



Finding optimal pedagogical content in an adaptive e-learning platform using a new recommendation approach and reinforcement learning

Youness Madani¹ · Hanane Ezzikouri¹ · Mohammed Erritali¹ · Badr Hssina²

Received: 24 July 2019 / Accepted: 30 October 2019 / Published online: 11 December 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

In the learning process, learners have different skills and each one has his own knowledge and his own ability to learn. The adaptive e-learning platforms try to find optimal courses for learners based on their knowledge and skills. Learning online using e-learning platforms becomes indispensable in the teaching process. Companies and scientific researchers try to find new optimal methods and approaches that can improve education online. In this paper, we propose a new recommendation approach for recommending relevant courses to learners. The proposed method is based on social filtering (using the notions of sentiment analysis) and collaborative filtering for defining the best way in which the learner must learn, and recommend courses that better match the learner's profile and social content. Our work consists also in proposing a new reinforcement learning approach which helps a learner to find the optimal learning path that can improve the quality of learning.

Keywords E-learning · Collaborative filtering · Social filtering · Semantic similarity · Fuzzy logic · Reinforcement learning

1 Introduction

With the exponential growth of the population that want to learn online, the e-learning platforms need to adapt and innovate in the way they suggest courses to learners. In the literature, we find a lot of methods and approaches that try to find optimal courses to a learner like those based on algorithms such as genetic algorithm, or the use of the machine learning approaches. In recent years companies and researchers begin to use the basis of recommendation systems in e-learning.

In an e-learning platform, there are different types of learners. For example, we can find learners that prefer learning through tutorials, other learners can prefer using videos for learning new courses, and others can choose using questions/answers for reaching their goals. The difference can be shown also in the motivation of learners, in their skills, and

also in their ability to master new concepts. With this difference between learners, the e-learning platforms must find new optimal approaches to take into account all the learners' preferences, and also for augmenting the quality of learning.

Recommender systems (RS) help people to find products and services which match their preferences and needs (Jang et al. 2019) and to reduce the amount of time they spend to find the items they are looking for. They are becoming increasingly important in a range of applications, such as e-commerce, music, film and book recommendation, web search, health and e-learning platforms (in the teaching process).

RS can be roughly divided into several types such as knowledge-based or content-based recommendation, Social filtering recommendation, collaborative filtering recommendation, and hybrid recommendation.

Social filtering (SF) recommendation uses the social content of users based on their tracks in different social networks, to recommend relevant content that match their social profiles. On the other hand, Collaborative filtering (CF) recommends rating or product for a user based on the rating preferences of similar users. CF is based on the assumption that users with similar taste have similar preferences to products or items, it is divided into a memory-based approach and model-based approach. In the model-based CF, models

✉ Youness Madani
younesmadani9@gmail.com

¹ Faculty of Sciences and Technics, Sultan Moulay Slimane University, Beni Mellal, Morocco

² Faculty of Science and Technics, Hassan II University of Casablanca, Mohammedia, Morocco

are developed using different algorithms (neural network, machine learning algorithms...) to predict and recommend relevant products or services. On the other hand, Model-based CF approaches based on the calculation of the similarity between users or items. they are divided into user-based CF and item-based CF.

In our work, each course has many pedagogical objectives(PO), and each PO has several concepts. our goal is to predict and recommend the suitable concepts for the learner(the concepts that the learner must learn). for example, the java course can have different pedagogical objectives (the installation of java/the basics of java/control structures/the tables is a PO; and installing JDK and JVM-variables-conditions and if statements-loops-switch statement-tables: are its concepts. Objects/the classes/Inheritance is another PO).

The ultimate goal of our approach is to recommend items that better match the learner's profile. Items in our case are the concepts that represent the pedagogical objectives of a course, for example, Java, Python or Php courses.

Our approach emerged from the idea that recommendation performances have a huge impact on teaching success. We use RS to estimate the potential preferences of learners and recommend relevant courses(or pedagogical objectives) for learners based on social content(using social networks) and the profile's content.

The proposed RS approach is divided into two parts:

- Social filtering (SF) approach: this step consists of using the social network's profiles of each learner for calculating two factors which we have called the **productivity** and the **motivation** of learners. In this step, we use the assumption that people with similar productivity and motivation will have similar profiles and skills in learning. This step will help us to find the K-nearest neighbors learners (similar learners).
- Collaborative filtering (CF) approach: In this step, we first construct a profile for each learner based on his knowledge, which means that after the construction of the profile we will find the rating of the learner for each PO. After that, we calculate the similarity between the active learner and the other learners, for recommending the PO of the most similar learner(closest learner). CF looks into the profiles of other learners to find the PO that is most suitable for the profile of the target user.

Our approach can deal with the traditional problems of recommender systems which are: the cold start problem, sparsity, and scalability, using the social content, the profiles of the learner and the similarity, for avoiding any problem that can affect the quality of recommendation.

In some cases, our e-learning system doesn't find all the necessary information for completing the recommendation's

work, which affects negatively the quality of learning. To remedy this problem, we propose a new reinforcement learning approach for finding the optimal way on which the learner must base for learning new concepts.

Reinforcement learning (RL) refers to a class of machine learning problems, whose purpose is to learn, from successive experiences, what needs to be done to find the best solution. In such a problem, we say that an "agent" (the algorithm, in the sense of the code and the variables it uses) interacts with the "environment" to find the optimal solution. Reinforcement learning differs fundamentally from supervised and unsupervised problems by this interactive and iterative side: the agent tries several solutions (**exploration**), observes the reaction of the environment and adapts its behavior (the variables) to find the best strategy (he "**exploits**" the result of his explorations). RL consists of letting the algorithm learn from its own mistakes, to learn how to make the right decisions.

In this work, RL comes to remedy the problems of recommendation systems especially the problems of sparsity and cold start. RL helps a learner to find the best way to learn based only on his knowledge, it gives to each learner in our platform the ability of adaptation using his knowledge and the content of different pedagogical objectives to find in the end the concepts that the learner must learn, and of course improving the quality of learning. With the use of RL, we based on the hypothesis that: intelligence is the ability to adapt.

The rest of this paper is organized as follows: Sect. 2 presents a literature review of the existing approaches, Sect. 3 describes the different steps of our proposed method, in Sect. 4 we going to present an implementation of the proposed approach. And finally, in Sect. 5 the conclusion with some perspectives is drawn.

2 Literature review

In recent years a lot of approaches have been developed for improving the quality of recommendation and avoiding the problems related to cold start, sparsity, and scalability. The field of application of RS has also become widespread, it moved from the field of e-commerce (recommending products) to other fields (Madani et al. 2017b) such as recommending films, services or also in the domain of education for recommending courses(e-learning platform).

Ar et al. in their article (Ar and Bostanci 2016) proposed a new CF approach based on genetic algorithm (GA) for improving the result of prediction with the use of different similarity measures. The Proposed solution aims to refine and improve the similarity values, i.e. weights, obtained using various metrics with GA so that the prediction accuracy of k-neighbour CF will be improved.

This paper presented an approach based on GA to reduce the mean absolute error MAE values for various metrics including Pearsons Correlation Coefficient PCC, EJC and vector cosine similarity (VCS). Results show that the evolutionary approach has significantly reduced the prediction error using the evolved weights and Vector Cosine Similarity has shown the best performance.

Authors of Koohi and Kiani (2017) proposed a new CF method to find neighbors users based on a user-based CF. They used the pearson correlation coefficient (PCC) as a traditional similarity measure to find neighbor users, and the k-means clustering method and the non-negative matrix factorization model (NNMF) for clustering as traditional clustering methods. First of all The proposed method tries to use three subspaces of items that users rated before. These three subspaces consist of a sub-space for interesting items, a subspace for neither Interested nor Uninterested NIU items and a subspace for uninterested items. For the experiments, Three well-known datasets are used on this paper to analyze the proposed approach: Movielens 100 k (ML_100k), Movie- lens 1 M (ML_1M) and Jester. Results show that the proposed method outperforms the other approach at the level of accuracy recall and precision and that clustering algorithms perform better than the similarity measures in finding neighbor users.

Liu et al. (2014) presented a new user similarity model to improve the recommendation performance when only a few ratings are available to calculate the similarities for each user. In this paper, the authors showed the drawbacks of some similarity measures such as cosine Pearson. The proposed similarity measure is a non-linear function in our model. which is the sigmoid function. The main objective of this approach is to improve the proximity-impact-popularity (PIP) similarity. Several experiments are conducted on three popular used data sets. And as experimental results, they see that the novel similarity measure can obtain better performance than most other methods. These results demonstrate the effectiveness of the novel similarity measure and it can overcome the drawbacks of the traditional similarity measures.

Authors in Chen et al. (2016) proposed a new CF similarity approach by improving the traditional adjusted cosine similarity algorithm. The user similarity calculation process can run offline, so you can reduce the recommended running time and improve the speed of recommendation, which solves the problem of the real-time for recommending. The proposed CF approach is based on optimized user similarity, A balancing factor is added to the traditional cosine similarity algorithm, which is used to calculate the project rating scale differences between different users. Experimental results show that the proposed improved collaborative filtering algorithm based on

user similarity can significantly optimize the accuracy of user similarity and get better recommendation results.

In the study of Nayebedeh et al. (2017), authors implemented two approaches which are collaborative filtering (CF) and social network recommendations system (SNRS). They used mean absolute error (MAE) and accuracy to compare the result of two mentioned approaches and found that the SNRS method as it is claimed to be an improved version of CF works more efficiency. The improved version which is known as Social Network Recommender System has better efficiency in comparison with traditional collaborative filtering, because, in this approach users preference, items general acceptance and influences from friends have been taken into consideration.

Wang et al. (2017) proposed a hybrid approach to evaluate the user similarity comprehensively and objectively. In this paper, the authors propose a new user similarity scheme by a hybrid method, which considers the influence of all possible rated items, the non-linear relationship between variables, the asymmetry between users, and the rating preference of users. They used a similarity measure that does not depend on the co-rated items between two users and can make full use of all rating information. According to the problems of the most used similarity measures such as co-rated problem, symmetry problem and flexibility problem authors proposed new similarity measures that can deal with those problems.

The study of Ren and Wang (2018) proposes a support vector machine (SVM) based collaborative filtering (CF) service recommendation approach, namely SVMCF4SR. The goal of SVMCF4SR is to correctly rank the items rather than to accurately predict their ratings. SVMCF4SR deals with the issues of scalability(SVMCF4SR can filter the services which may not be preferred by the active user, so that the resolution space can be reduced, and the recommendation's time is then shortened.) and sparsity (SVMCF4SR is effective and efficient no matter the historical rating data is sparse or dense, particularly, it still works well when the data is extremely sparse.). Authors use the principle of SVM to obtain a separating hyperplane which is the classifier for the active user. Using the classifier, the unknown services can be divided into two groups: services that are probably preferred by the active user, namely positive group; services which are unlikely to be preferred by the active user, namely negative group. As experimental results, the time efficiency of SVMCF4SR is higher than the prediction-based method. Moreover, SVMCF4SR has relatively high recommendation precision even if the rating matrix is extremely sparse.

The research of Chen et al. (2015) has adopted the neighbor based approach by applying the artificial immune system (AIS) in collaborative filtering for movie recommendation. The authors employed an artificial immune algorithm to train a set of immune networks. The rating data was treated as antigens, and a number of immune networks were

generated by copying the antigens as the antibodies of the immune networks. A modified similarity estimation formula based on the Pearson correlation coefficient is also developed. The authors compared their method to some state of the art techniques in terms of mean absolute error, precision, and recall. they applied the proposed approach to some well-known datasets. As experimental results, the approach produces very high precision and recall for these datasets. Although the proposed system still suffers some memory-based approach problems, such as cold start and data scalability.

In Bobadilla et al. (2009), the authors used collaborative filtering to an E-learning system. They add the knowledge level of the users in the collaborative filtering as additional characteristics, and they proposed a new metric to evaluate the proposed method. Although the authors are not used an e-learning dataset to evaluate their system they prove that the proposed metric outperforms some other metrics such us: the mean absolute error, mean squared error.

In the study of Wei et al. (2017), the authors propose two recommendation models to solve the complete cold start (CCS) and incomplete cold start (ICS) problems for new items. The authors proposed a hybrid approach based on CF and machine learning to improve recommendation performance for CS items. They use a deep learning approach(deep network: stacked denoising autoencoder **SDAE**) to extract items features from the content descriptions and used with a CF model (timeSVD++ model.) for cold-start item rating estimation. To predict ratings for CCS items, they predict the ratings for CCS items from their *M* most similar non-CS items. As experimental results, the authors used the large real-world dataset created by the Netflix Prize to evaluate the proposed models. The results showed that their models outperformed existing baseline approaches for cold-start item recommendation at the level of the recommendation prediction error RMSE.

Authors of Zhang et al. (2019) proposed a social recommendation algorithm based on stochastic gradient matrix decomposition in social networks. constructed a matrix with the social network and the user scoring matrix, and proposed a stochastic gradient descent algorithm for matrix factorization. Results showed that the proposed approach outperforms other state-of-the-art methods.

The use of reinforcement learning demonstrates its power to solve many real problems and applications in order to find and make better decisions. Its use in scientific research knows these recent years an explosive increase.

The study of Balasubramanian Velusamy and Abraham (2013) aims to provide mathematical modeling for an adaptive e-Learning environment using SARSA algorithm, by relating it to the concept of Reinforcement Learning. For that authors propose a number of states and actions that the agent(learner) can take to optimize the teaching process.

Authors demonstrate that SARSA algorithm suits the situation much more than Q-learning algorithm mainly because of its selection policy.

Fabiano et al. (2013) present an automatic, dynamic and probabilistic approach for modeling students learning styles (LS) based on reinforcement learning. Three different strategies for updating the student model (SM) are proposed and tested through experiments. The results obtained are analyzed, indicating the most effective strategy. In the proposed approach, students LS are stored as probability distributions in the SM, indicating the probability of preference for each LS. Experiments have shown that the proposed approach can automatically detect and precisely adjust students learning styles, based on the non-deterministic and non-stationary aspects of learning styles, and also the efficiency and effectiveness of the proposed approach.

The work of Raghuvveer et al. (2014) proposes reinforcement learning-based algorithm to analyze the learner information (derived from both implicit and explicit feedback) and generate the knowledge on the learners requirements and capabilities inside a specific learning context. The reinforcement learning system (RILS) implemented as a part of this work utilizes the knowledge thus generated in order to recommend the appropriate learning objects (LOs) for the learners. For that, the authors used the constructivist reward-based learning (CRBL) algorithm. The results have thus proved that the CRBL algorithm was able to effectively drive the distinctive state values of the learners to reinforce them with appropriate LOs.

In Tizhoosh et al. (2007), the authors focus on the investigation of the user-machine interface and the complexity of a dynamic environment like an e-learning application based on reinforcement learning. They present the techniques for developing user-oriented RL agents and discuss design requirements and limitations. The techniques for presenting the states and actions and defining the objective and subjective reward are introduced as well. Some hybrid systems using reinforcement learning techniques are provided.

Iglesias et al. (2003) studied the performance of a Reinforcement Learning model in a database design (DBD) AIES, where this performance is measured on several students required to acquire efficient teaching strategies. In this paper, authors eliminate the pedagogical strategy concept using a knowledge representation based on a reinforcement learning (RL) model that allows adaptive and intelligent educational systems (AIES) to adapt tutoring to students needs, optimally sequencing the content based on the students performance, lesson objectives and the relationships between course modules. Authors did experiments with some parameters of the learning algorithm to study their effects at the convergence of the system and, at the same time, we have experimentally shown that AIES can learn an optimal policy to teach students interacting with reasonably

few students. This makes possible its implementation in AIES with human students.

The paper of Javadi et al. (2012) improves the students behavior model in a tutorial-like system. In the proposed method, the student model is determined by high-level learning automata called level determinant agent (LDA-LAQ), which attempts to characterize and improve the learning model of the students. LDA-LAQ uses learning automata as a learning mechanism to show how the student is slow, normal or fast in the term of learning. This paper shows the new student how the learning model increases speed-accuracy using pursuit learning automata and Reinforcement Learning.

Wen-Jye Shyr in his paper (Shyr 2007) presents the development of the e-learning platform for reinforcement learning on temperature sensor experimental module based on the graphical monitoring and control system.

Our proposed approach in this work takes advantage of the notions of recommendation systems for proposing a new e-learning platform to improve the quality of learning. Unlike the proposed approaches that are based only on the profiles of learners for finding the optimal pedagogical content, our work in addition to the content of the learners' profiles uses the social content based on social networks. Our approach also uses the social interactions of the learner with his environment. Unlike existing methods that rely solely on the knowledge of learners, our method makes the adaptation based also on their productivity and motivation. The proposed method is based on a new idea to group learners (clustering) with the use of data from social networks which will help us avoid the problem of scalability. The proposed approach also uses CF principles to increase the quality of recommendation based on learners' knowledge. To avoid the problems of sparsity and cold start that the majority of recommendation systems suffer from it, we propose a new approach based on RL for finding the optimal way that the learner must take to find the best pedagogical content.

3 Proposed approach

Looking closely at web content, we remark that the number of learners that started learning online grows explosively. In addition to the number of courses published online. Without forgetting that each learner has his own rhythm of learning (every learner has a unique characteristic called its learning style) which depends on his skills and his ability to learn. With that, it becomes necessary to use the recommender systems for finding the optimal and relevant courses for learners.

As presented earlier, this work consists in proposing a new recommendation approach for recommending optimal courses to learners based on their profiles and their social

contents, to improve the teaching process and to reduce the amount of time they spend to find relevant courses.

Our proposed RS is based on two new proposed approaches:

- *Social filtering approach (SF)* that uses the social content of learners based on their social network's profiles, to define some factors like **productivity** and **motivation** of learners. These factors will be used as parameters for clustering (placing learners with similar social content in clusters).
- *Colaborative filtering approach (CF)* which consist in constructing learners' profiles based on their knowledge, and then calculating the similarity between the active learner and the other learners in the RS, or between the active learner and the different concepts of a courses' pedagogical objectives, based on a reinforcement learning approach.

we named our proposed approach: social collaborative filtering (SCF) because it uses both advantages of SF and CF for finding optimal relevant courses for learners (a hybrid approach between SF and CF). SCF consist of recommending relevant courses to the active learner based on the profiles and social content of similar learners.

Our social collaborative filtering approach looks for learners that share similar preferences with the target learner in terms of social and pedagogical content, and recommend the concepts of the most similar learner (closest learner). That is, recommend the concepts that the closest learner learned before in our platform. We based here on the assumption that says: "learning is active, learning is a social activity, learning is contextual, learning takes time, and learning needs motivation".

Our method takes into account also the learning styles of learners. As learners with different skill sets use our system. Some learners may be fast learners while some may be slow, some may need to practice more problems while others may need just example. from that, we proposed a new reinforcement learning approach.

In our RS each course is presented as a set of pedagogical objectives (PO), and each PO has its concepts. For example, for the PO "basics of java" of the java course, **Installation-Variables-Data Types-Conditionals and Control Flow-Arrays** are its concepts. The idea is that the learner must learn the concepts which are relevant to his profile and his social content.

3.1 Social filtering approach (SF)

SF is the first step in our approach. It uses the learners' social network profiles (Facebook and Twitter) to collect information such as published posts or tweets, likes, and

comments. These informations will be used after to calculate/define two measures that we named **productivity** and **motivation**.

The productivity define the period of the day in which the learner is more productive(publish a lot of content), for that we define three periods : (1) from 8 a.m. to 12 p.m., (2) from 2 p.m. to 6 p.m. and (3) from 7 p.m. to 11 p.m. The idea is to calculate the number of posts/tweets published, liked or commented by the learner in each of these three periods. And after the productivity is for the period with maximum posts/tweets. Formula 1 and 2 show how to calculate the productivity.

$$NPT = \text{Max}[Card(P1/T1), Card(P2/T2), Card(P3/T3)] \tag{1}$$

$$Productivity = \begin{cases} 1 & \text{if } NPT = Card(P1/T1) \\ 2 & \text{if } NPT = Card(P2/T2) \\ 3 & \text{if } NPT = Card(P3/T3) \end{cases}$$

where:

- NPT : is the number of posts (Facebook)/tweets (Twitter).
- $Card(P1/T1), Card(P2/T2), Card(P3/T3)$: is respectively the number of posts/tweets published, liked or commented in the period 1, 2 and 3.
- $Max[Card(P1/T1), Card(P2/T2), Card(P3/T3)]$: return the maximum number after the calculation of the cardinality in each period.

After all that, the productivity is equal to 1 if the learner is more productive in the period 1, equal to 2 if he is more productive in the period 2 and equal to 3 if he is more productive in the period 3.

The Motivation consists in defining the motivation of each learner either he is motivated, demotivated or neutral. For that and after we define the productivity, we collect the posts (from Facebook) and tweets (from Twitter) published in the period of activity(the period in which the learner publish and tweet a lot of content).

The next step, is to classify the retrieved posts and tweets into three classes: **positive**, **negative** or **neutral**. that is to say, each retrieved tweet or post can either express a positive sentiment, negative sentiment or neutral sentiment. Then if the majority class (MC) is the **positive** class, we consider the learner as motivated, else if MC is the negative class, we consider the learner as demotivated, and if MC is the neutral class we consider the learner as neutral.

For making the classification, we use a new method that classifies a tweet or a Facebook post (FP) into three classes(positive, negative or neutral). The proposed method is based on two approaches.

- The first approach is based on semantic similarity (SS) which consists in calculating the degree of similarity between each tweet or FP to classify and two words (Madani et al. 2019a) “Positive” and “Negative”. The first step of this approach is the application of different text preprocessing methods in order to extract opinion words, and after by using the Leacock and Chodorow (Leacock and Chodorow 1998) approach and the Word-Net dictionary we calculate the semantic similarity between each opinion word of the tweet/Facebook post to classify and the words “**positive**” or “**negative**”. All these steps will help us in calculating the *positivity* and the *negativity* of each tweet/Facebook post as presented in the following formulas.

$$Positivity = \sum_{i=1}^N Sim_{LC}(W_i, positive) \tag{2}$$

$$Negativity = \sum_{i=1}^N Sim_{LC}(W_i, negative) \tag{3}$$

with:

- N: Is the number of words in the tweet or the Facebook Post.
- W_i : is the opinion word number i in the tweet.
- $Sim_{LC}(W_i, positive)$: Is the semantic similarity using the **Leacock and Chodorow** approach between the opinion word i of the tweet/Facebook post and the word “positive”.
- $Sim_{LC}(W_i, negative)$: Is the semantic similarity using the **Leacock and Chodorow** approach between the opinion word i of the tweet or the Facebook post and the word “negative”.

- The second approach is based on the concepts of fuzzy logic (FL). We propose a new fuzzy logic system (FLS) that takes into account the fuzziness and the vagueness of sentiments. For that the positivity and the negativity calculated earlier will play the roles of inputs of our FLS, and by applying the different FL steps (Fuzzification using the trapezoidal membership function, fuzzy inference rules using the approach of Mamdani, and defuzzification using the centroid method) we find in the output the class of the tweet/Facebook post (positive, negative or neutral).

$$CV = FLS(Positivity, Negativity) \tag{4}$$

- CV: Classification value. That is, the class of the tweet/Facebook post (positive, negative or neutral).

Figure 1 shows the different steps of our FLS applied to the positivity and the negativity (taking into account the

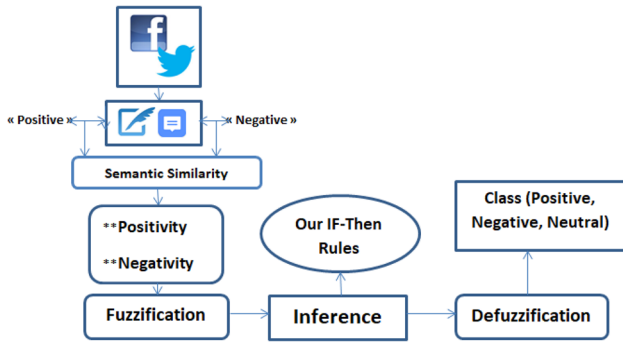


Fig. 1 Motivation steps

fuzziness of sentiments) for finding in the output the class of a tweet or a publication/comment Facebook.

The step of **fuzzification** consists of calculating the degree of belonging of the inputs to each fuzzy set. In our FLS each input can belong to three fuzzy sets: low, high or moderate. We use the trapezoidal membership functions for the fuzzification (Madani et al. 2019b).

For the inference rules, we created nine rules (IF–THEN) based on the system of Mamdani (Mamdani and Assilian 1975) as the following:

- **IF** Positivity is low **AND** Negativity is low **THEN** Class is Neutral.
- **IF** Positivity is moderate **AND** Negativity is moderate **THEN** Class is Neutral.
- **IF** Positivity is high **AND** Negativity is high **THEN** Class is Neutral.
- **IF** Positivity is low **AND** Negativity is moderate **THEN** Class is Negative.
- **IF** Positivity is low **AND** Negativity is high **THEN** Class is Negative.
- **IF** Positivity is moderate **AND** Negativity is high **THEN** Class is Negative.
- **IF** Positivity is moderate **AND** Negativity is low **THEN** Class is Positive.
- **IF** Positivity is high **AND** Negativity is moderate **THEN** Class is Positive.
- **IF** Positivity is high **AND** Negativity is low **THEN** Class is Positive.

After the application of these rules, we find in the output the degree of belonging of the output (class or sentiment) to each output fuzzy set (positive, negative and neutral).

At the end by the application of the defuzzification step, we find the final result (the class of the tweet or the publication Facebook).

After the classification of the tweets and the publications/comments on Facebook, the final step consists of defining the motivation of the target learner. For example, if the

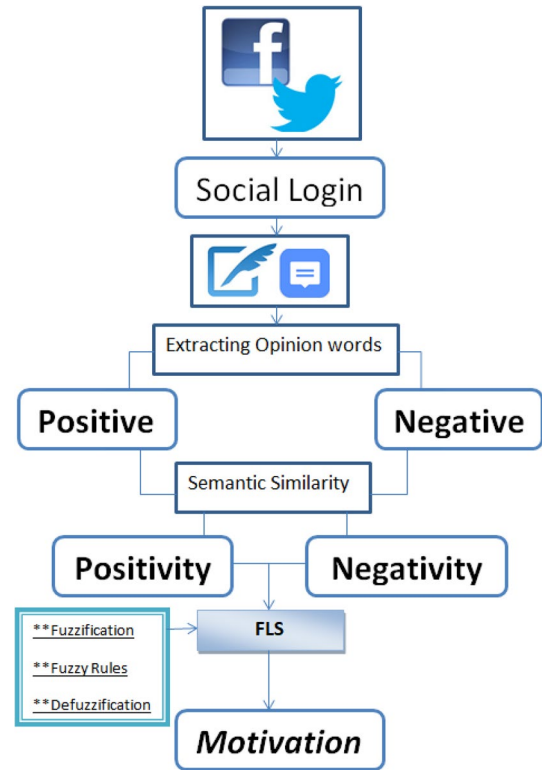


Fig. 2 Motivation steps

majority of the tweets and the publications are positive, the learner is considered as motivated.

The formula 4 shows how to define the motivation of learners.

$$Motivation = \max \left[\sum_{i=1}^p CV(T_p), \sum_{i=1}^n CV(T_n), \sum_{i=1}^{ne} CV(T_{ne}) \right] \tag{5}$$

with:

- $\sum_{i=1}^p CV(T_p), \sum_{i=1}^n CV(T_n), \sum_{i=1}^{ne} CV(T_{ne})$ are respectively the sum of the positive, negative and neutral posts/tweets.

Figure 2 shows how we define the motivation of a learner.

Before the definition of motivation, two important steps make the tweets ready for classification: The first step is the extraction of the tweets and Facebook posts published in the period of activity, in this step we used the Twitter4J API.¹

¹ <http://twitter4j.org>. Twitter4J is an unofficial Java library for the Twitter API. With Twitter4J, you can easily integrate your Java application with the Twitter service. Twitter4J is an unofficial library.

The second step consists of applying the different text pre-processing methods on the extracted tweets/Facebook posts or comments for preparing them to the classification and also for extracting the opinion words. In this step, we used the Apache OpenNLP library.²

These two measures (productivity and motivation) will be used after for clustering learners into clusters. Because the productivity and the motivation have both three possible values (1, 2 and 3 for the productivity; positive, negative and neutral for the motivation), we have 9 different clusters or classes (1-positive, 1-negative, 1-neutral, 2-positive, 2-negative, 2-neutral, 3-positive, 3-negative and 3-neutral).

At the end of the SF approach, we will classify learners into clusters based on social content. And for each new learner (target learner) for whom we want to recommend a course (pedagogical objective), we apply the different steps of SF to calculate the productivity and the motivation which will help us after in the clustering, for finding the K-nearest neighbors of the target learner (learners who share the same productivity and motivation with the target learner (belongs to the same cluster)).

3.2 Collaborative filtering approach (CF)

CF is the second step in our work. In our platform, each learner is presented with his profile of knowledge, and for constructing this profile each learner must complete a quiz which is in the form of multiple-choice questions. Each course in our system has its equivalent quiz. The questions of the quiz represent the different concepts of the pedagogical objectives (PO) of a course, that is to say, each concept represents an item in our platform, and doing the quiz is like giving a rate to each concept.

After that the learner completes the quiz and based on the result obtained, we construct his profile which is in the form of a vector. Each element of the vector represents a rating of a concept. The rating is equal to 0 if the learner answered correctly the equivalent question of the concept, and equal to 1 otherwise.

We assume that if the learner did not answer a question correctly, that means he must learn the concept related to this question, and we consider his rating as 1. Otherwise, if he answered a question correctly, he is already mastering the related concept, so it is not necessary to learn it again, for that we consider his rating as 0.

After we find the K-nearest neighbors (KNN) of the target learner using the social filtering approach (based on productivity and motivation), and after we construct his

profile (which is in the form of vector) based on the quiz, we calculate the similarity between the profile of the target learner and the profiles of the KNN learners to find the most similar learner (the closet learner). We based here on the assumption that the target learner and his closest learner have similar preferences and can learn similarly. From all that, at the end of this process, we recommend to the target learner the concepts that the closest learner already learned in our platform.

For calculating the similarity between learners, we use the cosine similarity.

3.2.1 Cosine similarity

The cosine similarity (Madani et al. 2018) calculates the similarity between two n-dimensional vectors by determining the cosine of the angle between them. This metric is frequently used in text mining. It uses the complete vector representation, that is to say, the objects' frequency (words). Two documents are similar if their vectors are combined. If two objects are not similar, their vectors form an angle (X, Y) whose Cosine represents the similarity value. The formula 6 is defined by the ratio of the scalar product of vectors X and Y and the product of the norm of x and y.

$$Sim_{cos}(X, Y) = \frac{\sum_{i=1}^n x_i \cdot y_i}{\sqrt{(\sum_{i=1}^n x_i^2)} \cdot \sqrt{(\sum_{i=1}^n y_i^2)}} \quad (6)$$

Each element of the vectors X and Y present the rating of a concept (0 or 1).

In our system, the profiles of the learners are represented by the following notation.

$$Profile = [R(C_1), R(C_2), R(C_3), \dots, R(C_n)]$$

where :

- C: is a concept.
- R(C): is the rating of the concept after doing the quiz (1 or 0).
- n: is the number of concepts in the different pedagogical objectives of a course.

3.2.2 Example of the construction of learners' profiles

We assume that a learner wants to learn the java course. First of all, he must do a quiz which is in the form of multiple-choice questions, where each question represents a concept. Suppose that java course has 4 PO:

- PO_1 : installation-Variables and Types-Conditionals-Loops-Arrays

² <https://opennlp.apache.org/>. The Apache OpenNLP library is a machine learning-based toolkit for the processing of natural language text.

- PO_2 : Functions-Objects-classes-Compiling and Running with Arguments
- PO_3 : Inheritance-Try and Catch-Abstract Classes-Polymorphism-Interfaces
- PO_4 : Using Generics-Collections-Exceptions-Generic Types-IO Streams

Based on the structure of these 4 POs, the java course has 19 concepts, which means that the quiz will have 19 question, and that the vector that represents the learner will have 19 elements with the values 0(the learner master the equivalent concept) or 1 (the learner did not master the equivalent concept).

Suppose that after that the learner does the quiz, he answered the questions equivalents to the concepts (**installation, Variables and Types, Conditionals, Classes, Inheritance, Polymorphism, Collections, Exceptions, Generic Types**) correctly, and he answered the questions of the other concepts incorrectly. So the final vector of the learner will have the following structure :

$Profile = [0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1]$

3.3 Recommender systems' challenges: scalability and sparsity

The recommender systems have some challenges related to the problems of **scalability** and **sparsity or cold start**, and each RS approach must take into account these two problems (dealing with them) for improving the quality of recommendation.

Our proposed approach can deal with these two problems, which helps us in avoiding any issue that can affect our system.

3.3.1 Scalability

The first main challenge faced by learners based on our social collaborative filtering approach is the problem of **scalability** as it can be shown when searching for the closest learner if our platform contains a very large number of learners and courses. to avoid this problem we adopt two levels:

- The first level consists of clustering the learners into clusters based on the social content(productivity and motivation) for decreasing the number of learners (conserving only the most similar learners to the target learner). After that we do not need to search for the closest learner into all the learners of our platform, but only into the adequate cluster of the target learner (the cluster that contains similar learners who have similar social content with the target learner).

- The second level consists of parallelizing our work using big data technologies (Hadoop MapReduce and Hadoop Distributed File System). The idea is that we share the work of recommendation between several machines, either in the phase of extraction of social content and in the phase of storage, or in the phase of the calculation of the productivity/Motivation and the cosine similarity.

3.3.2 Sparsity and cold start

The second main challenge in our system is related to the **sparsity and cold start** problems. They occur when it is not possible to make a reliable recommendation due to an initial lack of information (similar learners of the target learner). For example, for the first user of our platform, it is impossible to recommend courses based on social content because there are no learners in the system. or if we do not find any similar learners for the target learner.

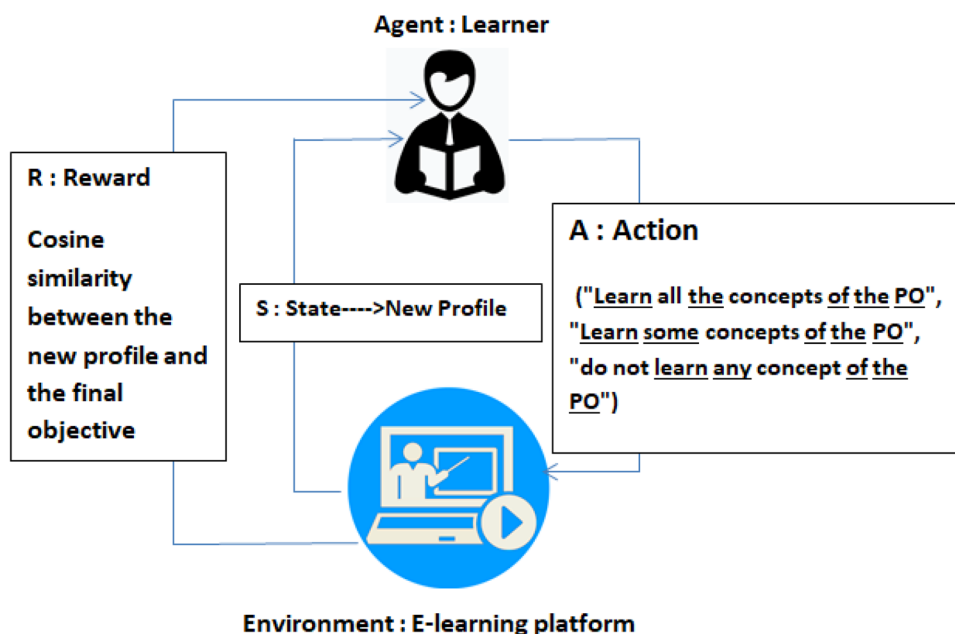
The cold start and sparsity problems occur in our approach on three levels. The first is when the first user wants to use our platform to have a recommendation. The second level which we can call complete cold star (CCS) occurs when no KNN learner (of the target learner) is available in our platform (no learner has similar social content with the target learner). And the third level is when the similarity between the profile of the target learner and the profiles of the learners of his cluster is less than a threshold, which means that among the similar learners no one has a similar profile with the target learner.

If we encounter one of these two problems (sparsity or cold start), we will be based only on the profile of learners and the concepts of the pedagogical objectives for recommending the most suitable concepts. The idea is that after the construction of the profile by completing the quiz we construct the profile's vector with a rating for each concept. And after, we will look for the optimal concepts that the learner must learn.

For that, we propose a new reinforcement learning approach based on the learners' profiles. Our idea is that each learner has his own learning styles, skills, and motivation, so we need to adapt the learning process based on the learner's profile (find the optimal path that the learner must follow to reach the goal optimally).

Reinforcement learning (RL): Reinforcement learning (RL) is an intelligent technique with the ability to learn from interaction with the environment. It learns from trial and error and generally does not need any training data or a user model. At the beginning of the learning process, the RL agent does not have any knowledge about the actions it should take. After a while, the agent learns which actions yield the maximum reward. The problem consists of an agent with its various states S and a set of actions per state A. The agent can move from one state to another by performing

Fig. 3 Agent–environment interaction



some action a , the next state gives a reward to the agent. The goal of the agent is to maximize the total reward.

If we compare all that with our proposed RL approach, The learner is the agent, and the environment is our e-learning platform with a course and its pedagogical objectives. The ultimate goal for the learner is to find the optimal path to reach the final goal of the course. The initial state of the agent (learner) in our RL approach is his profile (his knowledge) after doing the quiz which is in the form of a vector (each element has either the value 0 or 1). And the objective (final state) is the mastery of all the concepts of a course, which means the profile's vector of the learner in which each element has the value 0 (the learner master all the concepts).

In our proposed RL system the states are the different POs of a course, and in each state the learner can take three different actions to move from one state to another in order to reach the final goal, these actions are: "Learn all the concepts of the PO", "Learn some concepts of the PO", "do not learn any concept of the PO". After that the learner chooses an action, his state will be changed, that is the vector of his profile will be changed, and the reward produced is obtained by calculating the cosine similarity between the new profile (new state) and the final objective (the final vector in which all elements have the value 0). In each state, the goal is to choose the action that maximizes the cosine similarity with the final objective.

Figure 3 shows how the agent (learner) interact with the environment.

In each step and after the learner take an action, his state will be changed, which means that his profile's vector will be changed, and by using the cosine similarity we find the reward produced by taking this action.

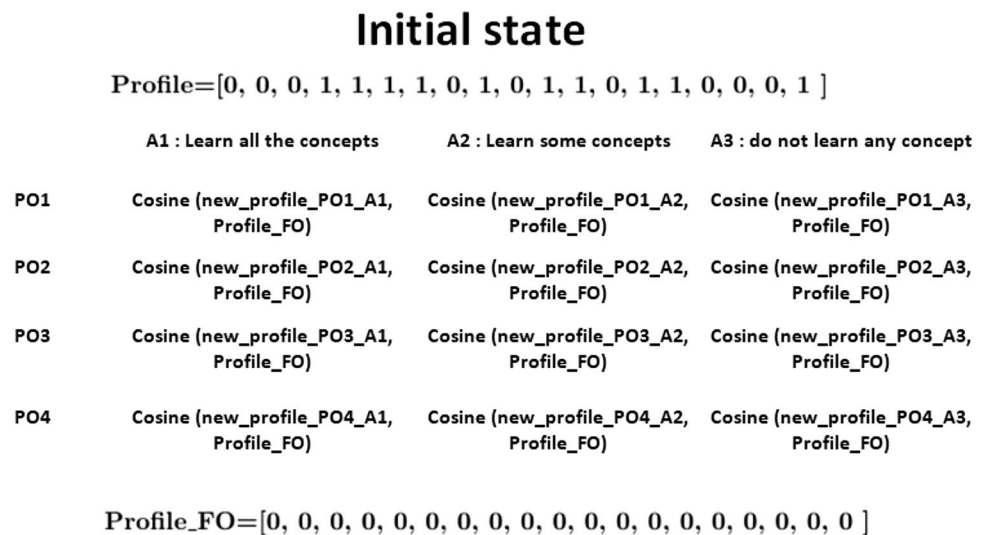
In our method, we start from an inefficient solution that is the profile of the learner after he passes the quiz, and that is improved progressively according to the experience of the agent (the learner). The learner will take actions ("Learn all the concepts of the PO", "Learn some concepts of the PO", "do not learn any concept of the PO") and the environment will give him a reward (cosine similarity between the new profile of the learner after he takes action and the vector of the final objective). The goal of our approach is to know what to learn from a number of pedagogical objectives to maximize the cosine similarity between the learner's profile and the final objective. So the learner will discover for himself the actions that give the highest rewards by trying them.

A learner experience using our approach is composed of 4 steps: a current state, take an action, next state (new state), and a reward. The value function or the update function of our approach is cosine similarity, this measure consists in associating with each possible state an estimation of the value for the learner to be in this state.

To fully explain our approach we will present how our RL approach works with an example.

A learner wants to use our e-learning platform for learning the java course, so the first thing to do is the extraction of his social contents for defining the productivity and the motivation in order to find the KNN learners (SF approach), the second step is the calculation of cosine similarity between the profile's vector of our target learner and the profiles of his KNN learners for finding the closest learner (CF approach). Suppose that after all these steps, we did not recommend the optimal content for the learner because of the problem of sparsity or cold start as

Fig. 4 Final structure of our approach



Final state : Final Objective

previously presented. In this case, we will be based on the proposed RL approach for giving to the learner the optimal content that must learn.

We based here on the example presented in Sect. 3.2.2, in which we assume that the course java has 4 POs.

The first step of our RL approach is the definition of the initial state of the target learner. The initial state in our approach is the initial profile’s vector of the target learner after he passes the QUIZ.

Profile = $[0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1]$

The second step is the definition of the final objective (FO) that the agent (learner) has to reach. In our RL approach the FO is that the learner master all the concepts of different POs of the java course, which means that the profile must be in the following form:

Profile_FO = $[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$

After the definition of the initial state and the final objective, we need to define the states and actions that the agent can take in our system. The states of our approach are **the profiles of the learner in each PO**, and the actions are three: “Learn all the concepts of the PO”, “Learn some concepts of the PO”, “do not learn any concept of the PO”.

Figure 4 shows the final structure of our approach.

For our RL agent(learner), the learning process has two components: exploration and exploitation. **Exploration** means that the agent tries to discover which actions yield the maximum reward by taking different actions repeatedly and in a random manner. In each action, we calculate the cosine similarity between the resulting profile and the profile of the final objective. **Exploitation**, on the other

hand, means taking the most rewarding actions. In other words, the learner follows the path with the high value of cosine similarity.

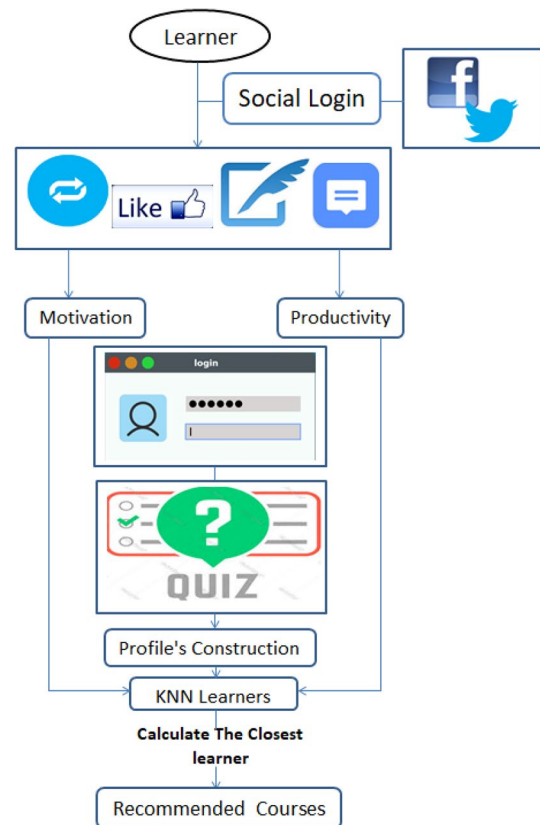


Fig. 5 Our work steps

At the end of our RL approach, we recommend to the target learner the optimal content. This content will play the role of his profile which will help us after in the recommendation process for future learners.

3.4 Our work steps and algorithm

As presented earlier, our approach is based on several important steps. The first is by using the social network profiles of learners for extracting some social factors

(Social login Madani et al. 2017a) like tweets, Facebook posts, likes, comments. These factors will be used in the calculation of the productivity and the motivation, which in turn will help us define the K-nearest neighbors. The second step is the construction of learners profiles based on a quiz. And finally, the calculation of the cosine similarity between the profile of the target learner and the profiles of the KNN learners for defining the Recommended courses (or Recommended pedagogical objectives).

Figure 5 shows the different steps of our work:

Algorithm 1 presents the algorithm of our approach.

Algorithm 1 Social Collaborative Filtering Algorithm

```

INPUT : Social Content
OUTPUT : Recommended Courses
M ← 0
D ← 0
N ← 0
Social_Login()
B1 ← Social_Factors(8, 12)
B2 ← Social_Factors(2, 6)
B3 ← Social_Factors(7, 11)
P ← Max(B1, B2, B3)
if P = B1 then
    Productivity ← 1
else if P = B2 then
    Productivity ← 2
else if P = B3 then
    Productivity ← 3
end if
Publications[] ← Extract_Posts_Tweets(Productivity)
for all Pub ∈ Publications do
    Class ← FLS(Pub)
    if class = "positive" then
        M ← M+1
    else if class = "negative" then
        D ← D+1
    else if class = "neutre" then
        N ← N+1
    end if
end for
Motivation ← Max(M, D, N)
if Motivation = M then
    The learner is motivated.
else if Motivation = D then
    The learner is demotivated.
else if Motivation = N then
    The learner is neutral.
end if
KNN_Learners ← Clustering(Learner's motivation, Productivity)
Closest_Learner ← Cosine(Target_Learner, KNN_Learner's )
Recommended_Courses(Closest_Learner)

```

where:

- $Social_{Login}()$: is a function that allows the learner login in our platform using their Facebook and Twitter accounts.
- $Social_Factors(8,12)$: retrieves the number of tweets, Facebook posts, likes or comments which are published between 8 am and 12 pm.
- B_1 : returns the number of posts, likes, tweets or comments published in the first period (from 8 a.m. to 12 p.m.).
- $Max(B_1, B_2, B_3)$: a function that returns the maximum of the three numbers(B_1, B_2, B_3).
- $Extract_{Posts_Tweets}(Productivity)$: this function extracts all the posts or tweets published by the learner in the period of activity.
- $FLS(Pub)$: this function classifies each tweet or Facebook post published in the period of activity into three classes(positive, negative or neutral). The classification is done using our proposed approach using the semantic similarity and the fuzzy logic system.
- Clustering (Motivation, Productivity): consists in clustering the learners into clusters for defining the K-nearest neighbors of the target learner based on the motivation and productivity.
- Cosine ($Target_Learner, KNN_Learners$): Calculate the cosine similarity between the profile of the target learner and the profiles of the KNN learners, for extracting the closest learner.

4 Experimental results and implementation

For demonstrating the power of the approach that defines the motivation, we did an experience which consists in classifying several tweets/Facebook posts using in a first time the semantic similarity and in a second time fuzzy logic+semantic similarity by calculating the classification and the error rate. Figure 6 presents the result obtained.

From Fig. 6, we remark that the two approaches (either by using a method based only on semantic similarity or by using it in combination with the fuzzy logic) give good result in term of classification and error rate, without forgetting that by enriching the classification with the fuzzy logic the classification rate is improved as 97%, and also the error rate is decreased from 7 to 3%.

Another experiment consists in comparing our approach with other techniques from the literature (an approach base on the AFINN dictionary, a dictionary-based approach

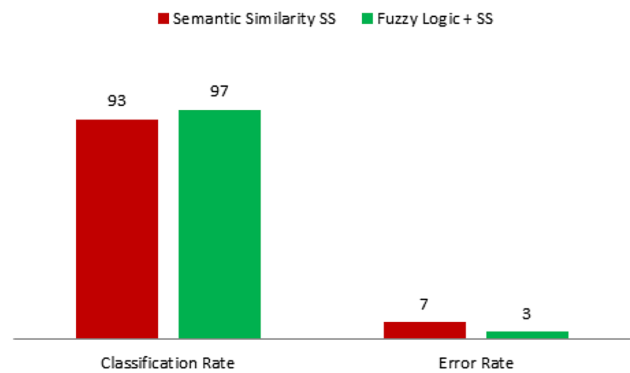


Fig. 6 Result of the experience 1

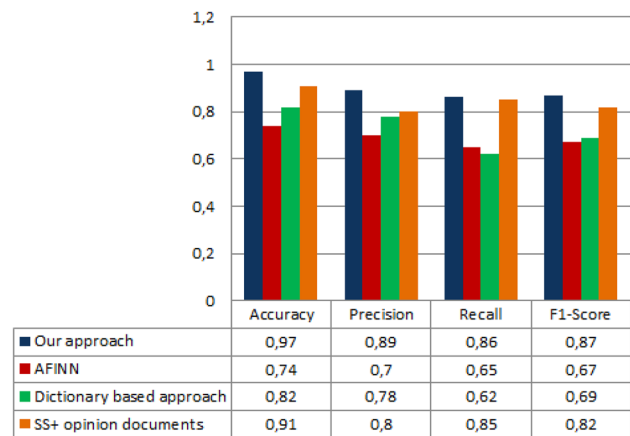


Fig. 7 Result of the experience 2

(Madani et al. 2017c), and an approach based on semantic similarity and opinion documents (positive, negative and neutre Madani et al. 2019a). For the comparison, we have based on the metrics of accuracy, recall, precision, and F1-Score. See Fig. 7.

Accuracy: the portion of all true predicted instances against all predicted instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Precision: the portion of true positive predicted instances against all positive predicted instances.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

Recall: the portion of true positive predicted instances against all actual positive instances.

Fig. 8 Social Login 1

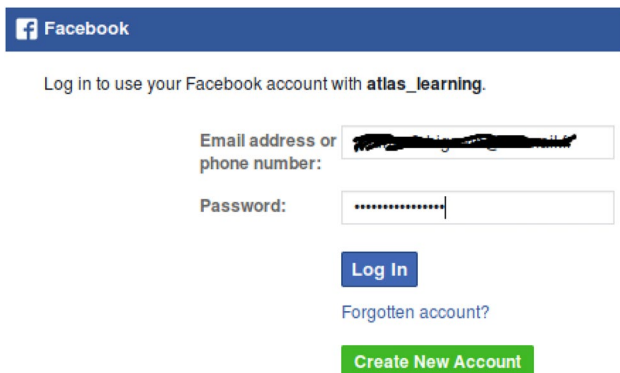
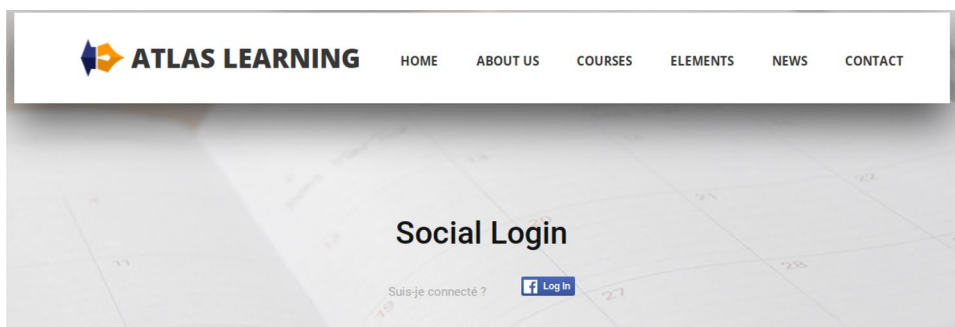


Fig. 9 Social Login 2

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

F1-score: a harmonic average of precision and recall.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (10)$$

where:

TP(True Positives): the correctly predicted positive values which means that the value of the tweet's class is positive and also the predicted value after the classification is positive.

TN(True Negatives): the correctly predicted negative values which means that the value of the tweet's class is negative and also the predicted value after the classification is negative.

FP(False Positives): When the tweet is negative but the predicted class after the classification is positive.

FN(False Negatives): When the tweet is positive but the predicted class after the classification is negative.

According to Fig. 7, our approach (based on fuzzy logic and the calculation of the semantic similarity between the

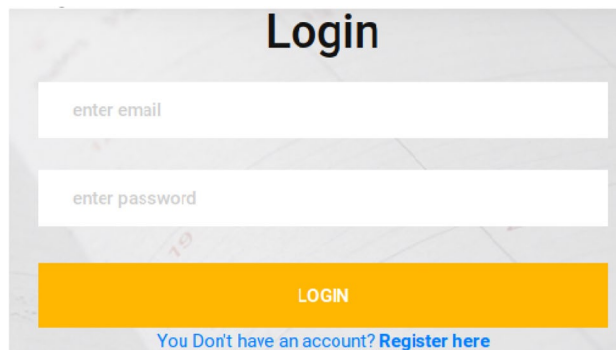


Fig. 10 Login page

tweet/Facebook post to classify and two words **positive** and **negative**) outperforms the other techniques at the level of accuracy (97%), precision (89%), recall (86%) and F1-score (87%).

4.1 Implementation

In this subsection we will show the implementation of different steps of our platform:

- The first step consists of log in using a social network (social login) for extracting the social content of the learner (Figs. 8, 9).
- After the extraction of the social network's data that will be used after in the step of clustering, the learner must log in our platform for doing a quiz (constructing the profile) (Fig. 10).
- The next step consists of choosing a course and doing the quiz (Figs. 11, 12).
- The final step is the recommendation of the optimal pedagogical objective for the target learner (Fig. 13).

Fig. 11 Construction of the profile 1



Fig. 12 Construction of the profile 2

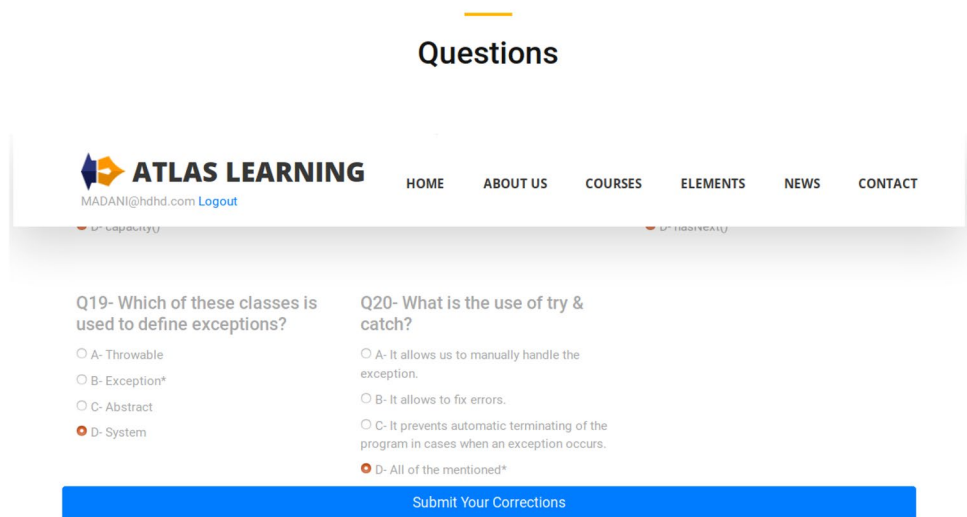
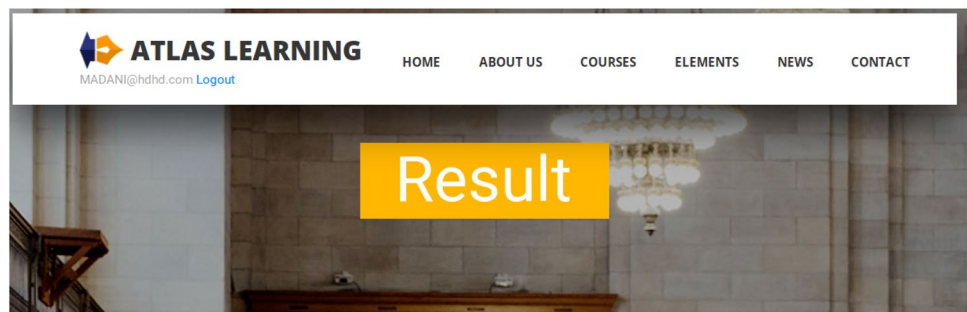


Fig. 13 Result of Recommendation



Java basis

5 Conclusion

In this paper, we have presented a new approach for recommending courses to learners based on their social contents and their profile (knowledge). Our method is a social collaborative filtering approach that uses the social content of learners such

as the tweets, facebook posts, likes and comments for grouping them into clusters, and also it is based on the knowledge of learners for finding the K-nearest neighbors learners of the target learner.

Our approach deals with the problems of scalability, sparsity and cold start for improving the quality of recommendation. To remedy the problems of sparsity and cold start, we

have proposed a new reinforcement learning approach for finding the optimal path with which the learner must learn to improve the results of the teaching process.

Our next work will consist in developing our approach and apply it in different domains like in an e-commerce website or in recommending films. and Also in proposing a new approach for calculating the similarity between the learners' profiles.

References

- Ar Y, Bostanci E (2016) A genetic algorithm solution to the collaborative filtering problem. *Exp Syst Appl* 61:122–128. <https://doi.org/10.1016/j.eswa.2016.05.021>
- Balasubramanian Velusamy SMA, Abraham G (2013) Reinforcement learning approach for adaptive e-learning systems using learning styles. *Inf Technol J* 12:2306–2314
- Bobadilla J, Serradilla F, Hernando A (2009) Collaborative filtering adapted to recommender systems of e-learning. *Knowl Based Syst* 22(4):261–265. <https://doi.org/10.1016/j.knosys.2009.01.008> (Artificial Intelligence (AI) in Blended Learning)
- Chen H, Li Z, Hu W (2016) An improved collaborative recommendation algorithm based on optimized user similarity. *J Supercomput* 72(7):2565–2578. <https://doi.org/10.1007/s11227-015-1518-5>
- Chen MH, Teng CH, Chang PC (2015) Applying artificial immune systems to collaborative filtering for movie recommendation. *Adv Eng Inf* 29(4):830–839. <https://doi.org/10.1016/j.aei.2015.04.005>. URL <http://www.sciencedirect.com/science/article/pii/S1474034615000518>, collective Intelligence Modeling, Analysis, and Synthesis for Innovative Engineering Decision Making Special Issue of the 1st International Conference on Civil and Building Engineering Informatics
- Dora FA, Lima LV, Fernandes MA, Lopes CR (2013) Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: An experimental analysis. *Exp Syst Appl* 40(6):2092–2101. <https://doi.org/10.1016/j.eswa.2012.10.014>
- Iglesias A, Martínez P, Fernandez F (2003) An experience applying reinforcement learning in a web-based adaptive and intelligent educational system. *Inf Educ* 2:223–240
- Jang YH, Park SC, Kim H (2019) Design and implementation of social content recommendation system based on influential ranking algorithm management. *J Ambient Intell Hum Comput*. <https://doi.org/10.1007/s12652-019-01275-5>
- Javadi SL, Masoumi B, Meybodi MR (2012) Improving student's modeling framework in a tutorial-like system based on pursuit learning automata and reinforcement learning. In: International Conference on Education and e-Learning Innovations, pp 1–6. <https://doi.org/10.1109/ICEELI.2012.6360564>
- Koohi H, Kiani K (2017) A new method to find neighbor users that improves the performance of collaborative filtering. *Exp Syst Appl* 83:30–39. <https://doi.org/10.1016/j.eswa.2017.04.027>
- Leacock C, Chodorow M (1998) Combining Local Context and WordNet Similarity for Word Sense Identification. In: Fellbaum C (ed) *WordNet: An electronic lexical database.*, MIT Press, chap 13, pp 265–283
- Liu H, Hu Z, Mian A, Tian H, Zhu X (2014) A new user similarity model to improve the accuracy of collaborative filtering. *Knowl Based Syst* 56:156–166. <https://doi.org/10.1016/j.knosys.2013.11.006>
- Madani Y, Bengourram J, Erritali M (2017a) Social login and data storage in the big data file system hdfs. In: Proceedings of the international conference on compute and data analysis, ICCDA '17, pp 91–97. <https://doi.org/10.1145/3093241.3093265>
- Madani Y, Bengourram J, Erritali M, Hssina B, Birjali M (2017b) Adaptive e-learning using genetic algorithm and sentiments analysis in a big data system. *Int J Adv Comput Sci Appl* 8(8):394–403
- Madani Y, Erritali M, Bengourram J (2017c) Classification of social network data using a dictionary-based approach. In: *Cloud computing and big data: technologies, applications and security*, Springer, Cham, Lecture Notes in Networks and Systems, pp 202–219. https://doi.org/10.1007/978-3-319-97719-5_14
- Madani Y, Erritali M, Bengourram J (2018) Semantic indexing of a corpus. *Int J Grid Distrib Comput* 11:63–80
- Madani Y, Erritali M, Bengourram J (2019a) Sentiment analysis using semantic similarity and Hadoop MapReduce. *Knowl Inf Syst* 59(2):413–436. <https://doi.org/10.1007/s10115-018-1212-z>
- Madani Y, Erritali M, Bengourram J, Sailhan F (2019b) A multilingual fuzzy approach for classifying twitter data using fuzzy logic and semantic similarity. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-019-04357-9>
- Mamdani E, Assilian S (1975) An experiment in linguistic synthesis with a fuzzy logic controller. *Int J Man Mach Stud* 7(1):1–13. [https://doi.org/10.1016/S0020-7373\(75\)80002-2](https://doi.org/10.1016/S0020-7373(75)80002-2)
- Nayebzadeh M, Moazzam A, Saba AM, Abdolrahimpour H, Shahab E (2017) An investigation on social network recommender systems and collaborative filtering techniques. [arXiv:abs/1708.00417](https://arxiv.org/abs/1708.00417)
- Raghuvveer VR, Tripathy BK, Singh T, Khanna S (2014) Reinforcement learning approach towards effective content recommendation in mooc environments. In: 2014 IEEE International Conference on MOOC, Innovation and Technology in Education (MITE), pp 285–289. <https://doi.org/10.1109/MITE.2014.7020289>
- Ren L, Wang W (2018) An svm-based collaborative filtering approach for top-n web services recommendation. *Fut Gen Comput Syst* 78:531–543. <https://doi.org/10.1016/j.future.2017.07.027>
- Shyr WJ (2007) Develop e-learning platform for reinforcement learning on temperature sensor. In: Apolloni B, Howlett RJ, Jain L (eds) *Knowledge-based intelligent information and engineering systems*. Springer, Heidelberg, pp 82–89
- Tizhoosh HR, Shokri M, Kamel M (2007) Reinforcement agents for e-learning applications. Springer London, pp 251–275. https://doi.org/10.1007/978-1-84628-758-9_9
- Wang Y, Deng J, Gao J, Zhang P (2017) A hybrid user similarity model for collaborative filtering. *Inf Sci* 418–419:102–118. <https://doi.org/10.1016/j.ins.2017.08.008>
- Wei J, He J, Chen K, Zhou Y, Tang Z (2017) Collaborative filtering and deep learning based recommendation system for cold start items. *Exp Syst Appl* 69:29–39. <https://doi.org/10.1016/j.eswa.2016.09.040>
- Zhang T, Li WP, Wang L, Yang J (2019) Social recommendation algorithm based on stochastic gradient matrix decomposition in social network. *J Ambient Intell Hum Comput*. <https://doi.org/10.1007/s12652-018-1167-7>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.