A Combined E-Learning Course Recommender System

https://doi.org/10.3991/ijet.v18i06.36987

Jallal Talaghzi[1(✉)], Mostafa Bellafkih[1], Abdellah Bennane[2],
Mohammed Majid Himmi[3], Mohammed Amraouy[1]
[1]National Institute of Posts and Telecommunications, Rabat, Morocco
[2]Inspectors Training Center for Education (CFIE) & Mohammed V University, Rabat, Morocco
[3]Faculty of Science, Mohammed V University Rabat, Morocco
j.talaghzi@gmail.com

Abstract—In this paper, we propose learners course recommender system of the E-Dirassa platform. This system, which distinguishes between new and active learners, adopts three complementary recommendation approaches: content-based recommendation as well as course-based collaborative filtering for active learners, and static profile-based collaborative filtering for new learners. Let us also note the use of multi-criteria decision aid (MDA) techniques for the choice of the textual similarity measure to be used in the first approach of content-based recommendation.

Keywords—recommender systems, content-based course recommendation, course-based collaborative filtering, static profile-based collaborative filtering, multi-criteria decision aid

1 Introduction

Recommender systems appeared shortly after the invention of the World wide web (www) and quickly became a focus of interest for many researchers. A brief historical review of recommender systems has been made in [1], studying two aspects: recommendation models and architectures of typical recommender systems. They find their place in various areas in the Web. Thus, they are present in recognized platforms such as Netflix as a playlist generator, Amazon as a product recommender system and Facebook as a content recommender system [2].

Nowadays, course recommendation is of interest to many researchers. Works like [3], [4] have focused on the development of course recommender systems. Other research has addressed specific problems like the information overload problem produced in big learning objects collections [5]. Course recommendation plays a very important role in online adaptive learning. In recent research [6], the recommended courses were classified as adapted object just like learning strategy, learning path, assessment, etc.

Recommender systems typically use collaborative filtering or content-based filtering. Collaborative filtering approaches build a model from the decisions made by other similar users. This model is then used to predict which items the user may be interested in. Content-based filtering approaches use a series of discrete, pre-labeled characteristics of an item in order to recommend additional items with similar properties. Current recommender systems generally combine one or more approaches in a hybrid system. A categorization of recommender systems has been made in [7].

Our job is to design learners course recommendation system in the E-Dirassa platform. This system will combine three recommendation approaches: content-based recommendation and course-based collaborative filtering for active learners, as well as static profile-based collaborative filtering for new learners (cf. Figure 1).

Moreover, content-based recommendation requires the use of a textual similarity measure. However, there are several measures in the literature [8], [9]. Each of these measures has advantages and disadvantages. This brings us back to wondering about the most effective similarity measures to use for text comparison. For this reason, we will use a multi-criteria decision aid to select the most appropriate textual similarity measure for our context.

We believe that using this combined recommender system will meet learners' needs and solve common recommender problems, especially the cold start problem and the over-specialization problem.

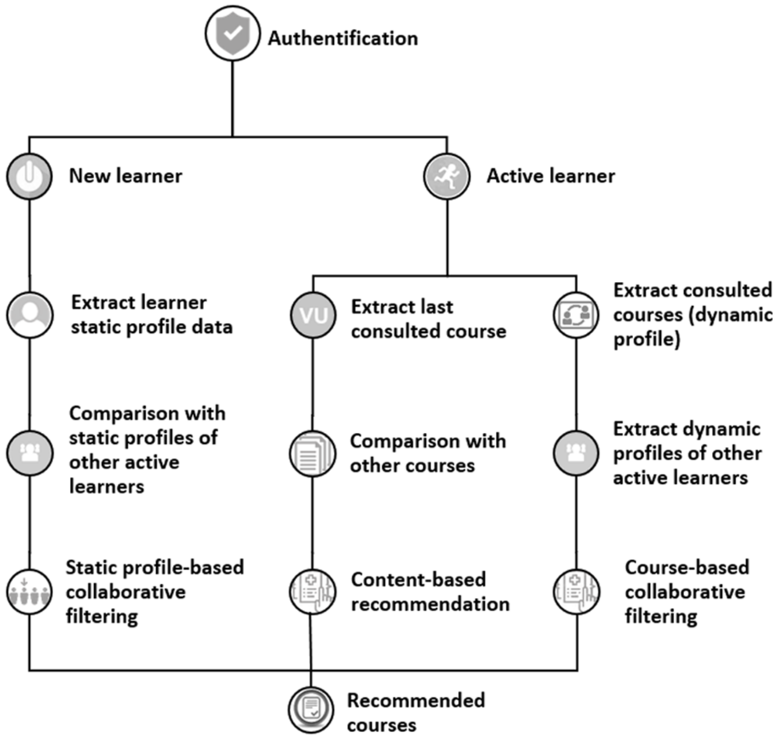


Fig. 1. General diagram of the course recommendation system

2 Related works

Sundari et al. [10] proposed a recommendation system to deal with random choices that do not take into consideration the chosen course level. This recommender system based on type classification rules (IF-THEN), adopts a methodological approach that is based on the student's performance in previous semesters and the student's choice of electives.

Taha et al. [11] underlined the problems related to the choice or the recommendation of the courses: on the one hand the students cannot confirm their interests from the title of a course. On the other hand, the advisor cannot track student performance in all past courses. For this reason, a collaborative filtering system was introduced in order to predict students' academic performance and interests based on the similar profiles of other students.

Lee et al. [12] designed a mobile course recommendation system. This system uses an inference engine that takes into consideration lesson sequences and student information. Association rules were used to find relationships between subjects by comparing the history of courses taken to existing course trees.

Sowmiya et al. [13] used a course recommendation approach based on collaborative filtering. Students are represented by a pair of courses taken and ratings. The objective was to recommend courses that increase the score of graduating attributes by calculating the similarity between students.

Mondal et al. [14] proposed a grade-based recommendation system. This system first ranks learners based on their past performance using the k-means clustering algorithm. Then, collaborative filtering is used in the cluster to recommend the most suitable courses.

Gulzar et al. [15] introduced an ontology-based course recommendation system aimed at supporting learner-centred learning. The proposed system is composed of four modules. Learner Profile Generation allowing to create a model of the learner, Recommender Generator, Recommendations Display and Course Ontology which describes the relation between the courses of the domain.

Ansari et al. [16] have proposed a hybrid course recommendation system that combines the two recommendation approaches: content-based filtering and collaborative filtering in order to benefit from the advantages of both approaches. This system is essentially based on learner behaviours and activities to recommend relevant courses.

The common object of the works described in this section is the prediction of relevant courses likely to interest learners. The difference lies in the approaches and methods used for the recommendation. We are talking about recommender systems using methods such as classification (supervised learning), clustering (unsupervised learning), association rules, course ontologies, collaborative filtering, etc. Other works have used hybrid course recommendation systems that combine several approaches. A systematic review of hybrid recommender systems [17] concluded that mixed hybrid recommender systems such as "Collaborative Filtering – Content Based Filtering" with "Collaborative Filtering – Knowledge Based Filtering" or other mixed hybrid systems are rarely used and advised to explore them in future works. In addition, this study mentioned that 15 articles out of 76 analyzed used a hybrid system combining collaborative filtering with content-based filtering. These systems have dealt with recommendation issues such as cold start, accuracy, scalability, and diversity. For this reason, our work will adopt three combined recommendation approaches; Collaborative Filtering with Content Based Filtering for active learners and Static Profile Based Collaborative Filtering for new learners.

3 Description of E-Dirassa learning management system (LMS)

In this section, we will present our LMS E-Dirassa. The general structure of this LMS has four spaces (Figure 2). Each space contains a set of rubrics that allow the user – depending on his role – to use the LMS. Some spaces have rubrics with the same name but with different uses. Since it is about learner recommending courses, we will focus in this LMS on the rubrics relating to courses and learners.

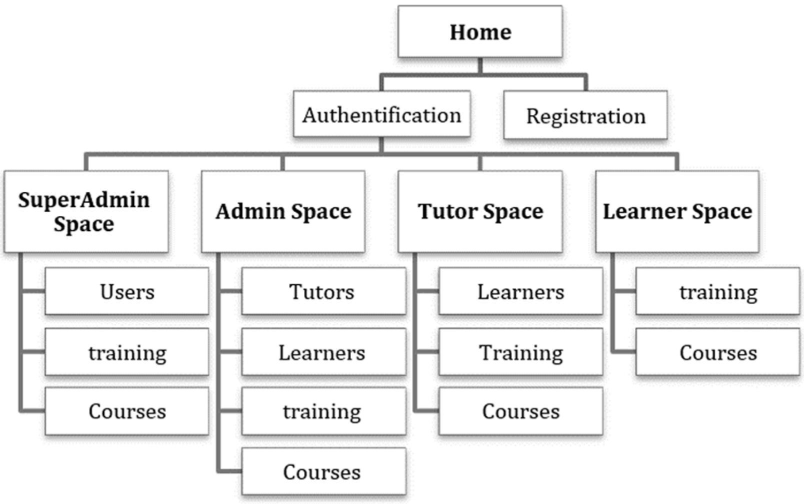


Fig. 2. General structure of the LMS E-Dirassa

3.1 Course description

A course is defined by a title, a description, a type, a status, a tutor and possibly a start and end date (Figure 3). In this article, we will use the course description to predict similar courses in the content-based recommendation approach. Each course may contain a set of grouped activities designed by tutors (Figure 4). The learner concerned by the course can consult and complete these activities in his own space (Figure 5).

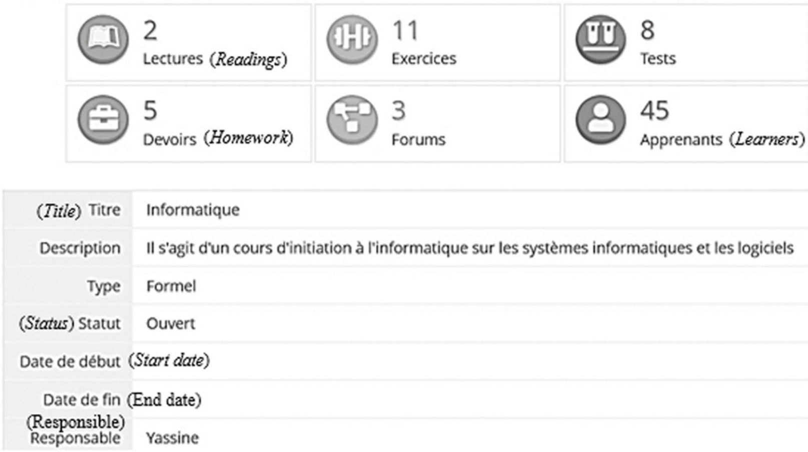


Fig. 3. Elements defining a course

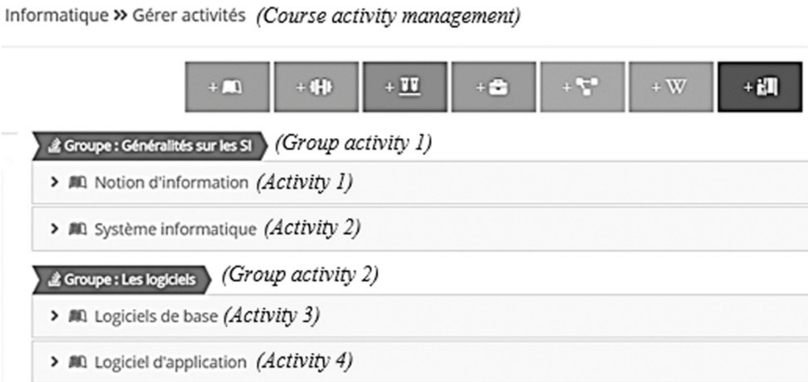
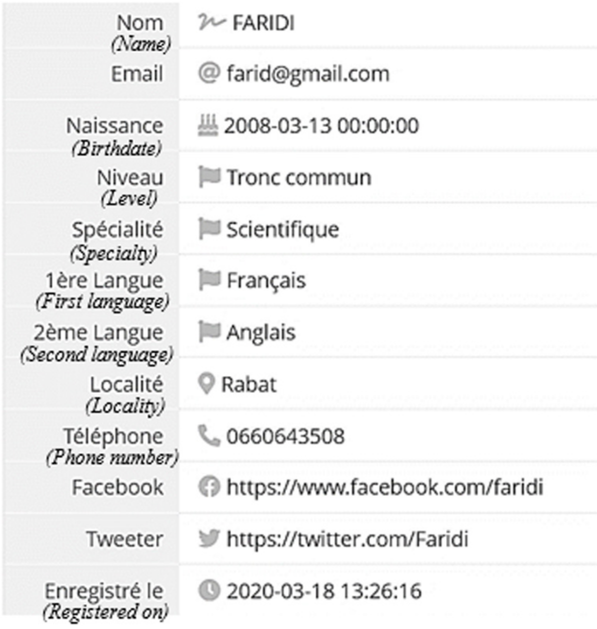


Fig. 4. Defining course activities

Fig. 5. Course activities in learner space

3.2 Learner profile

Learners are registered in E-Dirassa LMS by providing their personal information such as name, level, specialty, etc. (Figure 6). This information is used to build the learner static profiles that will later be used in course recommendations for new enrollees.



Nom *(Name)*	FARIDI
Email	farid@gmail.com
Naissance *(Birthdate)*	2008-03-13 00:00:00
Niveau *(Level)*	Tronc commun
Spécialité *(Specialty)*	Scientifique
1ère Langue *(First language)*	Français
2ème Langue *(Second language)*	Anglais
Localité *(Locality)*	Rabat
Téléphone *(Phone number)*	0660643508
Facebook	https://www.facebook.com/faridi
Tweeter	https://twitter.com/Faridi
Enregistré le *(Registered on)*	2020-03-18 13:26:16

Fig. 6. Learner's personal information

4 Course recommendation for active learners

Active learners are those who have already consulted at least one course on the platform. The recommendation of a new course can be done by offering them similar courses (content-based recommendation) or courses consulted by other learners (collaborative filtering). A comparison of the textual descriptions of the courses is necessary in order to present the most similar courses to those consulted.

4.1 Content-based recommendation

Content-based recommendation is a type of recommendation in which the decision to select a document or not is based on comparing the similarity between the texts. The main purpose of text comparison is to calculate how "close" two pieces of text are in sense or in surface proximity. The first approach is called semantic similarity and the second is called syntactic similarity.

Preprocessing of text descriptions. In order to produce the structures that will be used to represent the texts when calculating the similarities, the textual descriptions of the courses must first be decomposed into simpler lexical units. Several approaches are possible [18], [19], [20]:

Corpus-based similarity is a measure of semantic similarity that determines the similarity between words based on information obtained from large corpus. A corpus is a large collection of written or spoken texts that is used for linguistic research.

Knowledge-based similarity is one of the measures of semantic similarity that relies on identifying the degree of similarity between words using information derived from semantic networks.

String-based similarity is a measure of syntactic and semantic similarity which can be divided into two categories: Character-based Similarity and Term-based Similarity.

Corpus-based or knowledge-based semantic approaches have storage and complexity issues and are often domain-specific [21]. As the courses presented in the platform may come from several domains (Natural languages, Mathematics, Philosophy, Computer science, etc.), the use of the semantic similarity approach poses problems in this context. For this reason, we will adopt a syntactic similarity approach in order to compare the textual descriptions of courses that fall under different domains. More precisely, we will use the Term-based Similarity approach, which is a classic approach for syntactic similarity and which allows to define textual units using surface forms ("words"). These forms can be produced by simple techniques of automatic segmentation and lemmatization.

Text similarity measures. A measure of distance between words, is a way of representing by a number the difference between two words, or more generally two strings of characters. So, it's a form of mathematical distance and metric for strings. It measures the similarity or dissimilarity between two strings. Among the similarity measures most used in the comparison of texts: the Dice coefficient, the Jaccard index, the Euclidean distance, the cosine, …

Similarity Cosine is frequently used as a measure of similarity between two documents d_1 and d_2. This involves calculating the cosine of the angle between the vector representations of the documents to be compared.

$$sim_{cosinus}(d_1, d_2) = \frac{\overrightarrow{d_1}.\overrightarrow{d_2}}{\| d_1 \| \| d_2 \|} \tag{1}$$

Pearson's correlation coefficient calculates the similarity between two documents d_1 and d_2 as the cosine of the angle between their centered-reduced vector representations.

$$sim_{pearson}(d_1, d_2) = sim_{cosinus}(d_1 - \overline{d_1}, d_2 - \overline{d_2}) \tag{2}$$

Euclidean distance calculates the similarity between two documents d_1 and d_2 as the distance between their vector representations reduced to a single point.

$$sim_{euclidienne}(d_1, d_2) = \left\| \overline{d_1} - \overline{d_2} \right\| = \sqrt{\sum_{1}^{n} (d_1 - d_2)^2} \tag{3}$$

Where n is the total number of represented terms.

Jaccard coefficient or ***Jaccard index*** is the ratio between the cardinality (the size) of the intersection of the sets considered and the cardinality of the union of the sets. It allows to evaluate the similarity between the sets. The documents d_1 and d_2 are therefore represented, not as vectors, but as sets of terms.

$$sim_{Jaccard}(d_1, d_2) = \left\| \frac{d_1 \cap d_2}{d_1 \cup d_2} \right\| \tag{4}$$

The normalization of the Jaccard coefficient is given by:

$$sim_{NorJaccard}(d_1, d_2) = 1 - sim_{Jaccard}(d_1, d_2) \tag{5}$$

Dice index measures the similarity between two documents d_1 and d_2 based on the number of common terms in d_1 and d_2.

$$sim_{Dice}(d_1, d_2) = \frac{2N_c}{N_1 + N_2} \tag{6}$$

Where N_c is the number of common terms in d_1 and d_2, and N_1 (respectively N_2) is the number of terms in d_1 (respectively d_2). The normalization of the Dice index is given by:

$$sim_{NorDice}(d_1, d_2) = 1 - sim_{Dice}(d_1, d_2) \tag{7}$$

Performance criteria for textual similarity measures. In the platform, a course can include several activities (exercises, texts, videos, images, tests, surveys, etc.). Therefore, the measure of similarity in our case will relate to the descriptions attributed to the courses by the tutors. These descriptions can have different lengths (number of terms).

When we have to compare a short description with a longer one, we are in the case of documentary research and subsequently the measure must be sensitivity to common terms. When the course descriptions have quite similar lengths, we are in the case of textual data analysis. Our measure must then be more sensitive to non-common terms. Moreover, we wish to have a certain stability of the similarity measure between two given descriptions when they are compared exclusively or with other textual descriptions.

Thus, we define the following performance criteria of the textual similarity measures:

Stability (C1): Criterion C1 means that the distance between two descriptions remains unchanged when adding other descriptions in comparison. This property of invariance induces a certain stability of the results: indeed, the addition of a new text (new course description) will in no way affect the other distances.

The Table 1 below presents the distances between two descriptions when a third is added with non-intersecting terms for the first two descriptions. We calculated the coefficient of variation (CV) of the series of distances for each measurement to get an idea of the dispersion of distances around the mean. Thus, a CV equal to 0 shows that the similarity measure is perfectly stable. Since we favor the most stable similarity measures, another coefficient (CC1 = 1-CV) is calculated based on the CV to have increasing weightings towards the most stable measures.

Table 1. Stability coefficients of textual similarity measures

	Cosine	Pearson	Euclidean	Jaccard	Dice
Distance between two descriptions d_1 and d_2	0.5774	1.5813	3.7417	0.75	0.4
Distance with 1 term not common to d_1 and d_2	0.5774	1.1724	3.7417	0.75	0.4
Distance with 2 terms not common to d_1 and d_2	0.5774	1	3.7417	0.75	0.4
Distance with 3 terms not common to d_1 and d_2	0.5774	0.9050	3.7417	0.75	0.4
Distance with 4 terms not common to d_1 and d_2	0.5774	0.8448	3.7417	0.75	0.4
Distance with 5 terms not common to d_1 and d_2	0.5774	0.8032	3.7417	0.75	0.4
Coefficient of variation (CV)	0	0.277	0	0	0
CC1 = 1 – CV	**1**	**0.723**	**1**	**1**	**1**

We note that all the measures perfectly satisfy the stability criterion (C1) except for the Pearson correlation coefficient which is the least stable.

Sensitivity to non-common terms (C2): Criterion C2 means that the measure is sensitive only to the distinct parts between two textual descriptions. In other words, the measure will be sensitive to the "number" of different words between the compared descriptions. Table 2. represents the distances between two perfectly similar descriptions d_1 and d_2 at the start, then with non-common terms. The measure of sensitivity to non-common terms is given by CC2, and is equal to the coefficient of variation of the distance series for each measure of similarity.

Table 2. Sensitivity coefficients to non-common terms of textual similarity measures

	Cosine	Pearson	Euclidean	Jaccard	Dice
Two perfectly similar descriptions d_1 and d_2	0	0	0	0	0
d_1 and d_2 with 1 non common term	0.0339	0.0561	1	0.25	0.0769
d_1 and d_2 with 2 non common terms	0.0646	0.0568	1.4142	0.4	0.1429
d_1 and d_2 with 3 non common terms	0.0925	0.0551	1.7321	0.5	0.2
d_1 and d_2 with 4 non common terms	0.1181	0.0535	2	0.5714	0.25
d_1 and d_2 with 5 non common terms	0.1416	0.0523	2.2361	0.625	0.2941
CC2 = CV	**0.705**	**0.4912**	**0.5806**	**0.5964**	**0.6846**

Note that the distance that best satisfies criterion C2 is that of the cosine, while the Pearson correlation coefficient is the distance that is least sensitive to non-common terms.

Sensitivity to common terms (C2): Criterion C3 means that the measure is sensitive only to the common parts between two textual descriptions. In other words, the measure will be sensitive to the "number" of common words between the compared descriptions.

Table 3. Sensitivity coefficients to common terms of textual similarity measures

	Cosine	Pearson	Euclidean	Jaccard	Dice
Two completely different descriptions d_1 and d_2	1	1.7746	8.8318	1	1
d_1 et d_2 with 1 common term	0.968	1.7282	8.8318	0.8889	0.875
d_1 et d_2 with 2 common terms	0.9385	1.6987	8.8318	0.8	0.777
d_1 et d_2 with 3 common terms	0.9111	1.6783	8.8318	0.7273	0.7
d_1 et d_2 with 4 common terms	0.8857	1.6633	8.8318	0.6667	0.636
d_1 et d_2 with 5 common terms	0.8619	1.6519	8.8318	0.6154	0.583
CC3 = CV	**0.0557**	**0.027**	**0**	**0.1836**	**0.204**

Table 3 represents the distances between two perfectly different descriptions d_1 and d_2 at the start, then with common terms. The measure of sensitivity to common terms is given by CC3 which is equal to the coefficient of variation of the distance series for each similarity measure. Note that the Euclidean distance is totally insensitive to common terms. The most suitable measure for criterion C3 is the Dice index. The results obtained show that a measure can be suitable for a given criterion and less suitable for others. For this reason, we will use Multi-Criteria Decision Aid (MDA) methods to choose the most suitable one for comparing course descriptions.

Choice of similarity measure. Multi-criteria decision aid methods include methods for aggregating several criteria with the aim of selecting one or more actions or solutions. They aim to solve problems with several alternatives (measurements in our case) and by applying several criteria (C1, C2 and C3 in our case) simultaneously. These criteria are most often conflicting and have equal or unequal importance (weight, priority); multi-criteria analysis aims to provide tools that will allow progress in solving a decision problem where several opinions – often contradictory – must be taken into account. DMA is therefore can be considered as an activity that allows to build justified arguments for the taken decision [22]. There are several methods for AMD [23], for example:

- WSM (Weight Sum Method)
- WPM (Weight Product Method)
- AHP (Analytic Hierarchy Process)
- ELECTRE (Outranking method)
- PROMETHEE (Preference Ranking Organisation METHod for Enrichment Evaluations).

We chose to use WSM because it is ideal for one-dimensional problems. In addition, the calculations are less. But, before applying this method, it is first necessary to build the decision matrix that matches the textual similarity measures and the criteria.

The relative importance of the criteria granted by the decision makers is represented by weights. We consider – in our context – that the performance criteria of the measures have the same importance, which means that each of the criteria will have a weight equal to 1. The correspondence between the different criteria and alternatives (measures) allows us to establish the decision matrix represented by Table 4.

Table 4. Decision matrix

Criteria / Measures	C1	C2	C3
	Correspondence Coefficients Relating to Measures		
Cosine	1	0.705	0.0557
Pearson	0.723	0.4912	0.027
Euclidean	1	0.5806	0
Jaccard	1	0.5964	0.1836
Dice	1	0.6846	0.2043

WSM is based on the weighted sum model and the principle of additive utility. Thus, for a maximization problem the best choice is given by:

$$A_{WSM} = MAX_i \sum_{j=1}^{n} a_{i,j} \,.\, k_j \quad for\ i = 1, 2, \ldots, m \tag{8}$$

Where $a_{i,j}$ Is the correspondence coefficient of criterion j with measure i and k_j Corresponds to the weight given to criterion j (in our case the weights are equal to 1).

The measure with the highest score performs best. Thus, the application of this method gave the following results in Table 5.

Table 5. Scores of measures by the WSM method

Measure	WSM Score
Cosine	1.7607
Pearson	1.2412
Euclidean	1.5806
Jaccard	1.78
Dice	1.8889

We note that the WSM method advanced the Dice index compared to the other distances. We can then conclude that the Dice index is the most appropriate measure for our context.

4.2 Course-based collaborative filtering

Collaborative filtering is based on the adage: If two learners have consulted identical courses in the past, they have a high probability of consulting the same content in the future.

Digital traces (learner behavior) are the key to this type of recommendation. These traces have been used in several works on online learning such as detecting the Learner's motivational state in online learning situation [24], prediction of the Program for International Student Assessment (PISA) score [25] or recommending educational content based on learner's annotative activity [26].

Recommendations from collaborative filtering can be calculated in various ways. particularly according to the profile of the learners (User-based collaborative filtering), or by using course profiles (Course-based collaborative filtering).

There are at least two main types of Course-based collaborative filtering. First, binary models that are only based on whether or not a learner has viewed a given course. Then in models with annotations made by learner; learners are asked to rate or annotate the different courses.

Our platform does not yet allow users (learners) to rate a given course. For this reason, we have chosen to opt for course-based collaborative filtering using a binary model (course consulted or not). So, the correspondence between learners and courses can be presented in the form of a binary evaluation matrix (cf. Table 6).

Table 6. Binary evaluation matrix

	CS_1	**CS_2**	**CS_3**	**…**
L_1	1	0	1	…
L_2	1	1	0	…
…	…	…	…	…

Where L_i represents learner i and CS_j represents course j. Thus, a correspondence equal to 1 between a learner and a given course means that the latter has been consulted by the learner; a correspondence equal to 0 means the learner has not viewed the course yet and may be recommended in the future.

Each column of the binary evaluation matrix represents a course vector made up of zeros and ones. To recommend a course to a given learner, we must first find the similarity between all the pairs of course vectors. This similarity can be found in different ways. An evaluation of measures for item-based recommender systems has been produced by Demiriz [27], and concluded that the cosine approach is superior to all other approaches analyzed.

Note that the cosine of two binary vectors is a necessarily non-negative quantity. Thus, for n courses, a learner interested in a course i, can have as a recommendation the course j^* such that:

$$sim_{cosinus}(Cs_i, Cs_{j*}) = Max\left(sim_{cosinus}(Cs_i, Cs_j)\right) \; with \; j = 1, 2, \ldots, \; n \; and \; i \neq j \quad (9)$$

5 Course recommendation for new learners

New learners are those who have just registered on the platform without accessing any course. For these learners, only personal information such as age, school level, etc. is available. To recommend courses to them, we will proceed by an approach based on the static profile, since the recommendation based on the course is not possible in this particular case.

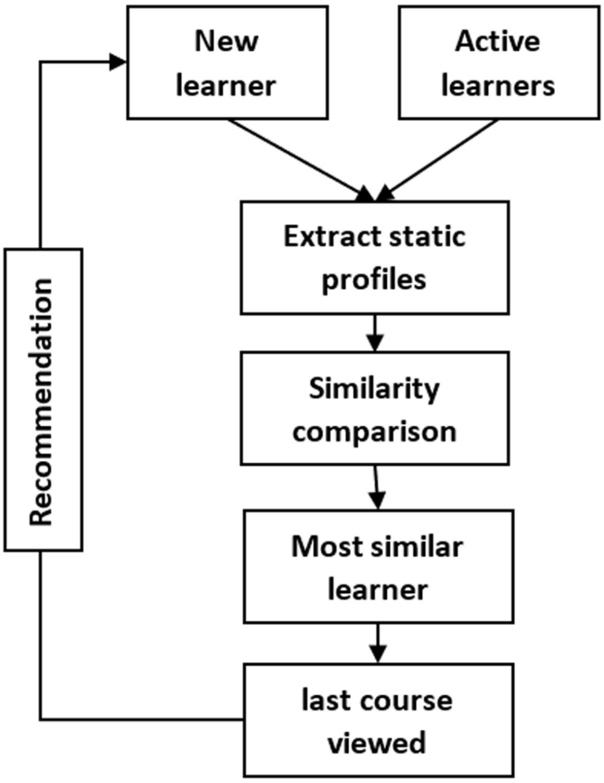


Fig. 7. Course recommendation scheme for new learners

Thus, our approach consists in comparing the static profile of a new learner with those of other active learners, selecting the most similar learner to our target and recommending the last course consulted by his counterpart (Figure 7).

5.1 Description of the static profile

Active learners in the platform have two types of profiles: a dynamic profile drawn from their activities on the platform, namely the courses consulted, and another static profile based on registration information. New learners do not have a dynamic profile (no course consulted) and the only information that we have is their static profiles.

Table 7. Codification of static profile items

Age	**[6 Years, 10 Years]**	**[10 Years, 14 Years]**	**…**
	1	**2**	**…**
Scholar level	**Level 1**	**Level 2**	…
	1	2	…
Language	**English**	**French**	…
	1	2	…
Speciality	**Speciality 1**	**Speciality 2**	…
	1	2	…

To build a learner's static profile, we will use self-reported enrollment information: age, scholar level, first language, second language, and optionally specialty.

Note that the items of the static profile have values of different types (numerical and textual). This heterogeneity of data makes it difficult to manipulate items for the selection of similar learners. For this reason, we will quantify these data by assigning them codes as shown in Table 7.

Thus, learners registered in the platform are assigned static profiles as mentioned in Table 8.

Table 8. Example of codified static learner profiles

	Age	Level	1st Language	2nd Language	Speciality
Learner 1	1	2	2	1	1
Learner 2	3	7	1	3	3
Learner 3	1	1	2	1	1
…	…	…	…	…	…

5.2 Similarity calculation between learners

The similarity will be calculated by comparing the static profile of a new learner with those of active learners in the platform. Each static profile is represented by a vector composed of five numerical coordinates. Several similarity measures exist for this type of data [28]. Euclidean distance is one of the classic similarity measures that has proven its effectiveness in several machine learning algorithms such as K-NN and K-Means. We will use it to calculate the distance between the new learner and the other active learners.

6 Discussion

The proposed course recommendation system distinguishes between two types of learners. On the one hand, active learners who have already used the course material in the platform and on the other hand new learners who have just registered and who have no experience on the platform (no course consulted).

For active learners, the recommendation system makes it possible to offer courses according to two approaches. The first approach is a content-based recommendation. While the second is course-based collaborative filtering.

The first recommendation approach makes it possible to offer learners similar courses to those already consulted. This says that a learner who has just consulted an algebra course in mathematics will exclusively have proposal courses in the same discipline. This approach will be beneficial as learners will have an opportunity to review in order to reinforce their acquired background. It also makes it possible to recommend new courses or even courses that are not popular, which is not the case for the second course-based collaborative filtering approach. The disadvantage of this approach is over-specialization, that is to say that we are limited to similar courses and that the answers are too homogeneous.

The second course-based collaborative filtering approach comes to address the over-specialization problem. Thus, a learner who has just consulted a course in mathematics may have proposals in other disciplines such as the French language or the physical sciences. This heterogeneity of response allows learners to discover other courses that interest their similar peers and not be limited to a single discipline.

The two approaches mentioned above have the common advantage of being non-intrusive systems [29], that is to say that there is no information requested from the users of the platform and that the data retrieved is correct and does not contain no reporting bias. In contrast, they have the disadvantage of cold start when a new learner uses the platform and there is no history on his behavior.

For this reason, we have proposed a third recommendation approach for new learners: static profile-based collaborative filtering. This approach makes it possible to divert the cold start problem and thus offer courses to a new registrant based solely on his static profile and that of other active learners.

7 Conclusion

We have proposed in this article a course recommendation system based on three complementary approaches: Content-based recommendation, course-based collaborative filtering and static profile-based collaborative filtering. Each of these approaches has advantages and disadvantages. Combining these approaches into a single recommender system has helped overcome the shortcomings that each of them has, in particular the problems of over-specialization and cold start. We also note the use of MDA techniques to select the most appropriate textual similarity measure for the first content-based recommendation approach.

8 References

[1] A. Beheshti, S. Yakhchi, S. Mousaeirad, S. M. Ghafari, S. R. Goluguri, and M. A. Edrisi, ‘Towards Cognitive Recommender Systems’, Algorithms, vol. 13, no. 8. MDPI AG, p. 176, Jul. 22, 2020. https://doi.org/10.3390/a13080176

[2] Beheshti, Amin, Shahpar Yakhchi, Salman Mousaeirad, Seyed Mohssen Ghafari, Srinivasa Reddy Goluguri, and Mohammad Amin Edrisi. ‘Towards Cognitive Recommender Systems’, Algorithms, vol. 13, no. 8, p. 176, 2020. https://doi.org/10.3390/a13080176

[3] H. Murad and L. Yang, 'Personalized E-Learning Recommender System using Multimedia Data', International Journal of Advanced Computer Science and Applications, vol. 9, no. 9. The Science and Information Organization, 2018. https://doi.org/10.14569/IJACSA.2018.090971
[4] S. Dol Aher, 'Course Recommender System in E-Learning', International Journal of Computer Science and Communication, vol. 3, pp. 159–164, Nov. 2022.
[5] O. Sanjuán, B. Pelayo García-Bustelo, R. Gonzalez Crespo, and E. Franco, 'Using Recommendation System for E-learning Environments at Degree Level', International Journal of Interactive Multimedia and Artificial Intelligence, vol. 1, pp. 67–70, Dec. 2009.
[6] J. Talaghzi, A. Bennane, M. M. Himmi, M. Bellafkih, and A. Benomar, 'Online Adaptive Learning', Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications. ACM, Sep. 23, 2020. https://doi.org/10.1145/3419604.3419759
[7] D. Vats and A. Sharma, 'Recommendation Systems: Classification, Open Issues and Recent Developments', International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 9. Blue Eyes Intelligence Engineering and Sciences Engineering and Sciences Publication – BEIESP, pp. 3224–3231, Jul. 30, 2019. https://doi.org/10.35940/ijitee.J8831.078919
[8] J. Wang and Y. Dong, 'Measurement of Text Similarity: A Survey', Information, vol. 11, no. 9. MDPI AG, p. 421, Aug. 31, 2020. https://doi.org/10.3390/info11090421
[9] M. Benard Magara, S. O. Ojo, and T. Zuva, 'A Comparative Analysis of Text Similarity Measures and Algorithms in Research Paper Recommender Systems', 2018 Conference on Information Communications Technology and Society (ICTAS). IEEE, Mar. 2018. https://doi.org/10.1109/ICTAS.2018.8368766
[10] M. R. Sundari, G. Shreya, and T. Jawahar, 'Course Recommendation System', International Journal of Computer Applications, vol. 175, no. 29. Foundation of Computer Science, pp. 13–16, Nov. 17, 2020. https://doi.org/10.5120/ijca2020920823
[11] K. Taha, 'CRS', Student Engagement and Participation. IGI Global, pp. 553–570, 2018. https://doi.org/10.4018/978-1-5225-2584-4.ch028
[12] Y. Lee, 'An Intelligent Course Recommendation System', The Smart Computing Review, vol. 1, no. 1. The Korea Academia-Industrial Cooperation Society, Oct. 2011. https://doi.org/10.6029/smartcr.2011.01.006
[13] S. A. Sowmiya and P. Hamsagayathri, 'A Collaborative Approach for Course Recommendation System', Lecture Notes in Electrical Engineering. Springer Singapore, pp. 527–536, Sep. 19, 2020. https://doi.org/10.1007/978-981-15-7241-8_39
[14] B. Mondal, O. Patra, S. Mishra, and P. Patra, 'A Course Recommendation System Based on Grades', 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA). IEEE, Mar. 2020. https://doi.org/10.1109/ICCSEA49143.2020.9132845
[15] Z. Gulzar and A. Leema, A Framework for Ontology Based Course Recommendation System. 2020.
[16] M. H. Ansari, M. Moradi, O. NikRah, and K. M. Kambakhsh, 'CodERS: A Hybrid Recommender System for an E-learning System', in 2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS), Dec. 2016, pp. 1–5. https://doi.org/10.1109/ICSPIS.2016.7869884
[17] E. Çano and M. Morisio, 'Hybrid Recommender Systems: A Systematic Literature Review', Intelligent Data Analysis, vol. 21, no. 6. IOS Press, pp. 1487–1524, Nov. 15, 2017. https://doi.org/10.3233/IDA-163209
[18] W. H.Gomaa and A. A. Fahmy, 'A Survey of Text Similarity Approaches', International Journal of Computer Applications, vol. 68, no. 13. Foundation of Computer Science, pp. 13–18, Apr. 18, 2013. https://doi.org/10.5120/11638-7118

[19] V. M.K and K. K, 'A Survey on Similarity Measures in Text Mining', Machine Learning and Applications: An International Journal, vol. 3, no. 1. Academy and Industry Research Collaboration Center (AIRCC), pp. 19–28, Mar. 30, 2016. https://doi.org/10.5121/mlaij.2016.3103
[20] J. Wang and Y. Dong, 'Measurement of Text Similarity: A Survey', Information, vol. 11, no. 9. MDPI AG, p. 421, Aug. 31, 2020. https://doi.org/10.3390/info11090421
[21] E. Negre, 'Comparaison De Textes: Quelques Approches...', Apr. 2013.
[22] W. Ouerdane, N. Maudet, and A. Tsoukiàs, 'Argumentation Theory and Decision Aiding', International Series in Operations Research & Management Science. Springer US, pp. 177–208, 2010. https://doi.org/10.1007/978-1-4419-5904-1_7
[23] M. El Amine, J. Pailhès, and N. Perry, 'Selection and Use of a Multi-criteria Decision Aiding Method in the Context of Conceptual Design with Imprecise Information: Application to a Solar Collector Development', Concurrent Engineering, vol. 24, no. 1. SAGE Publications, pp. 35–47, Jan. 14, 2016. https://doi.org/10.1177/1063293X15613838
[24] M. Amraouy, A. Bennane, M. M. Himmi, M. Bellafkih, and A. BenOmar, 'Detecting the Learner's Motivational State in Online Learning Situation Towards Adaptive Learning Environments', Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications. ACM, Sep. 23, 2020. https://doi.org/10.1145/3419604.3419760
[25] I. Smirnov, 'Predicting PISA Scores from Students' Digital Traces', Proceedings of the International AAAI Conference on Web and Social Media, vol. 12, no. 1. Association for the Advancement of Artificial Intelligence (AAAI), Jun. 15, 2018. https://doi.org/10.1609/icwsm.v12i1.14996
[26] O. Mazhoud, A. Kalboussi, and A. H. Kacem, 'Educational Recommender System based on Learner's Annotative Activity', International Journal of Emerging Technologies in Learning (iJET), vol. 16, no. 10. International Association of Online Engineering (IAOE), p. 108, May 25, 2021. https://doi.org/10.3991/ijet.v16i10.19955
[27] A. Demiriz, 'Enhancing Product Recommender Systems on Sparse Binary Data', Data Mining and Knowledge Discovery, vol. 9, no. 2. Springer Science and Business Media LLC, pp. 147–170, Sep. 2004. https://doi.org/10.1023/B:DAMI.0000031629.31935.ac
[28] M. J. Lesot, M. Rifqi, and H. Benhadda, 'Similarity Measures for Binary and Numerical Data: a Survey', International Journal of Knowledge Engineering and Soft Data Paradigms, vol. 1, no. 1. Inderscience Publishers, p. 63, 2009. https://doi.org/10.1504/IJKESDP.2009.021985
[29] O. Zaliane, 'Recommender Systems for E-learning: Towards Non-intrusive Web Mining', in Data Mining in E-Learning, vol. 4, Data Mining in E-Learning (Advances in Management Information), 2006, pp. 79–93. https://doi.org/10.2495/1-84564-152-3/05

9 Authors

Jallal Talaghzi is a pedagogical inspector in computer science and a current PhD student National Institute of Posts and Telecommunications, Rabat, Morocco. His research interests focus on Human computer interaction, Artificial intelligence and Adaptive learning.

Mostafa Bellafkih is professor researches in The National Institute of Posts and Telecommunications (INPT) in Rabat, Morocco since 1995. He received the PhD thesis in Computer Science from the University of Paris 6, France, in June 1994 and Doctorates Science in Computer Science (option networks) from the University of Mohammed V in Rabat, Morocco, in May 2001. His research interests include the network management, knowledge management, A.I., Data mining and Database.

Abdellah Bennane is professor researches at the Training Center of Teaching Inspectors, Rabat, Morocco. He is a professional in applied informatics in education. His recent research is e-learning, teaching software and use on machine learning techniques.

Mohammed Majid Himmi is a professor of higher education in the physics department at the faculty of science, Mohammed V University – Rabat. Head of the Computer Science, Applied Mathematics, Artificial Intelligence and Pattern Recognition Team. He has directed and co-directed multiple theses in signal processing, telecommunications, pattern recognition and artificial intelligence. He coordinates the funded project "Design and implementation of an e-learning platform for multimodal interaction".

Mohammed Amraouy is a pedagogical inspector in computer science and part-time trainer at Regional Center for Education and Training Profession Oujda, Morocco, and a current PhD student National Institute of Posts and Telecommunications, Rabat, Morocco. His research interests focus on Human computer interaction, Artificial intelligence and Online learning assessment.

Article submitted 2022-11-22. Resubmitted 2023-01-12. Final acceptance 2023-01-13. Final version published as submitted by the authors.