campos_2022_providing_recommendations_for_communities_of_learners_in_moocs_ecosystems

Year

2022

Author(s)

Rodrigo Campos and Rodrigo Pereira dos Santos and Jonice Oliveira

Title

Providing recommendations for communities of learners in MOOCs ecosystems

Venue

Expert Systems With Applications

Topic labeling

Fully automated

Focus

Primary

Type of contribution

Established approach (with customisations)

Underlying technique

Statistical methods (extracting top-1 or top-3 terms from the top documents)

Topic labeling parameters

(See `Algorithm 1`)

D: 30

T: 10

Label generation

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 follows some steps based on another method proposed by Nolasco and Oliveira, 2016.

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:

1. The Keywords Selection (KS)
2. The Text Selection (TS) with the Fast Keyword Extraction algorithm.

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.

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.

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.

Evaluation procedure

Approach = TS

Starting with the ''Selection of Candidates'', we selected top-30 documents associated with the topic.
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.

Motivation

This automatic identification of topic labels helps in describing item layer topics and in identifying the ‘‘user topics of interest’’.
Therefore, obtaining relevant strings to represent topics, verified in the provided evaluation (see `Label quality evaluation`), contributes to reduce the students’ knowledge gap by identifying their interests.

Topic modeling

Modified NMF method (Campos et al., 2020)
Baseline: LDA

Topic modeling parameters

Min nr of topics (kmin): 5
Max nr of topics (kmax): 30

Nr. of topics

14

Label

Top-1 / Top-3 terms extracted from the top 30 documents associated with a topic

Label selection

/

Label quality evaluation

Evaluation of topic labeling step through comparisons between our automatic labels and

providers' labels using the cosine distance.

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.

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)

Assessors

/

Domain

Domain (paper): Recommender systems (for MOOCs)
Domain (corpus): MOOCs courses

Problem statement

The growth of Massive Open Online Courses (MOOCs) brings some difficulties for students in choosing suitable courses in these ecosystems.
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.

Corpus

Origin: Various course providers
Nr. of documents: 106,574
Details: Almost 95% of the dataset comes from Udemy.

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.

Document

Module contents information, module title and description, videos, exercises, articles information, and the course short description (in the case of edX).

Pre-processing

Some data treatments are performed as well as for item topic modeling, such as

- non-noun removal
- stop words removal
- punctuation removal

```
@article{campos_2022_providing_recommendations_for_communities_of_learners_in_moocs_ecosystems,
  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.},
  author = {Rodrigo Campos and Rodrigo Pereira dos Santos and Jonice Oliveira},
  date-added = {2023-03-10 12:54:01 +0100},
  date-modified = {2023-03-10 12:54:01 +0100},
  doi = {https://doi.org/10.1016/j.eswa.2022.117510},
  issn = {0957-4174},
  journal = {Expert Systems with Applications},
  keywords = {Online education systems, Content-based recommendation, Topic
```

```
modeling, Non-negative matrix factorization, Unsupervised machine learning},
  pages = {117510},
  title = {Providing recommendations for communities of learners in MOOCs ecosystems},
  url = {https://www.sciencedirect.com/science/article/pii/S0957417422008375},
  volume = {205},
  year = {2022}}
```

#Thesis/Papers/Initial