH$, where the columns $W \in \mathbb{R}^{m \times k}$ are the topics with the weights relative to each term *m*, and the matrix $H \in \mathbb{R}^{k \times n}$ is composed of the documents associated with each topic *k*. Therefore, A is the product of these two non-negative factors W and H.

2.3 Related Work

Song et al. [19] present learning assistance to identify the user profile through the interactions that a user has in forums. To do so, all messages present in a thread are stored in a document. The SVM (Support Vector Machine) classification method is used to identify the categories of these threads. The experiments performed by Song et al. [19] only consider Coursera, showing that the proposal performs well for such a provider. Although it makes the forums categorized, the proposal does not go deeper into the CF process for users considering the techniques. It indicates such functionality as a future work whose purpose would be to investigate how to contemplate the individual learning data.

Jing and Tang [13] propose an algorithm framework called "Guess You Like" and integrate it as a block to the provider XuetangX. It aims to provide customized course recommendations for users. To do so, it applies a CF merged with a content-aware algorithm that extracts students' historical access behaviors. Moreover, it has as a differential the use of demographic data and courses prerequisites defined by the provider to compose the user profile. When compared to other CF approaches, the method demonstrated performance better than them. In spite of the positive results, the application of a CF approach was only possible, since the provider enables user data extraction (the experiments had data of 114,303 users), which may not be open in other providers due to data privacy terms. When pointing out future work, it is ratified the need for improving recommendations in MOOCs not only based on full courses, but also on "different kinds of content" [13].

Bhatt et al. [6] propose a recommendation of videos in multiple providers, called SeqSense, with the intention to benefit students who seek a more punctual knowledge. A CBR is applied, and it is used the transcript of the platform videos as input. The application of topic modeling with a sequence-based recommendation allows the method to find a sequence relationship between videos. For the modeling of topics, the LDA technique (one of the most used techniques for topic modeling) with a fixed definition of the number of topics Z = 30 was chosen.

In our solution, we chose a recommendation approach different from the "Guess You Like" algorithm [13] or the one presented by Song et al. [19] because of these work address recommendations in scenarios where the interaction between users are available;

therefore, they applied the CF approach. Similar to our proposal, SeqSense [6] fills a demand identified by Jing and Tang [13] to recommend not only complete courses but also the down level of contents. While SeqSense [6] recommends videos, our work aims to recommend modules (parts of the courses in MOOC). SeqSense [6] also has in common with our work the recommendation of multiple providers, allowing a greater variety of recommended items and allowing items to be more personalized and less sparse in the recommendation.

In relation to topic modeling, our solution adopts a method for automatically identifying the best value of topics k for a model, which guarantees that the modeling has the best coherence of topics for a given vocabulary. Our proposal differentiates itself from other MOOCs recommenders by applying matrix factorization topic modeling through NMF. Another aspect is the recommendation method itself which is a CBR but also making use of the modeled NMF topics to recommend.

3 PROPOSED RECOMMENDATION SYSTEM APPROACH

The proposed recommendation system demands to integrate data from several MOOCs platforms, so there is no need to plan, develop, and maintain an individual recommendation system for each platform. According to Abdalla et al. [1], the software ecosystem (SECO) perspective is useful in this demand for information systems as it provides the reuse of recommendation system elements. As such, MOOCs ecosystems (aggregating and inheriting characteristics from SECO) can collect and process data from these multiple platforms. MOOCs ecosystems also inherit the openness for external developers to contribute to the functioning of the ecosystem. Therefore, our proposed contribution interacting directly with the MOOCs providers' APIs become possible.

The proposed algorithm for recommending modules in MOOCs ecosystems is based on topic modeling. Therefore, the algorithm is divided into two main steps. A first step described in Section 3.1, which consists of obtaining data, modeling topics of the items, and creating multiple user profiles. And the second step presented in Section 3.2 that consists of the application of the CBR method.

3.1 Modeling Topics in MOOCs with NMF

As the MOOCs recommendation is merged with multiple databases, the modeling assists in identifying similar content through topic clustering and allows that modules from distinct MOOCs platforms be grouped according to the content in common. There are two topic models: item modeling, which refers to the content that can be recommended, i.e., all the modules in the selected MOOCs providers; and the user modeling, which corresponds to the modules of the platform that the interested student (target user) has attended (or is still attending). In some providers, such as in Khan Academy, the available information from the target student is just the watched videos. As a consequence, it is necessary a data treatment to standardize data from providers.

Our recommendation system recognizes data from any provider that has data extraction permission via RESTful API with JSON outputs. Otherwise, it is possible to extract the data using other techniques (such as crawlers) and then convert them to JSON, allowing the recommendation's input data to be standardized. Therefore, for each provider, it is created a variable "json_url" that stores the URL of the API, while the variable "j" makes an HTTP request passing the value of “json_url” as the parameter. Once the connection is established, the code loads the json content of "j" into the "content" variable.

To complete the item modeling, the data extraction starts in with the topic tree and thus traverses the children fields. Each provider has a particularity of data organization in the API. However, in this step, it is necessary to consider that each document of this collection is equivalent to an item able to be recommended to a user (modules or complete courses for providers that do not allow learning partitioning).

To describe the content of the document, the title and description of the module are extracted, as well as general course information, module contents, and videos, exercise, and/or articles that compose the module. These extractions are exemplified in Figure 1, which demonstrates the process in the Khan Academy provider's topic tree. At this point, the more content about the module exists, the more accurate the recommendation may be.

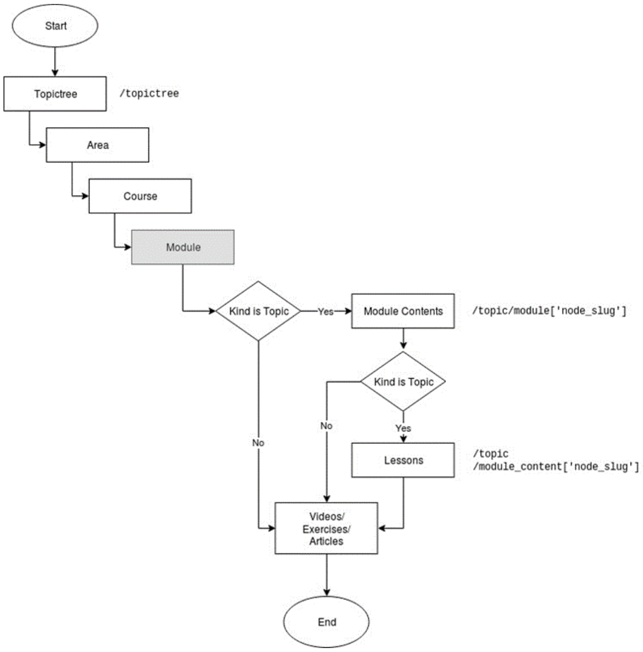

Figure 1: Flowchart of selected data types in Khan Academy's topic tree

With the necessary data properly extracted for each document, it is necessary to store it in an integrated way. The storage of these documents in our recommendation system is based on the use of a document-oriented database since the recommendation system can be extended. To do so, criteria such as scalability, consistency, and availability must be contemplated. Document-oriented databases are recommended for these cases, allowing the number of fields to be expanded or new features to be added [9].

After data storage, the algorithm can model the topics following the steps shown in Algorithm 1, using the NMF technique.

Algorithm 1: Automatic NMF topic modeling integrated with providers

```
   Input: JSON data dt from providers where each row
          represents a module and list of stopwords sw
   Output: W (document-topic matrix) and H (topic-term
           matrix)
1  list = TransformDataIntoUTF8List(dt);
2  tokenizer = LemmaTokenizer( );
3  Vectorizer = TfidfVectorizer(sw, tokenizer);
4  A = CreateDocumentTermMatrix;
5  vocabulary = Vectorizer(A);
6  kmin, kmax = SetValues( ) ; /* integer is required */
7  for $k = kmin$ to $kmax$ do
8      CalculateCoherence(k);
9      coherences = [k];
10 end
11 best_k = GetBestK(coherences);
12 W = GenerateDocumentTopicMatrix(A, best_k);
13 H = GenerateTopicTermMatrix(A, best_k);
```

The procedures executed to find k (at line 8 of Algorithm 1) are based on the approach used by Greene et al. [12] and reinforced in the comparisons made by Campos et al. [8]. The NMF is adopted since it provides a high level of interpretability in understanding the user-item interactions [2]. These procedures are important to satisfy a need that the traditional methods of topic modeling have of an input of k defined by the developer, which does not leave the model so dynamic and can cause a selection of a k with a low topic coherence. As such, tests to find k are based on applying the modeling method to different values of k (given a minimum and maximum k, called respectively *kmin* and *kmax* at line 7 of Algorithm 1) until a k that reproduces a topic coherence value higher than the others.

It can also be analyzed in Algorithm 1 that when receiving the *dt* data, the algorithm processes a series of data treatments, as well as: transforms this data into a grouped *list* in UTF8 by clearing special characters (at line 1); applies the Bag of Words Model, where each document is represented by a vector of unique terms (at lines 2 and 3); creates a Document-Term Matrix named as A (at line 4) by assigning a weight to each term using the instantiated *Vectorizer*, and extract the *vocabulary* of this matrix. The variables W and H are the resulting factors of the NMF applied to matrix A.

The second topic modeling corresponds to the creation of a user profile, i.e., the information about the student. As the objective of this work is to be able to gather data from multiple bases to enhance the user profile, we verify how these data can be integrated into the document's structure. In several cases, the information from enrolled modules, courses, or watched videos by the target user is made available, through the implementation of an authorization module. After access granted, it is possible to navigate through the target user flowchart that contains the content enrolled by this user. Each user document is represented by the title and description of the enrolled content depending on the selected provider.

Unlike item extraction, user data is not allocated in the recommendation system database. The information is grouped at runtime, i.e., at the time of the recommendation request depending on the authorizer's response. With these documents, it is possible to implement the same outline topic modeling logic as applied for the item in Algorithm 1, from turning that data into a list until the generation of the document-item matrix. However, the variables are named as *A_user*, *vocabulary_user*, *k_user*, and so on. The resultant matrices are *W_user* and *H_user*.

3.2 Content-Based Recommendation Method

The second step of the recommendation process involves the use of the CBR by cross-referencing the item and user topics combined with NMF in the documents-terms matrix. Considering that user-layer topics represent what data the student has already accessed, it is necessary to identify to which item topic a profile is more related. To do so, the method checks what is the most similar item topic based on topic similarity cosine (TS-COS) in each user's topic. TS-COS indicates that "the similarity between two topics is computed as the average pairwise cosine similarity between their top-10 most probable words" [4]. Once the comparison methods were implemented, it is possible to obtain the result of Figure 2a.

At the end of calculating all similarity relationships between user and item topics, it is possible to identify which topic item is most frequent in these relationships. As this selected topic indicates the one that the student has the biggest similarity, it is used for the recommendation effectively. Through matrix W (documents-topics), it is possible to verify which documents are best ranked for the selected target topic. The link between documents and model topics in matrix W is calculated by associative probability and it is represented by an index. In order to be able to select the indices of the most related documents in this model, it is necessary to reverse sort these values by selecting the top-ranked indexes. From such indexes, it is possible to locate the respective documents. However, these documents do not need to be represented by all existing fields in the database. It is possible to create a text clipping that can indicate the key information to represent each document. A possible example would be the module title. However, since there may be repeated titles in different providers, it was chosen to merge some information: module title, module URL, provider identifier, linked exercise or video URLs (exclusively for Khan Academy), and course URL. Six documents are extracted, as shown in Figure 2b.

A set s of 6 documents related to the topic is defined. However, it is necessary to identify what is the knowledge gap of the student, i.e., which modules linked to this topic the student has not yet had contact. With all extracted (each one representing modules or courses), the system verifies what documents of the user layer are in s. If the user layer contains modules or courses, it compares if documents are equal. If the user layer contains videos, the video identifiers contained in each module of s are extracted. Next, if a user watched video is in s, this document is discarded, i.e., it is understood that the student has already started this module, so it does not appear as a knowledge gap. In this case, the documents that are out of the watched videos are selected for recommendation. In Figure 2c, it is possible to observe that a topic with a set s of 6 modules has been extracted from W, for instance. However, a user

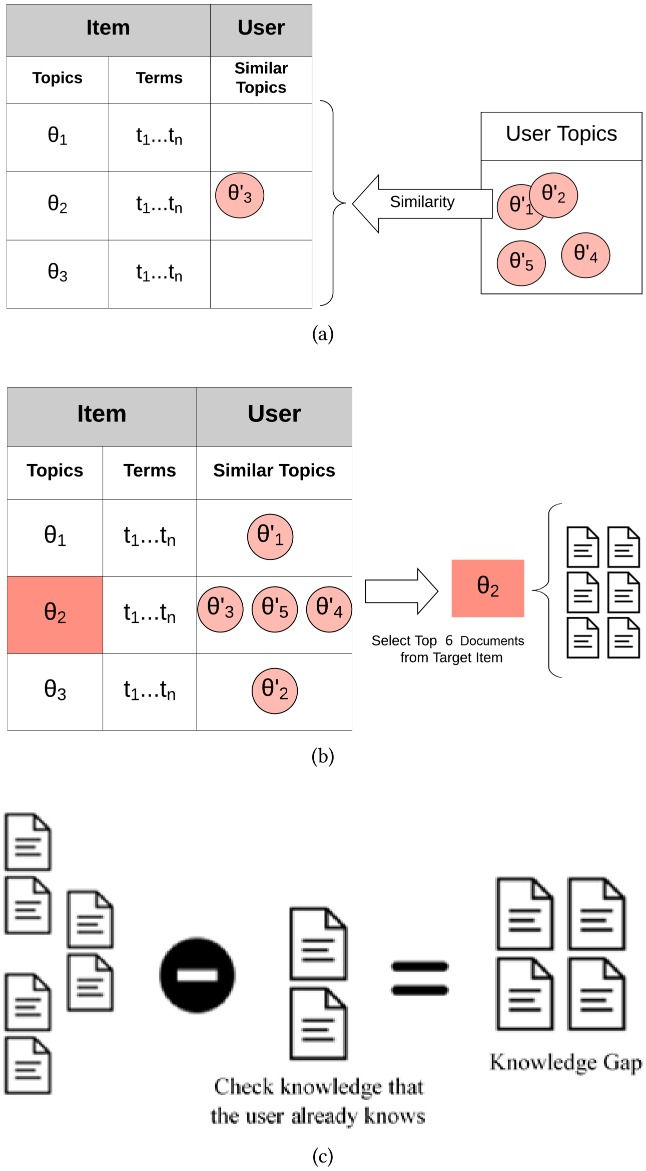

Figure 2: (a) Comparison methods between item and user, (b) documents extraction from target item, and (c) selecting the knowledge gap from model

has already watched at least one video in two of these modules. Thus, the recommendation gap retrieves only 4 modules as a result.

4 EVALUATION

An experiment was conducted with the extraction from multiple providers to validate if the proposed NMF topic modeling and the recommendation system as a whole are useful. Next, the same database was tested in a topic modeling applying the baseline LDA method, and then results were compared to verify the effectiveness of our method. Figure 3 details how the methodology is planned for this experiment.

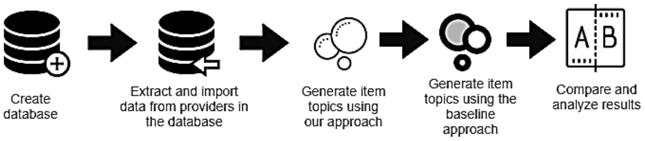
Create database
Extract and import data from providers in the database
Generate item topics using our approach
Generate item topics using the baseline approach
Compare and analyze results
A
B

Figure 3: Methodology of the proposed evaluation

4.1 Experiment

4.1.1 Datasets Information. The protocol for the evaluation of topic modeling in our method involves the entire provider's dataset. The providers chosen for this stage were Khan Academy, Udemy, and edX. The way providers make data provisioning is constantly changing. Until the implementation of this work, these providers have an open API that makes some platform data available via RESTful API. Information related to use can be found in the Khan API Wiki[1], in Udemy Documentation for developers[2], and in the edX Course Catalog API User Guide[3]. The API can be accessed in the browser at Khan full topictree[4], Udemy courses[5], or edX course catalog[6].

MongoDB (a free Document-oriented database) was chosen for data storage. The choice was due to the fact that it integrates data in JSON format, finding the BSON – binary that takes up less space and is faster. For using the service of the MongoDB, MongoDB's Database as a Service (DBaaS) was applied: MongoDB Atlas, free to use (through the M0 layer), exempting us in the planning or managing of the database's infrastructure. To better visualize and handle data, we opted for the User Interface called Mongo Compass.

This integrated data (a total of 63,124 modules) were used to fill the item and user layers with modules enrolled by a target user. Our method proposes that the automatic definition of the number of topics in NMF can be applied also to the domain of the MOOCs. The data input occurs through the API topic tree. However, only some specific fields are extracted. An example with real data of the Khan Academy's topic tree data and the module "Early Math", respectively in Figure 4 and Figure 5, shows the performance of this data collection process presented in Figure 1.

It is possible to observe in Figure 4 that the main page of the topic tree has only general information. Moreover, to access a course itself, it is necessary to go through two arrays. The first one accesses the area ("math", in the example given in position 0), and the second access the courses ("k-8th grades", in the example given in position 0). The number n of areas and courses is dynamic, according to the insertion of new contents in the providers. The course "k-8th grades" has a "slug" field. This field is responsible for storing the API address with specific course information.

4.1.2 Item Layer. From the modules properly extracted, with the appropriate fields selected and grouped, the steps to model the item topics are followed so that this data can be clustered according to the similarity between them. As these modules represent the

[1] https://github.com/Khan/khan-api/wiki/Khan-Academy-API
[2] https://www.udemy.com/developers/affiliate/
[3] https://course-catalog-api-guide.readthedocs.io
[4] http://www.khanacademy.org/api/v1/topictree
[5] https://www.udemy.com/api-2.0/courses
[6] https://api.edx.org/catalog/v1/catalogs/

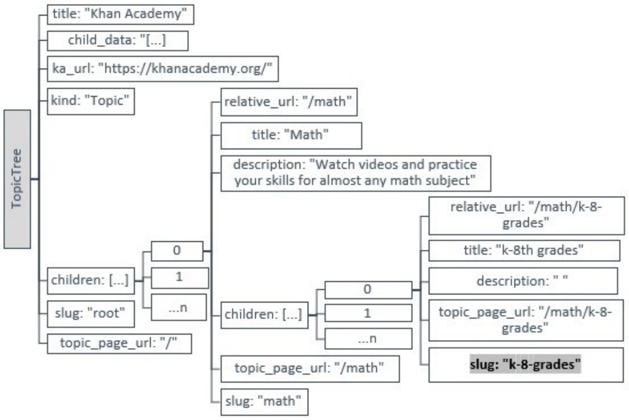

Figure 4: Extract of some Khan Academy API topic tree fields

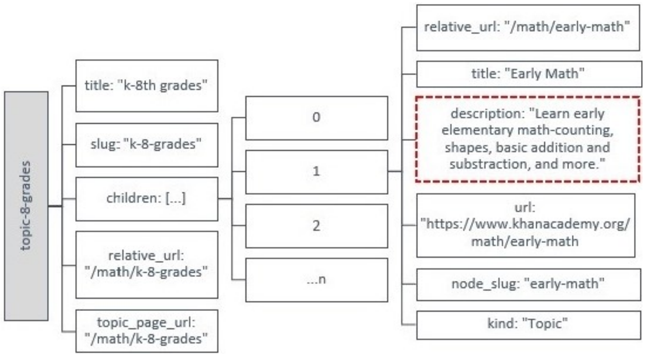
topic-8-grades
title: "k-8th grades"
slug: "k-8-grades"
children: [...]
relative_url: "/math/k-8-grades"
topic_page_url: "/math/k-8-grades"
0
1
2
...n
relative_url: "/math/early-math"
title: "Early Math"
description: "Learn early elementary math-counting, shapes, basic addition and substraction, and more."
url: "https://www.khanacademy.org/math/early-math
node_slug: "early-math"
kind: "Topic"

Figure 5: Extract of some Khan Academy API topic-8-grades fields

documents, it is possible to know the size of the vocabulary and consequently the document-item matrix created by the model with the library Scikit-learn. Therefore, the vocabulary has a total of 172,708 distinct terms and the matrix has a size of 63,124 x 172,708. For each line of the array, it is possible to know the weight that an item has in relation to the associated document, through the TfidfVectorizer that contemplate the Term Frequency (TF) and the Inverse Document Frequency (IDF), besides allowing to exclude words that are in the stop words list, such as the word "and".

For the stage of automatic choice of the number of topics k, it is necessary to inform a minimum and maximum k for the system test where NMF chooses the best topic coherence. In the implementation of NMF, the system uses the Scikit-learn library which only requires that the matrices H and W be initialized. The matrices are filled for each k value hypothesis. Next, the coherence is calculated. The value of k that presents the greatest coherence is selected to represent the final k of that document collection. In the first moment, for the collection of this experiment with these datasets, it is identified that the best value to be applied would be k=9.

With the result of the best value for k, it is possible to generate H and W. From H, it is possible to list the 20 terms that are at the top of each topic. Although the consistency of the topics has been the best, human analysis of the terms may point to some modeling flaws. It is possible to observe some problems that if solved could optimize the clustering and the ranking of the terms so that the top-20 of each topic is closer to reality. To address these issues, some steps have been taken:

- Inclusion of other stop words: words such as "khanacademy" and "dr" appear several times in the providers because they are part of a common vocabulary in the courses. However, they are not relevant words for the topic description. In this context, with the first results of the modeling, it is possible to identify these words and add them as an extension of the already imported stop words, i.e., words that are not counted in the model.
- Retrieving the title and description of the courses: as each document contains a module accompanied by information such as the course, often the title and description is repeated every time one of its modules is presented. With this, the weight of the words of these fields becomes very large by repetition. To prevent it, the courses' titles and descriptions are removed, keeping only the module information and below, as shown in Figure 1.
- Define the min_df: it is defined the value 5. Then, when building the vocabulary, the algorithm ignores terms that have a document frequency strictly lower than 5.

With these corrected steps, the vocabulary changes to 23,749, and consequently the matrix A changes to 63,124 x 23,749. After performing the procedures to calculate the best value for k, the new comparative identified a total of 8 topics, with a coherence of 0.3480. Figure 6 represents the values of k and the coherence of the respective topic considering each k.

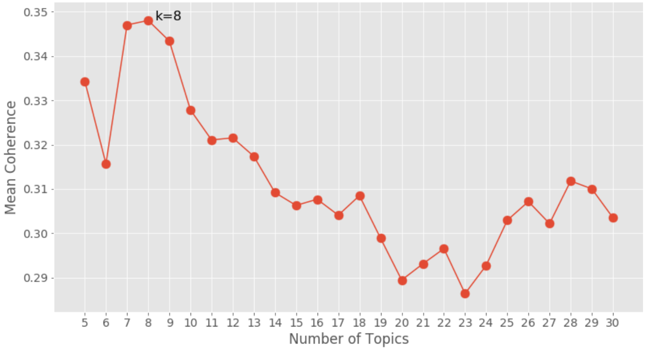

Figure 6: Topic coherence in each k-value for the item layer

4.1.3 User Layer. The API connection process accesses the user page (student) of the Khan Academy, Udemy, and edX according to the authorization algorithm. The protocol used to grant authorization is OAuth (versions 1 and 2). Titles and descriptions of videos watched by the student are grouped to be generated the document-item matrix. From this matrix, it is possible to collect vocabulary and train another Word2Vec model with the user's document corpus. The TC-W2V method is applied to extract the best value of k (topics) in a defined value range, which in the case of the user layer is represented by a minimum of 2 and a maximum of 5. After setting the minimum and the maximum number of topics for that user, the coherence is calculated for each k and the best value is k=2.

After generating *H_user* and *W_user*, it is possible to apply the method that compares each of the two topics with the other topics of the item layer. Thus, the target topic of a user is identified, extracting the documents that are in the knowledge gap of this user. The recommendations result for the target user is represented in Table 1.

This procedure of extracting the user's API until knowledge gap generation is performed every time a user requests for a recommendation. While the item's Word2Vec remains the same in all recommendations (if there is no inclusion or deletion of items), the user template is always regenerated.

4.2 Baseline Approach Comparison

For the results generated by our method to be validated, the topic coherence of this model is compared to the topic coherence in a topic model that applies the baseline LDA. This evaluation criterion is chosen because it is a more adequate one to evaluate models of topics originated with LDA as stated by O'Callaghan et al. [16] since it allows to capture of the semantic interpretability of exposed topics. To do so, it is used the terms that best describe each topic, the so-called terms descriptor.

The performance of our model is compared to the LDA baseline approach. For effective comparison, in addition to being used the same database in both, we also chose (manually) the number of 9 topics in the LDA model. The stop words are also the same and a Lemmatization process is also applied. As the metric to evaluate the coherence in our method is the TC-W2V, using as input a Word2Vec model built from the corpus, the same technique is adapted to evaluate the traditional LDA. Therefore, the top-20 terms of each topic were grouped, inserted in the coherence method and the result is 0.2209, i.e., less than the coherence of our method.

5 RESULTS

To evaluate the proposed method, we executed an experiment with real data to validate the relevance and usability of the recommendation system and its methods with real data. Such datasets allowed us to validate the topic modeling with NMF. We calculated the metric of topic coherence to compare this approach with a baseline method using LDA. Our results demonstrated that our NMF approach outperforms baseline approaches.

Our work builds an information system for the open world of MOOCs, covering the challenges highlighted in "4.3.1. Information Ecosystem Development" from the Grand Research Challenges in Information Systems in Brazil [5], since we consider scalability, flexibility, and adaptation. Scalability by defining a method to identify the best number of topics in our topic modeling without the need for manually modifying the model, also with an automatic data extraction method from MOOCs providers. The topic modeling process also helps our work to contemplates a crucial aspect in recommendation systems, according to Aggarwal [2]: to decrease the prediction time (i.e., the time that takes to the user receives the answer). When considering the flexibility aspect, we have that our recommendation system is not associated with a specific provider, but with the ecosystem (by addressing the MOOCs ecosystems approach) which allows us to amplify recommendations to other ecosystem actors. Moreover, we consider that our implementation features are also flexible, as our techniques can be implemented differently (e.g. adopting a different programming language), not being exclusive. Upon completion of future stages of implementation, the adaptation aspect will be contemplated by opening the code, allowing contributions by other developers in any recommendation system module.

The problem investigated in this work is also inserted in the challenges highlighted in "4.3.2. Open and Collaborative Processes in Information Ecosystems" [5]. Our solution contemplates ecosystem characteristics (e.g. similarity between courses of multiple providers) which makes the learning process of providers (information systems) openly and collaboratively integrated. This is possible by the fact that when using the ecosystem perspective, providers and users become contributors to the learning processes so our recommendation system interacts and assists these ecosystems connections (e.g., finding suitable content for a student) so that processes are integrated.

6 CONCLUSION

Online distance education enables several universities and firms to make online courses available to students via the Internet in an innovative and democratic way. With the expansion of this movement through MOOCs, it may be difficult for students to choose the best courses among all providers, considering the increasing number of courses available in these MOOCs ecosystems. Therefore, studies have aimed to use topic modeling to create distributions and apply recommendation systems to recommend videos, courses, forums, etc. However, several students are still seeking more personalized recommendations through the collection of part of the courses from multiple MOOC providers, creating a set of modules to fill a knowledge gap. Also, with new data privacy issues, it is required that recommenders include just authorized data from these providers.

The effective contribution of this work is a proposal for a recommendation system in MOOCs ecosystems that combines a content-based recommendation with NMF for topic modeling. Hereby, it is possible to include scientific contributions involving the recommendation process and the work construction such as part of course recommendation besides the whole courses. In this paper, we also developed a new method for extracting API data from multiple MOOCs providers, in JSON format.

As future work, we intend to implement a sorting method to pedagogically verify a learning sequence of the recommended modules. Next, it is needed to execute one more evaluation for more users from multiple providers to validate the ordering and the relevance of recommended items. To do so, it is possible to build a web interface that makes it more dynamic and facilitates user interaction with the recommender system.

ACKNOWLEDGMENTS

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

Table 1: List of Recommended Modules for the Target User

Title	Provider	Link
Set Up Your Website	Udemy	/course/digital-marketing-data
Video Marketing	Udemy	/course/digital-marketing-masterclass
FBA Strategies Effective FBA Launch Formula	Udemy	/course/sells-like-hot-cakes-turnkey-amazon-fb-shopify-system/
FBA Strategies Creating A Perfect Listing Images and Keywords	Udemy	/course/sells-like-hot-cakes-turnkey-amazon-fb-shopify-system/
Conversion Rate Optimization How to have more traffic leads and sales	Udemy	/digital-marketing-management
Why NOW is the Perfect time to start making videos	Udemy	/be-comfortable-and-confident-on-camera

REFERENCES

[1] André Abdalla, Victor Ströele, Fernanda Campos, José Maria N David, and Regina Braga. 2018. Software Ecosystem Platform for Recommendation Systems. In *Proceedings of the XIV Brazilian Symposium on Information Systems (SBSI'18)*. ACM, New York, NY, USA, 70:1—-70:9. https://doi.org/10.1145/3229345.3229418

[2] Charu C. Aggarwal. 2016. *Recommender systems*. Springer International Publishing, Cham. https://doi.org/10.1002/9781119054252 arXiv:1202.1112

[3] Luisa Aires. 2015. E-Learning, Online Education and Open Education: A Contribution to a Theoretical Approach. *RIED. Revista Iberoamericana de Educación a Distancia* 19, 1 (sep 2015), 253–269. https://doi.org/10.5944/ried.19.1.14356

[4] Nikolaos Aletras and Mark Stevenson. 2014. Measuring the Similarity between Automatically Generated Topics. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*. Association for Computational Linguistics, Gothenburg, Sweden, 22–27. http://staffwww.dcs.shef.ac.uk/

[5] Renata Araujo. 2017. Information Systems and the Open World Challenges. *Boscarioli, C, R.M Araujo e R.S.P Maciel. I GranDSI-BR - Grand Research Challenges in Information Systems in Brazil 2016-2026. Special Committee on Information Systems (CE-SI)* (2017), 42–51.

[6] Chidansh Bhatt, Matthew Cooper, and Jian Zhao. 2018. SeqSense: Video Recommendation Using Topic Sequence Mining. In *Proceedings of the International Conference on Multimedia Modeling*, Springer (Ed.). Springer International Publishing, Cham, Switzerland, 252–263. https://doi.org/10.1007/978-3-319-73600-6_22

[7] Rodrigo Campos, Rodrigo Pereira Santos, and Jonice Oliveira. 2018. Web-Based Recommendation System Architecture for Knowledge Reuse in MOOCs Ecosystems. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, Salt Lake City, UT, 193–200. https://doi.org/10.1109/IRI.2018.00036

[8] Rodrigo Campos, Rodrigo Pereira Santos, and Jonice Oliveira. 2019. A Recommendation System Enhanced by Topic Modeling for Knowledge Reuse in MOOCs Ecosystems. In *Reuse in Intelligent Systems*. CRC Press, Boca Raton, Florida, EUA, 116–142.

[9] Alejandro Corbellini, Cristian Mateos, Alejandro Zunino, Daniela Godoy, and Silvia Schiaffino. 2017. Persisting big-data: The NoSQL landscape. *Information Systems* 63 (2017), 1–23.

[10] Christian Desrosiers and George Karypis. 2011. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In *Recommender Systems Handbook*. Springer US, Boston, MA, 107–144. https://doi.org/10.1007/978-0-387-85820-3_4

[11] Hendrik Drachsler, Katrien Verbert, Olga C. Santos, and Nikos Manouselis. 2015. Panorama of Recommender Systems to Support Learning. In *Recommender Systems Handbook*. Springer, Boston, MA, 421–451. https://doi.org/10.1007/978-1-4899-7637-6_12 arXiv:arXiv:1011.1669v3

[12] Derek Greene, Derek O'Callaghan, and Pádraig Cunningham. 2014. How Many Topics? Stability Analysis for Topic Models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg, Berlin, Heidelberg, 498–513. arXiv:1404.4606 http://arxiv.org/abs/1404.4606

[13] Xia Jing and Jie Tang. 2017. Guess You Like: Course Recommendation in MOOCs. In *Proceedings of the International Conference on Web Intelligence*. ACM, New York, NY, USA, 783–789. https://doi.org/10.1145/3106426.3106478

[14] Da Kuang, Jaegul Choo, and Haesun Park. 2015. Nonnegative Matrix Factorization for Interactive Topic Modeling and Document Clustering. In *Partitional Clustering Algorithms*, Cham (Ed.). Springer, Cham, 215–243. https://doi.org/10.1007/978-3-319-09259-1__7

[15] Diogo Nolasco and Jonice Oliveira. 2016. Topic Modeling and Label Creation: Identifying Themes in Semi and Unstructured Data. In *Brazilian Symposium on Database - Topics in data and information management*. Eduardo Ogasawara, Vaninha Vieira. (Org.), Porto Alegre: SBC, 87–112. http://sbbd2016.fpc.ufba.br/sbbd2016/minicursos/minicurso4.pdf(inPortuguese)

[16] Derek O'Callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. 2015. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications* 42, 13 (2015), 5645–5657. https://doi.org/10.1016/j.eswa.2015.02.055

[17] Tianhao Pan, Weifeng Zhang, Ziyuan Wang, and Lei Xu. 2016. Recommendations Based on LDA Topic Model in Android Applications. In *Proceedings - 2016 IEEE International Conference on Software Quality, Reliability and Security-Companion, QRS-C 2016*. IEEE, Vienna, 151–158. https://doi.org/10.1109/QRS-C.2016.24

[18] Christos H Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. 2000. Latent Semantic Indexing: A Probabilistic Analysis. *J. Comput. System Sci.* 61, 2 (2000), 217—-235. http://web.cse.msu.edu/{~}cse960/Papers/LSI/LSI-papadimitriou.pdf

[19] Jeungeun Song, Yin Zhang, Kui Duan, and M Shamim Hossain. 2017. TOLA: Topic-oriented learning assistance based on cyber-physical system and big data. *Future Generation Computer Systems* 75 (2017), 200–205. https://doi.org/10.1016/j.future.2016.05.040